The Nature and Development of Processes in Mathematical Investigation

Yeo Boon Wooi Joseph

A thesis submitted to the
National Institute of Education,
Nanyang Technological University
in fulfilment of the requirement for the degree of
Doctor of Philosophy

2013

"Investigation [is] just a vehicle for other learning… This other learning might be seen as learning to be mathematical."

Barbara Jaworski (1994, p. 4)

TABLE OF CONTENTS

PART TWO: RESEARCH METHODOLOGY AND DEVELOPMENT OF DATA ANALYSIS TOOLS

PART THREE: DATA ANALYSIS AND FINDINGS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGEMENTS

I would like to express my heartfelt appreciation to my loving and caring wife, Ivy, for her constant support. She was always by my side, expressing her valuable opinions and inputs for my research, encouraging me to strive forward, taking care of our kids and managing the household so that I could focus on my work and PhD, as well as massaging my neck and shoulders as I worked through long hours. She helped to videotape one of my lessons as well. I would also like to thank my parents for helping to take care of the children.

My research would not be possible without the help rendered by Wong Chi Man, a teacher of the school for the main study. She settled all the administrative works in her school for me by coordinating with the students, booking rooms for the lessons, and helping in the videotaping of the lessons, pretest and posttest. I would like to thank the following volunteers for helping to videotape and supervise the students while they were taking the pretest and posttest: Annabelle Ang, Beh Yu Lin, Belinda Chew, Eugene Lee, Grace Lee, Liu Yi Jie, Ann Neo, Perlin Pek Qu Zhi, Sangeetha d/o Chandra, Jasmine Sia and Anson Tan. I would also like to thank Ivy Chow and Siu Kai Chong for their help in doing the inter-coder reliability tests.

My sincere gratitude goes to my first supervisor, Dr Yeap Ban Har, for providing invaluable assistance in the first part of my research. My sincere thanks also go to my second supervisor, Dr Foong Pui Yee, for guiding me in the second stage of my research. Lastly, I am greatly indebted to my third supervisor, Dr Yap Sook Fwe, for seeing me through the final stretch of my study. Her patience and insights have helped me complete this unforgettable journey.

ABSTRACT

The purpose of this research study is to examine the nature and development of cognitive and metacognitive processes that students use when attempting open investigative tasks. Mathematical investigation is important in many school curricula because many educators think that school students should do some real mathematics, the mathematics which academic mathematicians do in their daily and working lives, investigating and solving problems to discover new mathematics. They believe in the benefits of the processes that these mathematicians engage in, e.g. problem posing, specialising, conjecturing, justifying and generalising. Thus it is vital to understand the nature of these processes (i.e. the types of investigation processes and how they interact with one another), and how they can be developed, so that the teachers are better informed to cultivate these processes in their students. Currently, there is a research gap in this field, as there are few empirical studies on processes in mathematical investigation. Therefore, this research study could add value to the advancement of mathematics education in this area.

The sample for the main study consisted of 10 Secondary Two (equivalent to Grade 8) students from a high- performing Singapore school. They went through a teaching experiment consisting of a familiarisation lesson and five developing lessons. The duration of each lesson was two hours. They sat for a pretest at the end of the familiarisation lesson, and a posttest at the end of the last developing lesson. Each student was separately videotaped thinking aloud while working on two open investigative tasks (one from Type A and the other one from Type B) in each test. The verbal protocols were transcribed and coded using a coding scheme, which had passed

an inter-coder reliability test. The coded transcripts were then analysed qualitatively to validate and refine the two theoretical investigation models for cognitive and metacognitive processes formulated for this research, to study the effect of these processes on the investigation outcomes, and to examine the development of these processes. A scoring rubric was also devised to score the pretest and the posttest in order to study the effect of the teaching experiment on the development of the investigation processes quantitatively using descriptive statistics.

The findings indicated that the two types of investigative tasks tend to elicit different types of investigation processes and investigation pathways: for Type A, students set out to search for any pattern by specialising, conjecturing, justifying and generalising; for Type B, students posed specific problems to solve by using other heuristics, such as reasoning, and then they extended the task by changing the given in order to generalise. Some new cognitive and metacognitive processes and outcomes were also found, which resulted in the refinement of the two theoretical investigation models. Data analysis showed that there was no direct relationship between the completion of an investigation pathway and the types of investigation outcomes produced. The study also identified the processes that had helped the students to produce significant or non-trivial outcomes in their investigation, the processes that were developed more fully in the students during the teaching experiment, and the processes that were still lacking in the students. The implication was that it is possible to develop investigation processes by teaching the students these processes and providing them the opportunity to develop these processes when they attempt suitable investigative tasks. The research also revealed which processes took a longer time to develop, so more attention should be paid to cultivate these processes during teaching.

PART ONE: CURRENT STATE OF RESEARCH ON MATHEMATICAL INVESTIGATION

This thesis is divided into three parts. Part One will survey the current state of research on mathematical investigation to provide a direction of research for the present study. It will consist of two chapters. Chapter 1 will explain the background, purpose and significance of the current study. Chapter 2 will review relevant existing literature on mathematical investigation, thinking processes and research designs.

Part Two of the thesis will describe the research methodology and the development of instruments for data analysis. It will contain three chapters. Chapter 3 will describe the research methodology. Chapter 4 will explain the development of a coding scheme for coding students' verbal protocols during their thinking aloud in the pretest and the posttest. Chapter 5 will elaborate on the development of some tools to analyse the data.

Part Three of the thesis will provide the details of the data analysis and the implications of the findings. It will consist of four chapters. Chapters 6-8 will answer the three research questions. Chapter 9 will conclude the present study by providing some implications of key findings for teaching and directions for further research.

CHAPTER 1: INTRODUCTION TO THE PRESENT RESEARCH STUDY

This research study investigates the nature and development of cognitive and metacognitive processes that Secondary 2 (equivalent to Grade 8) students engage in when they attempt investigative tasks. The nature of the processes refers to the types of investigation processes and how they interact with one another. The rationale for this study stems from the need to address mathematical investigation more holistically in the light of the Singapore mathematics curriculum and the little that is known about students' thinking processes during mathematical investigation. Although some research studies show that their subjects have benefited from doing mathematical investigation, there are very few studies that actually examine how students think when they attempt these tasks. More research is needed to understand the nature and development of students' thinking processes during mathematical investigation so as to inform teachers on how they can develop such processes in their students. This chapter will introduce the present study by providing the background of the research, the rationale for carrying out the present study, a statement of the research problem, and the research questions. It will then discuss the significance of the current study.

1.1 BACKGROUND OF THE RESEARCH

(a) Mathematical Investigation in the Local Scene

In 1997, the Ministry of Education (MOE) in Singapore formulated a vision, *Thinking Schools, Learning Nations* (TSLN), which described a nation of thinking citizens capable of meeting the challenges of the future (Ministry of Education of Singapore, 1997). Since 2003, MOE has focused on an aspect of TSLN: nurturing a spirit of *Innovation and Enterprise* (I&E) which will help to build up a core set of life skills and attitudes that the ministry wants in their students (Ministry of Education of Singapore, 2003). In 2004, Prime Minister Lee Hsien Loong called on the teachers to "teach less to our students so that they will learn more" (Singapore Government, 2004). *Teach Less, Learn More* (TLLM) "builds on the groundwork laid in place by the systemic and structural improvements under TSLN, and the mindset changes encouraged in our schools under I&E" (ibid.). In fact, as far back as 1988, in a paper *Agenda for Action: Goals and Challenges* presented to the Parliament by the then First Deputy Prime Minister Goh Chok Tong, he described that the role of education was "to nurture inquiring minds and to create a lively intellectual environment which will ultimately spread throughout Singapore society" (Yip & Sim, 1990, p. 3). These initiatives serve to emphasise the intention of the Singapore government and its Ministry of Education to nurture thinking and innovation in their students.

In line with these initiatives, the Pentagon Model was adopted as the framework for the Singapore mathematics curriculum in 1990 (Ministry of Education of Singapore, 1990; Wong, 1991). The primary goal of the model is mathematical problem solving and there are five components that help students to attain the goal: skills, concepts, attitudes, metacognition and processes (SCAMP). The latest Pentagon Model, with some minor refinements since its introduction in 1990, is illustrated in Figure 1 below (Ministry of Education of Singapore, 2012).

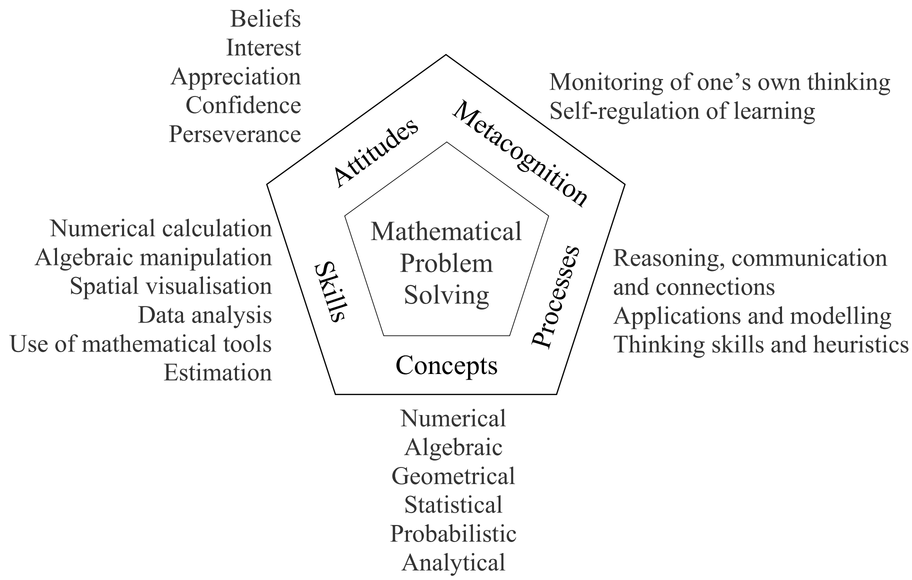


Figure 1. The Pentagon Model

The emphasis on mathematical problem solving signals the intention of Singapore mathematics educators to stimulate their students to think mathematically (Ministry of Education of Singapore, 1990). A list of heuristics and thinking skills is provided in both the primary and secondary syllabi, but the use of these heuristics and thinking skills is not fully reflected in most local textbooks (Fan & Zhu, 2000) and many teachers are not sure how to incorporate them into their teaching (Lee & Fan, 2004). Most teachers also interpret school mathematical problems as consisting only of word problems because such word problems have been predominant in the textbooks (Fan & Zhu, 2000). However, with the introduction of the TSLN vision in 1997 by the Ministry of Education, greater emphasis has been placed on developing thinking students, which translates to the use of more authentic problems in mathematics.

There was very little research on mathematical problem solving in Singapore before the 1990s (Chong, Khoo, Foong, Kaur & Lim-Teo, 1991). But in the 1990s, there was an increasing interest in studies on problem solving, partly because the National Institute of Education (NIE) had begun its Master and PhD programmes in mathematics education, and partly because the Pentagon Model, with its central goal of mathematical problem solving, was implemented in schools in 1990. In the initial stage, many of these research studies were on solving word problems, but with the introduction of the TSLN vision in 1997 and its emphasis on thinking skills, attention was then turned towards non-routine problem solving. From then onwards, there have been many more research studies on this kind of problem solving involving more authentic mathematical problems (Foong, 2009).

Although the Singapore secondary school mathematics curriculum has also specified the use of "open-ended investigations" (Ministry of Education of Singapore, 2000), many secondary school teachers are not familiar with it. Whenever I spoke to these teachers (both beginning and experienced teachers) about mathematical investigation, their most common reply was, "What is that?" The teachers know what problem solving is, because it is the central goal of the Singapore mathematics curriculum framework, but most of them have no idea what an investigation is. Some believe that investigation is just like problem solving, while others think that investigation is similar to guided discovery learning where students are guided to investigate and discover certain mathematical concepts that the teacher has in mind.

If teachers are confused about what constitutes a mathematical investigation, then they may not be able to teach their students effectively (Frobisher, 1994). Doyle (1983) argued that different tasks require different strategies to solve them, and what students learn depends, to a large extent, on the tasks that are given to them. Thus it is important for teachers to understand the types of tasks and their features so that the teachers can choose suitable tasks to elicit the appropriate student learning (Hiebert & Wearne, 1993) because "the nature of tasks can potentially influence and structure the way students think" (Henningsen & Stein, 1997, p. 525). Therefore, there is a need to clarify the construct of mathematical investigation and characterise the thinking processes that students engage in when they attempt investigative tasks.

As a result of the unfamiliarity with mathematical investigation among Singapore teachers, it is not surprising that there are very few studies on investigation. For local Master and PhD dissertations, there are at least 100 empirical studies on mathematical problem solving in the last two decades, but there is only one empirical study on mathematical investigation (Foong, 2007). Moreover, the only Master thesis on mathematical investigation (Ng, 2003) dealt with the general benefits of investigation, without studying the actual thinking processes that students engaged in when they did investigation in detail. Other than the single Master thesis, there is only a theoretical journal paper by Teong (2002) and a conference presentation of an empirical study by Ng, Teo and Leow (2005) on mathematical investigation.

(b) Mathematical Investigation in Other Countries

In other countries, the shift towards mathematical problem solving began much earlier than in Singapore. In the United Kingdom (UK), the Cockcroft Report (Cockcroft, 1982) stated that "the ability to solve problems is at the heart of mathematics" (p. 73). However, the report also mentioned that "mathematics teaching at all levels should include opportunities for … investigational work" (p. 71) and "the idea of investigation is fundamental both to the study of mathematics itself and also to an understanding of the ways in which mathematics can be used to extend knowledge and to solve problems in many fields" (p. 73). In fact, mathematical investigation had already been introduced in the UK during the 1960s by the Association of Teachers of Mathematics (ATM) through workshops and publications, but it was given further emphasis by the Cockcroft Report (Jaworski, 1994). Therefore, problem solving and mathematical investigation are an integral part of many school curricula in the UK.

The Australian national curriculum also stipulates that "mathematical investigations can help students to develop mathematical concepts and can also provide them with experience of some of the processes through which mathematical ideas are generated and tested" (Australian Education Council, 1991, p. 14). The introduction of the Mathematics in the New Zealand Curriculum (MINZC) in 1992 also places a greater emphasis on problem solving and investigation (Hawera, 2006).

In the United States of America (USA), there was a radical swing from the 'back to basics' movement to 'problem solving' in the late 1970s (Askey, 1999; Goldin, 2002; Howson, 1983; Howson, Keitel & Kilpatrick, 1981), culminating in the publication of *An Agenda for Action* by the National Council of Teachers of Mathematics (NCTM) which recommended that "problem solving be the focus of school mathematics in the 1980s" (NCTM, 1980, p. 1). This was supported by the 1980 NCTM Yearbook *Problem Solving in School Mathematics* (Krulik, 1980). Although the Americans do not use the term 'investigation' (Evans, 1987) but they refer to investigative tasks as 'open problems' (Orton & Frobisher, 1996), they do use the term 'to investigate': "Our ideas about problem situations and learning are reflected in the verbs we use to describe student actions (e.g. to investigate, to formulate, to find, to verify) throughout the Standards" (NCTM, 1989, p. 10). Therefore, the Standards do recognise investigation as an integral part of problem solving in the school curricula.

Another goal of the Standards of the National Council of Teachers in Mathematics (NCTM, 1989, 1991, 2000) is to make mathematics classrooms reflect the practices of mathematicians (Brenner & Moschkovich, 2002; Lampert, 1990; Schoenfeld, 1992). This model views students as academic mathematicians and it suggests that students should focus on a variety of rich mathematical activities that parallel what mathematicians do, e.g. posing problems to solve, formulating and testing conjectures, constructing arguments and generalising (Moschkovich, 2002a), which are similar to the processes in problem solving and investigation (Civil, 2002). Therefore, the use of mathematical problems and investigative tasks that stimulate thinking has the potential to create a "microcosm of mathematical culture" (Schoenfeld, 1987, p. 213) in the classroom where students engage in activities that are central to academic mathematicians' practices.

Hence, whether school mathematics curricula are viewed from the perspective of problem solving or from the perspective of academic practice, mathematical investigation plays a very important role in mathematics education. However, just like in Singapore, there is plenty of research on problem solving in other countries like the UK and the USA (Cai, Mamona-Downs & Weber, 2005) but little research on mathematical investigation (see Section 2.1.5 for justification of such a claim). Moreover, the relatively few empirical studies on mathematical investigation (e.g. Bailey, 2007; Tanner, 1989) usually examined the general benefits that their subjects have experienced while doing investigation, without looking into the actual processes that the students engage in and how these processes could be developed. "Research into the effectiveness of process-based teaching … is, however, limited, partly because process-based mathematical learning environments are extremely rare in schools." (Boaler, 1998, p. 42) Therefore, there are gaps in the research on mathematical investigation, especially on the nature and development of processes that students engage in when they attempt investigative tasks.

1.2 RATIONALE FOR CARRYING OUT THE PRESENT RESEARCH

As explained in the previous section, mathematical investigation is very important in many school curricula, including the Singapore curriculum, because many educators believe that investigation can help students develop mathematical concepts and solve problems in many fields. "Investigation [is] just a vehicle for other learning … This other learning might be seen as learning to be mathematical." (Jaworski, 1994, p. 4) However, very little is known about the nature and development of the cognitive and metacognitive processes in investigation, and how mathematical investigation actually helps students to think more mathematically and to apply mathematics in unfamiliar situations, since there are few empirical studies in this area. Therefore, I have decided to undertake the present study in order to contribute to the research on investigation processes, so as to inform educators on how they could incorporate mathematical investigation into the classroom more effectively.

1.3 STATEMENT OF THE RESEARCH PROBLEM

The purpose of the present study is to address some of the research gaps on the nature and development of investigation processes as outlined in the previous sections. The study seeks to understand the nature of the processes in mathematical investigation: how students think when they do mathematical investigation, e.g. how they attempt to understand the task, the types and qualities of the problems that they pose, how they go about solving the problems posed, how they formulate conjectures, whether they try to justify their conjectures or accept these conjectures as true without any reasonable basis, whether they attempt to generalise whenever possible, how they extend the task, what they do when they are stuck, and whether they monitor their own progress. In other words, this study seeks to understand how these investigation processes interact with one another and with the outcomes of the investigation, i.e. the relationship or the effect of the processes on the outcomes, and vice versa.

The sample consisted of 10 Secondary 2 students from one of the high-performing schools in Singapore. The students underwent a teaching experiment to develop their cognitive and metacognitive processes when they attempted two types of open investigative tasks. In addition, the processes were captured by videotape when the students thought aloud during a pretest and a posttest. The interactions of these processes and the outcomes of the investigation can be depicted in a model as investigation pathways. The comparison between the pretest and the posttest will also shed some light on the effectiveness of the teaching experiment on the development of the investigation processes.

1.4 RESEARCH QUESTIONS

The research questions (RQ) for the present study are presented below and reproduced in Chapter 3 as well. The RQ would become clearer after the review of literature in Chapter 2 and the definitions of terms in Chapter 3. RQ1 will examine the types and interaction of the investigation processes from a macroscopic angle by looking at the investigation pathways and their relationship with the outcomes. RQ2 will explore the nature of the processes from a microscopic viewpoint by analysing the interactions between these processes and the outcomes in detail. RQ3 will analyse the development of these processes during the teaching experiment.

RQ1: What is the relationship between the investigation pathways of Secondary 2 students and their outcomes across the two types of investigative tasks?

RQ2: What is the effect of the cognitive and metacognitive processes of Secondary 2 students on the outcomes of their investigation?

RQ3: What is the effect of the teaching experiment on the development of Secondary 2 students' mathematical investigation processes?

1.5 SIGNIFICANCE OF THE RESEARCH STUDY

The present study would contribute to current research in the following areas.

- This study would serve to enhance our understanding of the nature of the cognitive and metacognitive processes engaged by Secondary 2 students when they do investigative tasks, and the effect of these processes on producing significant outcomes. The mathematical investigation models, designed for this study based on theoretical underpinnings and refined by empirical data, would also help to characterise the types and interactions of the processes in mathematical investigation.

- This study would seek to inform teachers on how to develop the thinking processes of their students when they attempt mathematical investigation.

- This study would lend its weight to a growing pool of research studies which suggest that doing investigation helps students to think mathematically and to apply mathematics in unfamiliar situations. The characterisation of the nature and development of the mathematical investigation processes may also help future researchers to study these processes in greater detail and depth.

1.6 CONCLUDING REMARKS FOR THIS CHAPTER

Chapter 1 has set the background, motivation and purpose in carrying out the present research study. Chapter 2 will then review relevant current literature on mathematical investigation, cognitive processes, metacognitive processes and research designs, to set the direction of research and the research methodology for the present study.

CHAPTER 2: REVIEW OF RELEVANT LITERATURE

As this study looks into the nature and development of secondary school students' mathematical investigation processes, there is a need to review relevant research literature on mathematical investigation, cognitive and metacognitive processes, and research methodologies on data collection, analysis and development of processes.

This chapter will begin with a literature review of mathematical investigation and problem solving (Section 2.1). As "there is little doubt that a great deal of overlap exists between problems and investigations" (Frobisher, 1994, p. 152), there is a need to understand what constitutes an investigative task so as to draw a clear boundary on the area of research in this study, especially when there are other types of tasks that seem to contain some elements of investigation, e.g. guided discovery and open problems. Empirical studies on mathematical investigation will also be reviewed to guide and inform the present research. As there is an overlap between investigation and problem solving processes, the literature review of cognitive processes (Section 2.2) and metacognitive processes (Section 2.3) will include those done on both investigation and problem solving, since there is a lack of research studies on the processes of mathematical investigation. The review will include theoretical models of investigation and problem solving processes that could be modified and used as a theoretical model for the current study. Lastly, selected literature on methods of data collection, analysis and development of processes will be examined to inform the research methodology for the present study (Section 2.4).

2.1 LITERATURE REVIEW ON MATHEMATICAL INVESTIGATION

Before any research on mathematical investigation can be carried out, it is essential to understand the similarities and differences between an investigative task and a mathematical problem as described in current literature so that the construct of investigation can be clearly defined for the present study (Section 2.1.1). This is especially important because there is no clear and consistent definition of the construct of investigation in existing literature as there are conflicting views about what constitutes an investigation. Moreover, the overlap between mathematical investigation and problem solving is also not clearly explicated in current literature.

It was found necessary during an extensive review of literature to separate an investigative task from investigation as a process (Section 2.1.2), and from investigation as an activity involving an open investigative task (Section 2.1.3), in order to resolve the conflicting views. There is also a need to consider two types of investigative tasks for the current research because of the two types of problem-posing processes (Section 2.1.4). Finally, empirical studies on mathematical investigation will be examined to set the direction of research for the present study (Section 2.1.5).

2.1.1 Mathematical Problems and Investigative Tasks

Many teachers often use the word ‘problems’ to describe the typical exercises in mathematics textbooks (Fan & Zhu, 2000), but are these exercises really problems? And what is the difference between a mathematical problem and an investigation? In this section, what constitutes a mathematical problem and an investigative task will be clarified. An extensive review of relevant literature has brought forth two different viewpoints about the characteristics of a mathematical problem.

(a) First Viewpoint of Mathematical Problem

The first viewpoint is that whether a situation is a problem depends on the individual. One of the earliest references to this view was from Henderson and Pingry (1953). They believed that a situation is a problem to a person if he desires to obtain a goal but is unable to obtain it straight away. Similarly, Reys, Lindquist, Lambdin, Smith and Suydam (2012) defined a problem as “a situation in which a person wants something and does not know immediately what to do to get it” (p. 115) and that this difficulty requires “some creative effort and higher-level thinking” (ibid.) to resolve. Schoenfeld (1985) also emphasised that the “difficulty should be an intellectual impasse rather than a computational one” (p. 74). He gave the example that inverting a 27×27 matrix would be a tedious task for him but inverting a matrix was not a problem to him. Thus tediousness in applying a computational procedure is not a factor in determining whether a situation is a problem. However, what happens if the students do not know how to invert a 27×27 matrix? They may have learnt how to invert a 2×2 matrix but not many of them are aware of the procedure to invert a square matrix of a higher order. Therefore, inverting a 27×27 matrix can still be a problem to these students. Hence, it appears that it is possible for a problem to be a procedural one. Let us examine this idea in greater detail by considering a typical textbook exercise question:

Task 1: Quadratic Equation

Solve the quadratic equation $x^2 + 2x - 3 = 0$.

This task may just be a routine practice of procedural skills that students have learnt earlier in the class (Moschkovich, 2002a) and so they may know immediately what to do to solve it. However, this can be a problem to students who have not been taught the procedure, or a problem to low-achieving students who have just learnt the procedure but do not know how to apply it properly. Nevertheless, with enough practice, this task can become a routine exercise to the students. Some educators call this type of tasks "routine problems" (Orton & Frobisher, 1996, p. 27), but these tasks may not be problems to some students. Moreover, for students who do not practise these 'routine' tasks found in the textbook, then these tasks are not even 'routine' to them. Cockcroft (1982) used the term 'familiar or unfamiliar tasks' to indicate whether the tasks are familiar or unfamiliar to a student. Let us contrast Task 1 with another example:

Task 2: Last Digit

Find the last digit of 3^{2012}.

The main purpose of this task is for students to make use of some problem-solving strategies, such as looking for patterns, to solve it. But for students who have been exposed to such tasks before, this task may no longer be a problem. Moreover, this task may not pose a problem to high-ability students who have not encountered such problems before but are able to solve it without much difficulty. Hence, from the first viewpoint, both Tasks 1 and 2 can be problems to students who are "*unable to proceed directly* to a solution" (Lester, 1980, p. 30). Schoenfeld (1985) believed that "being a 'problem' is not a property inherent in a mathematical task" (p. 74). Therefore, whether a mathematical task is a problem depends on the individual.

(b) Second Viewpoint of Mathematical Problem

The second viewpoint of what constitutes a mathematical problem involves the nature and purpose of the task. Task 1 (Quadratic Equation) is not a mathematical problem because it requires only a procedure to solve it and the purpose of such a task is to "provide students with practice in using standard mathematical procedures, for example, computational algorithms, algebraic manipulations, and use of formulas" (Lester, 1980, p. 31). Task 2 (Last Digit) is different from Task 1 because Task 2 requires "some creative effort and higher-level thinking" (Reys et al., 2012, p. 115) to solve, not just a direct application of a procedure. Therefore, Task 2 is a mathematical problem even though it may not pose a problem to some high-ability students. In this respect, the second viewpoint is different from the first one.

Since Tasks 1 and 2 are inherently different and they may or may not be a problem to an individual, some educators (e.g. NCTM, 1991, p. 25; Schoenfeld, 1985, p. 74) have used the phrase 'mathematical tasks' instead of 'mathematical problems'. Therefore, Task 1 will be called a 'procedural task' since it involves the practice of procedures, while Task 2 can be called a 'problem-solving task' since it requires the use of some problem-solving strategies. The phrase 'problem-solving task' can be misleading because the term 'problem-solving' suggests that the task may pose a problem to the person when it may not be so. Nevertheless, the phrase 'problem-solving task' will still be used to emphasise the problem-solving strategies involved in this type of tasks, even if the task may not pose a problem to some students.

(c) Differences between Investigative Task and Mathematical Problem

However, there is another important criterion in the second viewpoint to decide whether a mathematical task is a problem: the existence of "a clearly defined goal" (Henderson & Pingry, 1953, p. 230). For example, Task 2 has a clearly defined goal in its task statement: find the last digit. Contrast this with the next example:

> **Task 3: Powers of 3**
>
> Powers of 3 are 3^1, 3^2, 3^3, 3^4, … Investigate.

Orton and Frobisher (1996) believed that mathematical tasks, such as Task 3, do not specify a goal in their task statements. However, from another perspective, the word 'investigate' in the task statement is still a goal, albeit an ill-defined "general goal" (ibid., p. 26), and students can choose any specific goal to investigate (Cai & Cifarelli, 2005), e.g. is there a pattern in the last digit of consecutive powers of 3? Nevertheless, Task 3 does not have a clearly defined goal since it does not tell the students what to investigate. Thus "very few mathematics educators would classify explorations of this kind as problems" (Orton & Frobisher, 1996, p. 27). On the other hand, mathematics educators in some countries (e.g. the USA) would call Task 3 an 'open problem' (Evans, 1987) which is defined to be a problem "when no goal is specified" (Orton & Frobisher, 1996, p. 32). But if the rather strict definition that a mathematical problem must have a clearly-defined goal is used, the term 'open problem' is an oxymoron.

In this thesis, Task 3 (Powers of 3) will be called an investigative task, as its purpose is to investigate any pattern: this kind of tasks must be open in the sense that the task statement does not contain any clearly defined goal; while Task 2 (Last Digit) will be called a problem-solving task or a mathematical problem, as it has a clearly defined goal and it involves the use of some problem-solving strategies. Although Task 1 (Quadratic Equation) also has a clearly defined goal, it is called a procedural task, as explained earlier. On the other hand, all the three types of tasks can pose a problem to any individual who does not know how to solve them. For example, if a student does not know what to investigate for the investigative Task 3, then the task is still a problem to that student.

Other than the difference in the goal of the task as explained above, another difference between an investigative task and a mathematical problem is that the former have multiple correct answers while the latter has only one correct answer. For example, Task 2 has only one correct answer (the last digit of 3^{2012} is 1) while Task 3 has multiple correct answers, such as:

- the last digit of powers of 3 has a repeating pattern of period 4: 3, 9, 7, 1;
- the last two digits of powers of 3 has a repeating pattern of period 20:
  03, 09, 27, 81, 43, 29, 87, 61, 83, 49, 47, 41, 23, 69, 07, 21, 63, 89, 67, 01;
- the sum of all the digits of powers of 3 is divisible by 3.

Becker and Shimada (1997) and Chow (2004) called mathematical tasks with multiple correct answers, such as Task 3, 'open-ended' because the answer, which is at the 'end' of solving the task, is open. Bailey (2007) defined investigation as "an open-ended problem or statement that lends itself to the possibility of multiple mathematical pathways being explored, leading to a variety of mathematical ideas and / or solutions" (p. 103). Therefore, investigative tasks are open in the sense that students can set different goals to pursue and as a result, there are multiple correct answers; while mathematical problems are closed in the sense that there is a clearly defined goal which will lead to only one correct answer (Evans, 1987).

On the other hand, it is also possible to extend a closed mathematical problem. For example, in Task 2, after finding the last digit of 3^{2012}, a student may extend the problem to find the last two digits of 3^{2012}. Thus some educators (e.g. Maher, 2005; Orton & Frobisher, 1996) would also like to consider mathematical problems to be open in the sense that they could be extended. The idea of extension in solving a mathematical problem is the changing of the given in the original task statement (Brown & Walter, 2005). In Task 2, the given is 'find the last digit' and an extension is to change the given to 'find the last two digits'. However, for Task 3, which is an investigative task, students can find all the patterns for powers of 3 under the umbrella of the original task without having to change the given, i.e. without extending the investigative task.

(d) Summary of Literature on Investigative Tasks and Mathematical Problems

To summarise, this thesis uses the term 'mathematical tasks' to refer to all types of tasks involving mathematics. Examples of such tasks discussed so far are 'procedural tasks', 'problem-solving tasks' (which is synonymous with 'mathematical problems') and 'investigative tasks'. From another perspective, all of these tasks can be problems to students who do not know how to do or solve them. From the above review of current literature, there are at least two main differences between an investigative task and a mathematical problem:

(i) An investigative task is open in the sense that it does not have a clearly defined goal in its task statement and students are expected to set their own goals to investigate so as to discover any underlying pattern or mathematical structure; while a mathematical problem is closed in the sense that it has a clearly defined goal in terms of a problem for students to solve, although the students can extend the problem by changing the given goal.

(ii) An investigative task is open in the sense that it has multiple correct answers; while a mathematical problem is closed in the sense that it has only one correct answer, although students can extend the problem which can then lead to multiple correct answers in its various extensions.

2.1.2 Problem Solving and Mathematical Investigation as Processes

In this section, the similarities and differences between problem solving and mathematical investigation as processes will be examined.

(a) Conflicting Views about Mathematical Investigation

Although many mathematics educators (e.g. Orton & Frobisher, 1996; Pirie, 1987) agree that there is a great deal of overlap between problem solving and investigation, these same educators still end up separating them into two distinct processes: investigation must involve (open) investigative tasks while problem solving must involve (closed) mathematical problems. Other educators also stress the openness of investigation. For example, Bastow, Hughes, Kissane and Mortlock (1991) defined mathematical investigation as the "systematic exploration of open situations that have mathematical features" (p. 1), and Lee and Miller (1997) believed that "investigations, by their very nature, demand an open-minded, multifaceted approach" (p. 6). Delaney (1996) believed in the more open spirit of the process-dominated investigation while Ernest (1991) described investigation as the exploration of an unknown land without any fixed destination and thus open. Evans (1987) observed that an investigative task leads to a divergent activity since students can set different goals to pursue, but a mathematical problem leads to "problem solving [which] is a convergent activity" (p. 27) since there is only one goal to achieve.

Therefore, the questions that need to be addressed are, "If most literature separate investigation and problem solving so distinctly, then why do the same writers also claim that these two processes overlap? In what ways do they overlap and in what ways are they different? Must mathematical investigation be open?"

(b) Separating Investigative Tasks from the Process of Investigation

After further research, it was discovered that the problem lies in the usage of the term 'investigation', which is used by different educators to mean different constructs. For example, Orton and Frobisher (1996) used the term 'investigation' to refer to the task when they compared investigations with problems, while Evans (1987) used the term 'investigation' to mean the process when he contrasted investigation with problem solving. Ernest (1991) has observed that there has been a fairly widespread adoption of the term 'investigation' as the task itself when investigation is actually a process. This is what Jakobsen (1956, as cited in Ernest, 1991) called a metonymic shift in meaning, which replaces the whole activity with one of its components. Thus most educators do not seem to distinguish between the investigation process and the investigative task.

However, there is a need to differentiate between these two constructs because it will resolve the apparent conflict that most literature separate investigation and problem solving so distinctly and yet the same writers also claim that these two processes overlap. This distinction is not merely academic as it will also shed light on the processes involved in mathematical investigation and problem solving, which will become clearer at the end of this section.

(c) Characterising the Process of Investigation

First, mathematical investigation as a process will be characterised from various sources. The Cambridge Dictionaries Online (Cambridge University Press, 2012) defined the word 'investigate' as to 'examine a crime, problem, statement, etc. carefully, especially to discover the truth'. Height (1989) explained that "the essence of mathematical investigation is to inquire into situations which are of a mathematical nature" (p. 1). Thus mathematical investigation, as a process, is a careful examination of a mathematical task in order to discover some underlying mathematical facts or structures. There are two different perspectives on the 'process' in a mathematical investigation: from one angle, an investigation is one whole process (Frobisher, 1994), just like problem solving is one whole process (Shufelt, 1983); from another angle, there are many processes involved in an investigation (Frobisher, 1994). So what types of processes does the process of mathematical investigation involve?

Jaworski (1994) described the teaching of three teachers as "investigative in spirit, embodying questioning and inquiry" (p. 96). Her conceptualisation of an investigative approach to mathematics teaching is one that involves specialising, conjecturing, justifying and generalising. Speaking of the same three teachers, Jaworski clarified, "Making and justifying conjectures was common to all three classrooms, as was seeking generality through exploration of special cases" (p. 171). Exploration of special cases or specific examples is called 'specialising' by Mason, Burton and Stacey (1985). In fact, the main processes in most models of mathematical investigation (e.g. Frobisher, 1994; Height, 1989) are specialising, conjecturing, justifying and generalising (see Section 2.2.2 for the models). These are the four main mathematical thinking processes advocated by Mason et al. (1985) which they have applied to solving mathematical problems rather than investigative tasks.

Pólya (1957) also advocated something similar to mathematical investigation in his approach to mathematics teaching. He called his approach 'heuristic reasoning', as opposed to rigorous proof. "Heuristic reasoning is often based on induction or on analogy" (p. 113), and both induction and analogy involve examining specific examples, which is also called 'specialising' by Mason et al. (1985). Therefore, Pólya's (1957) idea of heuristic reasoning involves specialising with the intention to generalise from specific examples, or to use analogy to discover some mathematical facts. Along the way, the person will formulate and justify some conjectures. But these processes are similar to those for mathematical investigation discussed in the previous paragraph. Pólya wrote that "we need heuristic reasoning when we construct a strict proof as we need scaffolding when we erect a building" (p. 113) and he believed that it is bad to confuse heuristic reasoning with rigorous proof.

Lakatos (1976) called his approach to mathematics teaching a 'heuristic approach' to proofs and refutations, as opposed to a deductivist approach of using formal proofs. In the heuristic approach, students explore the problem by examining specific examples in order to come up with some conjectures which will then be proven or refuted. Again, these processes are similar to those for mathematical investigation. Lakatos was against the deductivist approach of using formal proofs that come out of nowhere because "it seems impossible that anyone should ever have guessed them" (p. 142). He observed that the "deductivist style hides the struggle, hides the adventure" (ibid.) of discovering a solution to a problem or a proof for a theorem, and "the zig-zag of discovery cannot be discerned in the end-product" (p. 42) of the deductivist approach. Although both the heuristic approach and the deductivist approach use the process of reasoning, the kinds of reasoning are different for each approach.

Therefore, both Pólya's (1957) heuristic reasoning and Lakatos' (1976) heuristic approach involve processes similar to those of mathematical investigation, as opposed to rigorous or formal proofs. However, both Pólya (1957) and Lakatos (1976) used mathematical problems and not investigative tasks in their writings. Hence, mathematical investigation, as a process, does not depend on the open nature of investigative tasks. Even when faced with a mathematical problem which has a clearly defined goal, students can still engage in mathematical investigation if they try to specialise with the intention to formulate and test conjectures so as to generalise. Hence, by looking at literature from various sources, the whole idea of a mathematical investigation is a process involving the four main thinking processes of specialising, conjecturing, justifying and generalising. These processes will be examined in more detail in Section 2.2.3 later.

(d) Relationship between Investigation and Problem Solving

However, these four main thinking processes in investigation are very similar to the processes in problem solving. In fact, Mason et al. (1985) have applied these four processes to problem solving rather than mathematical investigation. So, what is the difference between investigation and problem solving? The difference will become clearer from the discussion of the following mathematical problem.

> **Task 4: Handshake Problem**
>
> At a workshop, each of the 20 participants shakes hands once with each of the other participants. Find the total number of handshakes.

There are several methods available to solve this problem. Some higher-ability students are able to reason as follows: since every different pair of participants will give rise to one distinct handshake, the total number of handshakes is the same as the total number of different pairs of participants, i.e. $^{20}C_2$. This first method involves a 'deductive approach' using some logical arguments and there is no sense that the students are investigating since they do not specialise, formulate or test conjecture, or generalise. On the other hand, some students will not be able to reason like this directly. They may try to solve the problem by starting with smaller numbers of participants (specialising) in order to find a pattern for the total number of handshakes (generalising). Along the way, they may formulate and test their conjectures. Thus this second method involves an 'inductive approach' where there is the sense that the students are doing some investigation to solve the problem. In fact, some teachers do tell their students to 'investigate' when they do not know how to solve such mathematical problems immediately (personal communication).

Hence, there are generally two approaches to problem solving: the inductive approach which involves specialising, conjecturing, justifying and generalising (which is the investigation process), and the deductive approach which uses other heuristics such as logical arguments. These are the same two approaches discussed by Pólya (1957) and Lakatos (1976) as explained earlier in this section. In other words, problem solving involves investigation as a process. Figure 2.1 illustrates this relationship between mathematical investigation and problem solving.

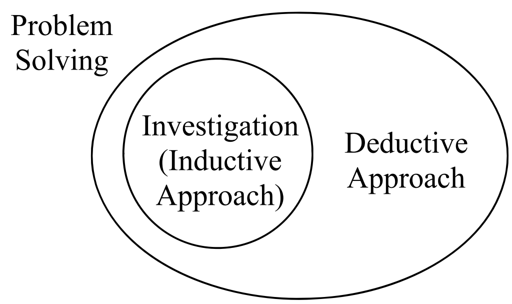


Figure 2.1 Relationship between Investigation and Problem Solving

A caveat is necessary here. By separating the inductive approach (or the investigation process) from the deductive approach to problem solving so neatly in this thesis, it does not mean that there is no deductive reasoning in mathematical investigation. In fact, some form of reasoning is required throughout the entire investigation process:

- specialising may require some form of reasoning to guide the choice of examples to examine;
- justifying conjectures make use of some form of reasoning or deductive proof.

However, the main characteristic of the investigation process is that it starts with specialising, while the deductive approach uses reasoning without specialising. If a problem can be solved using other heuristics, such as reasoning, without formulating any conjecture, then the deductive approach is quite distinct from the inductive approach. But the deductive approach may sometimes end up needing a conjecture to be proven. This is where problem solving overlaps investigation, since both of them now involve the main processes of conjecturing and justifying. A further argument to prove there are similarities between the investigation process and problem solving is to consider the following investigative task which is obtained by rephrasing Task 4 (Handshake Problem). The opening up of mathematical problems to become investigative tasks will be discussed in more detail in Section 2.1.4 later.

Task 5: Investigate Handshakes

At a workshop, each of the 20 participants shakes hands once with each of the other participants. Investigate.

Tasks 4 and 5 look similar except for the last part of the task statement. As explained earlier in this section, Task 4 is a mathematical problem because the goal is clearly defined: find the total number of handshakes. So when students try to solve Task 4, they are engaging in problem solving. However, Task 5 is an investigative task without a clearly defined goal: investigate, but investigate what? So when students attempt Task 5, they are engaging in mathematical investigation.

Suppose the first problem that the students pose for Task 5 is to find the total number of handshakes. If the students use the *same* method to find the total number of handshakes for both the problem-solving Task 4 and the investigative Task 5, then why is it that the *same* process is called problem solving for the former task but investigation for the latter task? If a student specialises in order to solve the problem-solving Task 4, then the student is actually solving a mathematical problem by investigation. On the other hand, if a student uses other heuristics such as deductive reasoning to solve a problem in the investigative Task 5, then the student is not investigating in that sense. Therefore, this further substantiates that problem solving and investigation are closely related, and that investigation does not depend on whether the task is a mathematical problem or an investigative task.

After separating the investigation process from the investigative task, and characterising the investigation process as consisting of specialising, conjecturing, justifying and generalising, we are now in a better position to understand the apparent contradiction in current literature on mathematical investigation as described earlier in part (a) of this section: investigative tasks must be open while mathematical problems must be closed, and yet there are overlaps between investigation and problem solving.

(e) Resolving Conflicting Views about Mathematical Investigation

We will begin by looking at the following task:

> **Task 6: Number Trick**
>
> Jill has a trick she does with numbers. Here it is. How do you think it works?
>
> $$\begin{array}{r} 854 \\ \underline{-\ 458} \\ 396 \\ \underline{+\ 693} \\ \underline{1089} \end{array}$$
>
> Jill says that every time she does her trick, the answer is always 1089.
> Investigate Jill's trick. (Orton & Frobisher, 1996, p. 39)

Although the final sentence in the task statement tells the students to investigate Jill's trick, the task statement also states the goal clearly in the third sentence: "How do you think it works?" Thus Task 6 is a mathematical problem according to what has been discussed in literature in Section 2.1.1 since an investigation must not contain any clearly defined goal (Orton & Frobisher, 1996). But these same writers gave this task as an example of an investigation. This contradiction could easily be resolved if the investigation process was separated from the investigative task. Although Task 6 is not an investigative task, it usually involves the investigation process because most students would start with some specific examples (specialising) before they try to generalise. They might formulate some conjectures, e.g. Jill's trick only works for certain numbers, and they would have to test these conjectures by finding counter examples or by using some reasoning or deductive proofs. Therefore, most students will do some investigation to try to solve this problem.

Another example of an apparent contradiction in literature is how Maher (2005) used the term "structuring their investigations" (p. 1) when she described how her students approached what she called "open-ended and well-defined tasks" (p. 2). An example of such a task is to find the number of 5-cube-tall towers that can be built if the cubes have two colours. From the literature review in Section 2.1.1, this task has a well-defined goal and so it is a mathematical problem. But what makes it open-ended is that the students could pose other problems, e.g. a student called Ankur extended the task by asking, "Find as many towers as possible that are 4-cube tall if you can select from three colors and there must be at least one of each color in each tower" (ibid., p. 6). However, the extension of a mathematical problem is different from an investigative task. But if the investigation process is separated from the investigative task, then Maher's idea of an investigation is the same as the investigation process which she has applied to mathematical problems instead of investigative tasks.

(f) Summary of Literature Review on Investigation as a Process

To summarise, there is a difference between an investigative task and the process of investigation. In this thesis, an investigative task will always be open in the sense that it will not have a clearly defined goal, while the investigation process will be characterised by the four main mathematical thinking processes of specialising, conjecturing, justifying and generalising. There are generally two approaches to problem solving: the inductive approach involving specialising (which is similar to the investigation process); and the deductive approach involving heuristics other than specialising (which will be called 'other heuristics' in this thesis), such as the use of reasoning. Thus problem solving involves the process of investigation. On one hand, it is possible for students to engage in the investigation process during the solving of a mathematical problem. On the other hand, it is possible for students *not* to be involved in the investigation process during the solving of specific problems when attempting an investigative task if they have solved the problems using other heuristics. However, the students will still be engaged in investigation as an *activity* when they attempt the investigative task, which will be explained in the next section.

2.1.3 Problem Solving and Mathematical Investigation as Activities

In this section, the similarities and differences between problem solving and mathematical investigation as activities will be examined.

(a) Another Conflicting View about Investigation and Problem Solving

Many educators (e.g. Cai & Cifarelli, 2005; Height, 1989) believe that mathematical investigation involves both problem posing and problem solving because students need to pose their own problems to solve when given an investigative task. But this contradicts the notion that problem solving involves the process of investigation, as explained earlier in Section 2.1.2. Thus there is a need to resolve this issue by separating investigation as a process from investigation as an activity.

(b) Difference between Investigative Task and Investigation as an Activity

The usage of the term 'activity' in mathematics education research literature is not new. Christiansen and Walther (1986) distinguished a task from an activity even though these two terms are often treated as synonyms (Mason & Johnston-Wilder, 2006). A task refers to what the teacher sets while the activity refers to what the student does in response to the task (Christiansen & Walther, 1986). "The purpose of a task is to initiate mathematically fruitful activity [by] learners" (Mason & Johnston-Wilder, 2006, p. 25). The distinction between a task and an activity is important because the original purpose of a task may be lost during its implementation (Stein, Grover & Henningsen, 1996). Some educators (e.g. Jaworski, 1994; Mason, 1978) are worried that teachers may teach mathematical investigation in an algorithmic manner by stereotyping certain mathematical processes as a set of procedures to be learnt by students. For example, Lerman (1989) observed a lesson by an experienced teacher who taught mathematical investigation by telling his students what to do to arrive at an answer when they were stuck, instead of asking guiding questions to stimulate thinking and further investigation. Thus a task that is intended to be open can be closed by the teacher in its implementation.

Therefore, there is a need to separate the activity from the task. The term 'investigation as an activity' will be used to refer to the mathematical activity performed by the students when they attempt an investigative task, while students solving mathematical problems will be engaged in 'problem solving as an activity'.

(c) Difference between Investigation as a Process and Investigation as an Activity

However, the above only explains the difference between an investigative task and investigation as an activity. What about the difference between investigation as a process and investigation as an activity, and the difference between problem solving as a process and problem solving as an activity?

An analogy would be helpful. The four phases of Pólya's (1957) mathematical problem-solving model are (i) understanding the problem, (ii) devising a plan, (iii) carrying out the plan, and (iv) looking back. From another perspective, the first phase of understanding the problem is what a student should do *before* solving the problem and the fourth phase of looking back is what the student should do *after* solving the problem. The actual process of solving the problem begins in the second phase of devising a plan and continues into the third phase of carrying out the plan. This could be made clearer if the entire mathematical *activity* of problem solving is distinguished from the actual *process* of solving the problem. Another real life analogy would be cooking. *Before* a person can cook, he or she has to prepare the ingredients. *After* the person has cooked, he or she has to scoop up the food from the pan or pot into a dish.

However, the whole *activity* of cooking involves not only the actual *process* of cooking, but what the person needs to do before and after the cooking.

Similarly, when students attempt an investigative task, they are engaged in investigation as an activity. This activity includes processes such as understanding the task and posing a problem to solve *before* the process of problem solving. As explained in Section 2.1.2, the students can then solve the problem by using specialising (inductive approach or the process of investigation) or by using other heuristics (deductive approach). *After* problem solving, the students can engage in the process of checking the solution, or the process of posing more problems to solve without changing the given in the original task, or the process of extending the task by changing the given. The relationships among investigation as an activity involving an investigative task, problem solving as a process, and investigation as a process, are explicated in the model in Figure 2.2.

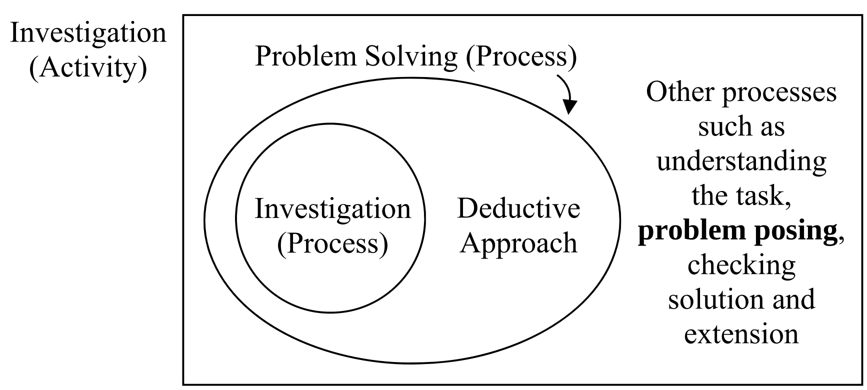


Figure 2.2 Mathematical Investigation as an Activity

From the model in Figure 2.2, it is observed that investigation (as an activity) involves problem solving (as a process), while problem solving (as a process) involves investigation (as a process). Therefore, the conflicting views about the relationships between investigation and problem solving could be resolved if the three constructs (namely, task, process and activity) are separated.

To complete the discussion, Figure 2.3 shows the relationship between problem solving as an activity involving a mathematical problem, and investigation as a process. A main difference between problem solving as an activity in Figure 2.3 and investigation as an activity in Figure 2.2 is the additional process of problem posing in investigation. The purpose of this kind of problem posing is to generate problems to solve, which is different from posing a simpler or related problem as a problem-solving heuristic. The differences among the various types of problem posing will be discussed in more detail in Section 2.1.4 and Section 2.2.3(b).

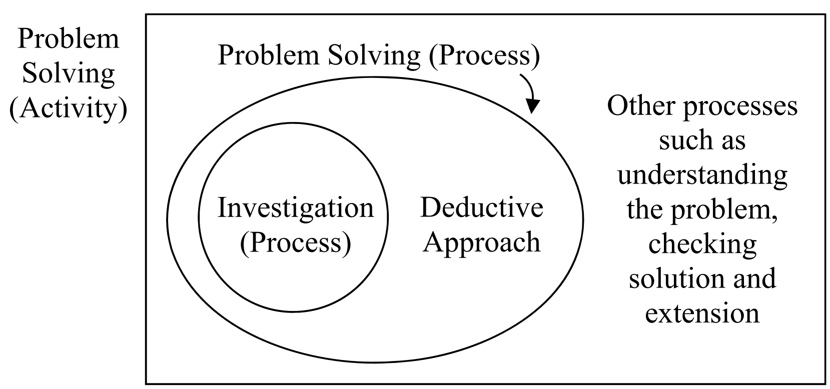


Figure 2.3 Problem Solving as an Activity

(d) Summary of Literature Review on Investigation as an Activity

To summarise the literature review so far, the differences between investigative tasks and mathematical problems have been examined, and the relationships between investigation and problem solving as both processes and activities have been clarified. This will help to define the terms used in the thesis (see Section 3.5 later). In addition, from this point onwards, the term ‘investigation’ will always be used to refer to an investigation activity involving an (open) investigative task; but when there is a need to refer to investigation as a process, i.e. the inductive approach to problem solving, then either the term ‘investigation process’ or ‘process of investigation’ will be used.

2.1.4 Two Types of Investigative Tasks: Type A and Type B

As investigation consists of problem posing and problem solving, there is a need to review literature on problem posing. This will be done under the literature review of cognitive processes later in Section 2.2.3(b). In this section, there is a need to address another issue regarding problem posing involving investigative tasks such as Task 3 which is reproduced below:

Task 3: Powers of 3

Powers of 3 are 3^1, 3^2, 3^3, 3^4, … Investigate.

(a) Conflicting View about Investigation and Problem Posing

As explained in Section 2.1.1, Task 3 does not contain a clearly-defined goal but a “general goal” (Orton & Frobisher, 1996, p. 26) as suggested by the word ‘investigate’. The *general* goal of such investigative tasks is to find the underlying patterns or mathematical structures. To tie in with the literature on problem posing (e.g. Brown & Walter, 2005), the students can pose the *general* problem, “Is there any pattern? If yes, what is the pattern?” To attain the general goal, students are supposed to set more *specific* goals to investigate (Cai & Cifarelli, 2005) or pose more *specific* problems to solve (Frobisher, 1994; Krutetskii, 1976). To use Task 3 as an illustration, students could pose specific problems like these:

- Is there a pattern in the last digit of consecutive powers of 3?
- Is there a pattern in the last two digits of consecutive powers of 3?

However, anecdotal evidence suggests that most students do not even know what kinds of specific problems to pose, so they will just search for *any* pattern. This was supported by findings from an initial exploratory study conducted by me (see Section 3.3), where quite a number of students searched for any pattern without posing any specific problem. In fact, many of these students did not even find a pattern in the last digit or the last two digits of consecutive powers of 3, but they found patterns in the following:

- the first digit of consecutive powers of 3;
- the total number of digits of consecutive powers of 3;
- the sum of digits of consecutive powers of 3.

The students in the initial exploratory study generated some powers of 3 and then tried to find *any* pattern *anywhere*. For example, they looked at the first digits in their examples, could not observe any pattern, then moved on to the second digits and so forth. When that failed, some of them even tried to count the total number of digits to see if there was any pattern. In other words, the students did not set out to find a specific pattern, but they tried to find any pattern anywhere. This approach is different from posing a specific problem right at the beginning to, say, search for a pattern in the first digit of powers of 3 and then generating examples for the sole purpose of examining the first digit of powers of 3.

Therefore, for investigative tasks such as Task 3 (Powers of 3), there is actually no need to pose any specific problem because the students can just search for *any* pattern, which is the general goal or general problem of such investigative tasks. This appears to contradict some literature (e.g. Cai & Cifarelli, 2005; Krutetskii, 1976) which suggest that students should set more specific goals or pose more specific problems to investigate.

(b) Converting Mathematical Problems into Investigative Tasks

To resolve the issue that students should pose specific problems in investigative tasks, there is a need to consider a second type of investigative tasks that are obtained by opening up mathematical problems. Frobisher (1994) suggested that it is "nearly always possible to restate [a problem] in order to make it into an investigation" (p. 158). Task 4, which was first stated in Section 2.1.2 and reproduced here, will be used as an example:

Task 4: Handshake Problem

At a workshop, each of the 20 participants shakes hands once with each of the other participants. Find the total number of handshakes.

To illustrate Frobisher's (1994) idea using Task 4, the task could be opened up by removing the intended problem from the task statement and replacing it with the word 'Investigate' (see Task 5).

Task 5: Investigate Handshakes

At a workshop, each of the 20 participants shakes hands once with each of the other participants. Investigate.

The purpose of opening up a mathematical problem, such as Task 4, to become an open investigative task, such as Task 5, is to allow students to pose their own problems to solve from the beginning. Frobisher (1994) believed that students need to pose more specific problems, such as the original intended problem of finding the total number of handshakes, because it is not easy for students to just search for any pattern for this kind of tasks without having a specific problem in mind. Another specific problem that students can pose at a later stage is to find a general formula for the total number of handshakes for *n* participants.

(c) Type A and Type B Investigative Tasks

However, Task 5 is different from the usual investigative tasks found in most literature, such as Task 3 (Powers of 3), in two ways:

- Task 5 requires students to first pose a specific problem to solve because they probably could not search for any pattern without having any specific problem to solve; while Task 3 allows students to pose the general problem of searching for any pattern without the need to pose any specific problem.

- Task 3 requires students to investigate by specialising; while Task 5 allows the use of other heuristics, such as reasoning, to solve the specific problem posed, although it is sometimes possible to use specialising to solve.

Therefore, there seem to be two different types of investigative tasks that will elicit somewhat different kinds of processes. The usual investigative tasks found in most literature, such as Task 3 (Powers of 3), will be called Type A investigative tasks; while investigative tasks converted from mathematical problems, such as Task 5 (Investigate Handshakes), will be called Type B investigative tasks. This thesis will examine the processes that students engage in when attempting both of these types of tasks. An interesting issue for the present study is to examine whether the students are able to pose the original intended problem for Type B tasks, which will be addressed in the data analysis in Section 7.3.2(a).

(d) Other Types of Tasks, Activities and Pedagogies Related to Investigation

There are other types of tasks, activities and pedagogies similar to mathematical investigation. For example, some researchers (e.g. Cifarelli & Cai, 2004) use the term 'exploration' to refer to investigation involving open investigative tasks, while others (e.g. Brown, 1996) use the same term to refer to the process of investigation for mathematical problems. Similarly, some educators (e.g. Becker & Shimada, 1997; Orton & Frobisher, 1996) use the terms 'open problems' and 'open-ended problems' to refer to investigative tasks, but others (e.g. Maher, 2005) use the same terms to refer to mathematical problems that can be extended. On the other hand, one would expect an investigative approach to the teaching of mathematics to include investigative tasks, but some researchers (e.g. Orton & Frobisher, 1996) believe that an *ad hoc* use of investigative tasks in the classroom is *not* an investigative approach. Rather, an investigative approach involves a drastic change of the traditional method of teaching where the teacher will no longer teach the content. Instead, the teacher will act as a facilitator while the students attempt shorter investigative tasks in the classroom and longer project work outside curriculum time.

There are other types of tasks, activities and pedagogies that seem to include some elements of investigation, e.g. project work (Cockcroft, 1982; Nielsen, Patronis & Skovsmose, 1999; Wolf, 1990), guided discovery (Bruner, 1961; Collins, 1988; Yeo, Hon & Cheng, 2006), playing mathematically-rich games (Ainley, 1988, 1990; van Oers, 1996; Yeo, 2007), real-life and workplace problems (Carraher & Schliemann, 2002; Moschkovich, 2002b; Skovsmose, 2002), and mathematical modelling (English, 2007; Kaiser & Sriraman, 2006; Lingefjärd & Meier, 2010). However, these are not relevant to the current research because the tasks used in the present study are the two types of open investigative tasks described earlier. Therefore, the review of literature on all these other types of tasks, activities and pedagogies will not be included in this thesis.

2.1.5 Empirical Studies on Mathematical Investigation

As educators use different terms to mean mathematical investigation, e.g. exploration (Cifarelli & Cai, 2004) and open problems (Orton & Frobisher, 1996), I have searched and reviewed many academic books, edited books, journals and conference proceedings that contain all these terminologies. From the references cited in these sources, I have searched and reviewed relevant references that contain any terminology related to problem solving and investigation, which in turn give rise to more references to search for. However, most empirical research in current literature turns out to be on mathematical problems and not investigation. On the other hand, most literature that contains the word ‘investigation’ turns out to be studies where the researchers (e.g. Kaur, 1995; Lampert & Ball, 1998) have referred to their own investigation of other mathematics education issues, or studies where the researchers (e.g. Maher, 2005) have applied the investigation process to mathematical problems, as discussed earlier in Section 2.1.2. In the end, it turns out that there are far fewer empirical studies on investigation, although there is quite a fair bit of theoretical research on investigation, such as those theoretical articles or books cited earlier in Sections 2.1.1 to 2.1.4. Empirical studies on investigation processes are rare.

This section will review different kinds of empirical studies on mathematical investigation from different countries: informal research, typical action research, ethnographic studies and teaching experiments. However, empirical studies that track the actual cognitive processes of students during investigation will be dealt with later in Section 2.2.4.

(a) Typical Informal Research Studies on Mathematical Investigation

Since mathematical investigation was introduced in the UK in the 1960s by the Association of Teachers of Mathematics (ATM) and officially affirmed with the publication of the Cockcroft Report in 1982 (Jaworski, 1994), one might expect that a good source of research in mathematical investigation would be their mathematics education journals. However, both journals, *Mathematics Teaching* and *Mathematics in School*, published by ATM and the Mathematics Association (MA) respectively, are essentially non-research journals. There are a lot of teaching ideas on investigation in these journals, but the claim that doing mathematical investigation benefits students is usually based on some anecdotal evidence (Lesh, Lovitts & Kelly, 2000).

A typical report (e.g. Bird, 1980) gave the background of the investigation, followed by a detailed account of the students' investigation and sometimes samples of their work. The focus was on the content of the investigation that the students managed or did not manage to discover. Some reports ended just like this but others (e.g. Davies, 1980) included a short reflection on the benefits of the investigation. Bishop (1982) claimed that at least half a dozen articles in the current issue in front of him (*Mathematics Teaching*, No. 98, March 1982) had "a clear research slant" (p. 65) and he was glad that the column headed *Research* in previous issues of the journal had stopped because the problem with labelling one article 'Research' implied that the other articles in the journal were not research studies. But the journal adopted the column headed *Research* again from No. 174, March 2001, onwards, suggesting that subsequent editors of the journal might not view the other articles in the journal as formal research. Nevertheless, the journal did have some more formal research studies which will be described next.

(b) Typical Action Research Studies on Mathematical Investigation

A study in the UK by Tanner (1989) was published in *Mathematics Teaching*. It described how he had conducted a research on the implementation of mathematical investigation in his school because of the arrival of the GCSE with its coursework component on investigation. They sought the help of ATM, which provided advice, moral support and financial assistance to Tanner and his staff to start an action research group with the aim of improving teacher performance in the implementation of problem solving and investigation. During the research, Tanner found that it was necessary to provide guidance to most students during investigation, but "the most effective interventions question students rather than dictate to them" (p. 22). He observed that group work and discussion helped the students to generate and test ideas, and to practise the communication of ideas which would force them to clarify and redefine their ideas if necessary. The outcome was that the teachers were more willing to accept and discuss unorthodox methods with their students and "the students no longer demand: What page? when the teacher enters the room as often as they used to" (ibid.). However, Tanner recognised that these findings were not based on systematic observations and testing, though he believed that they were nevertheless true for their classes because they were based on many hours of observations.

(c) Ethnographic Research Study on Open-Ended Approach by Boaler (1998)

Boaler (1998) conducted a three-year ethnographic research in the UK where one school used the traditional textbook approach, while the other school used open-ended activities at all times because it was involved in a small-scale pilot of a new GCSE examination that assessed process as well as content. Although Boaler did not use the term 'investigation', she referred to the project-based approach as process-based teaching with its emphasis on processes similar to those for investigation described in Section 2.1.2. Each project for the students in the experimental school lasted for two to three weeks. Examples of a starting point for projects were, "The volume of a shape is 216. What can it be?" or "What is the maximum sized fence that can be built out of 36 gates?" "The students were then encouraged to develop their own ideas, formulate and extend problems, and use their mathematics." (p. 49) If the students needed to use certain mathematics content or procedural skills that they did not know of, the teacher would simply teach it to them. Boaler found that the students had greater success in the new form of GCSE examination that rewarded problem solving as well as procedural knowledge (this exam was discontinued in 1994). This suggested that it was possible to teach students how to solve real-life and unfamiliar problems by adopting an entirely new open-ended project-based pedagogy in the classroom.

(d) Teaching Experiment by Lampert (1990)

In the USA, Lampert (1990) conducted a teaching experiment for fifth grade students in a public school for three years. A typical teaching unit consisted of several lessons for exploration of open tasks and mathematical discourse. For example, Lampert began by asking her class to prepare their own tables of squares from 1^2 to 100^2 using calculators and challenging them to find patterns in the tables. The students' most sophisticated conjecture, asserted towards the end of the lesson, was about the pattern of the last digit of the squares. They even proved that the pattern would go on forever using the underlying mathematical structure. Lampert then made use of the students' discovery of the pattern in the last digit of squares to extend the problem to "What is the last digit in 5^4? 6^4? 7^4?" and later to "What is the last digit of 7^5?" Throughout the whole activity, Lampert acted only as a facilitator to follow and engage in mathematical arguments of the students during the classroom discourse and she never told the students whether their answers were right or wrong. Instead, the students were expected to come to their own conclusions, based not on what the teacher, a better student or the textbook said, but based on mustering enough evidence to prove or disprove a conjecture. Lampert recognised that learning is "both the activity of acquiring knowledge and the knowledge that is acquired" (p. 59). What she described in her research was the activity of learning, but she admitted that it remained to be seen what knowledge her students had acquired and whether it was transferable to situations in the real world.

(e) Narrative Inquiry on Pre-service Teachers by Bailey (2007)

In New Zealand, Bailey (2007) used a research methodology called narrative inquiry to investigate her professional practice as a teacher educator using mathematical investigation to develop a cohort of about 200 pre-service primary teachers' understanding of what it means to teach and learn mathematics. She cited literature to support her claim that narrative inquiry, which is a form of storytelling with constant reflection, has become a valued form of research in recent years. Bailey admitted that the investigative approach was new to her, but as she taught the pre-service teachers using this approach, she herself also became a learner in mathematical investigation. Thus she was able to identify with her students the initial feelings of discomfort with an unfamiliar approach and being stuck in the process of investigation. In the end, she found that all four pre-service teachers, whom she interviewed, had experienced a deeper level of learning using the investigative approach.

(f) Research Study on Magic Squares by Ng (2003)

In Singapore, there were only two research studies on mathematical investigation. The first study was a Master's thesis by Ng (2003). He conducted a series of lessons on the investigation of magic squares over a period of six months for a Primary 6 Gifted Education Programme (GEP) class of 23 boys and six girls. The duration of the investigation varied from 15 minutes in normal classroom teaching to three hours in the computer laboratory, where his students used EXCEL to help sum up each row, column and diagonal of the magic squares. A total of 23 hours 10 minutes were spent on the investigation. Ng's role was to guide the students in the investigation and class discussions. The investigation started with constructing a 3×3 magic square and then extended to the construction of higher-order magic squares. Eight of the 16 sessions were video-taped and transcribed for analysis. Ng found that the investigation promoted flexibility and inventiveness in problem solving, provided opportunities for rich mathematical communication that helped students construct mathematical knowledge, created opportunities for students to persevere in problem solving, and triggered students to make their own mathematical conjectures.

(g) Research Study on Investigation and Mini Project Work by Ng, Teo and Leow (2005)

The second study on mathematical investigation in Singapore was conducted by another researcher and two teachers on investigation and mini project work (Ng, Teo & Leow, 2005). Two classes of Secondary 2 (equivalent to Grade 8) students were exposed to some investigative tasks during lessons, and they were also divided into groups to do a mini project. The students' class work and projects were assessed using a scoring rubric. The researcher (i.e. Ng) believed that investigative tasks should be used to practise and consolidate a skill or concept, and to follow up or extend a topic, and that project work should be used to provide connections between the different units in a topic through a concept map. She also viewed investigative tasks and mini project work as 'alternative assessment' to assess other aspects of students' learning, rather than to learn through mathematical investigation and project work.

(h) Summary of Literature Review on Empirical Studies

The literature review of empirical studies on mathematical investigation has revealed that there are very few such studies, and that most of these studies have focused on the general benefits of investigation, rather than the nature and development of processes in mathematical investigation.

2.1.6 Summary of Literature Review on Mathematical Investigation

The review of relevant literature on mathematical investigation in Section 2.1 has helped to reconcile conflicting views in existing literature between problem solving and investigation, and to clarify various constructs such as (i) the differences among investigation as a task, a process, and an activity, and (ii) the relationships between problem solving and mathematical investigation. This will help to define clearly the terms used in the present study. The literature review has also revealed a lack of empirical studies on the processes of investigation. Thus the present research will try to address this gap in current research by studying the nature and development of thinking processes in mathematical investigation.

2.2 LITERATURE REVIEW ON COGNITIVE PROCESSES

Since there is not much research on the processes of mathematical investigation as explained earlier in Section 2.1, the review of literature in this section will have to draw mostly from research studies on the processes in problem solving. This is justifiable because there is a great deal of overlap between the processes in problem solving and in mathematical investigation as explained in Sections 2.1.2 and 2.1.3. This section will begin by clarifying the construct of thinking processes used in research literature. It will then review models of cognitive processes in mathematical investigation and problem solving in order to understand the types and interactions of these processes, followed by the study of each of the eight main investigation processes in detail. Empirical studies of cognitive processes in both problem solving and mathematical investigation literature will be reviewed to set the direction of the present research study. Lastly, this section will end with a summary of the main findings on the nature and development of cognitive processes in current literature.

2.2.1 Thinking Processes

(a) Thinking

There are many types of thinking, e.g. analytical thinking, logical thinking, critical thinking, creative thinking, reasoning, higher order thinking, lower order thinking, lateral thinking, inductive thinking, deductive thinking and mathematical thinking. Krulik and Rudnick (1993) believed that there are different levels of thinking in some kind of hierarchical order (see Fig. 2.4). They acknowledged that the levels are not distinct but there are some overlaps. Nevertheless, each level makes use of the types of thinking contained in the levels that lie below it. At the lowest level, recall thinking means the automatic recall of knowledge and facts. The next level of thinking, basic thinking, includes the understanding and application of mathematical concepts. These two levels of thinking are considered lower order thinking, while the next two levels, critical and creative thinking, are higher order thinking. Critical thinking involves the skills to analyse and evaluate, while creative thinking is the ability to synthesise and generate new ideas. Krulik and Rudnick believed that reasoning entails the three highest levels of thinking: basic, critical and creative thinking.

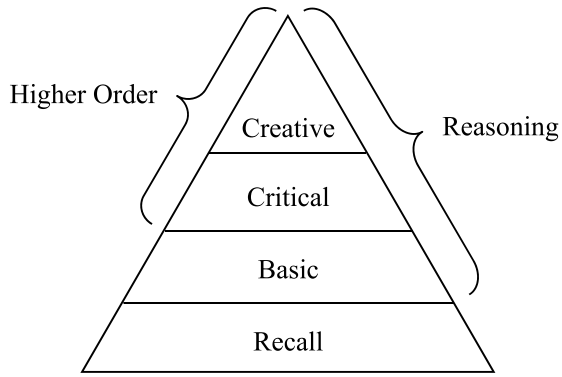


Figure 2.4 Model of Hierarchy of Thinking

Mathematical problem solving and investigation require not only basic and critical thinking, but also creative thinking because the "ability to integrate results of an investigation into an effective plan or solution to solve a problem" (Bloom, Engelhart, Furst, Hill & Krathwohl, 1956, p. 170) involves the idea of synthesising and generating new ideas. In fact, problem solving and investigation also require the lowest level of recall thinking as students need knowledge and facts to think. Schoenfeld (1985) called these 'resources' that are essential for problem solving.

(b) Processes

Backhouse, Haggarty, Pirie and Stratton (1992) defined a process as "a way of doing something, or a mode of action" (p. 90) and Shufelt (1983) viewed problem solving as a process where one *does* something to solve a problem. But a thinking process may not result in a physical action, e.g. a person may just reason something in his or her mind. Therefore, a thinking process is a mental process that one *does* in his or her mind which may not result in a physical action. However, there are two different perspectives of 'process' in investigation: from one angle, an investigation is one whole process (Frobisher, 1994), just like problem solving is one whole process (Shufelt, 1983); from another angle, there are many processes involved in an investigation (Frobisher, 1994). Frobisher classified these investigation processes into five categories:

(1) Communication Processes (e.g. explaining, talking, agreeing, questioning)
(2) Reasoning Processes (e.g. collecting, clarifying, analysing, understanding)
(3) Recording Processes (e.g. drawing, writing, listing, graphing)
(4) Operational Processes (e.g. collecting, sorting, ordering, changing)
(5) Mathematical Processes (e.g. pattern searching, conjecturing, proving, generalising).

The first four types of processes are general processes that are useful in mathematical investigation while the last type is what he called "mathematics-specific processes" (p. 161) or "processes 'unique' to mathematics" (p. 163). The problem with this definition of mathematical processes is that some of these processes are not unique to mathematics only, e.g. scientists also formulate and test conjectures in scientific experiments so as to generalise into scientific laws. On the other hand, some of the general processes also pertain to mathematics, e.g. analysing in reasoning processes is often associated with mathematical problem solving and investigation; and processes such as mathematical communication (Pimm, 1987) and mathematical writing (Morgan, 1998) are very much mathematical in nature. Therefore, there are grey areas in Frobisher's (1994) classification, but it may not be necessary to draw such a clear boundary between general processes and mathematical processes. Nevertheless, some of the processes are thinking processes (e.g. reasoning and mathematical processes) while others do not involve much thinking (e.g. recording processes). Other researchers, such as Mason et al. (1985), have also identified four main mathematical thinking processes (namely, specialising, conjecturing, justifying and generalising), which are similar to Frobisher's (1994) mathematical processes. These mathematical thinking processes will be discussed in greater detail later in Section 2.2.3.

There is another type of process that involves more than just thinking. It is "thinking about your own thinking" (Schoenfeld, 1987, p. 189) or metacognition. Although the Singapore school mathematics curriculum distinguishes between processes and metacognition in its Pentagon Model (see Section 1.1), the term 'processes' will be used in the present research study to include both cognitive and metacognitive processes. Sometimes, there is a need to use the phrase 'thinking processes' to emphasise the thinking behind these processes. When this term is used, it also refers to both cognitive and metacognitive processes although, strictly speaking, metacognition is a meta-thinking process and not a thinking process. The literature review on metacognition will be dealt with later in Section 2.3.

To summarise, processes can be thinking or non-thinking, and thinking processes can be cognitive or metacognitive. On the other hand, processes can also be mathematical or non-mathematical. Sometimes, it is not easy to classify a process neatly into a category because there are grey areas between the various categories of processes. The next section will focus on cognitive processes that are useful in investigation.

2.2.2 Models of Cognitive Processes in Mathematical Investigation

Clement (2000) believed that one of the most important current needs in basic research on thinking processes is the need for an insightful explanatory model of students' thinking processes. This type of explanatory models is often iconic in nature and the purpose of the model is to give satisfying explanations for patterns in observations (Lesh et al., 2000). As there are few such models on investigation in the current literature, existing models of cognitive processes in problem solving will also be examined, since most of these processes are similar for investigation and problem solving (see Section 2.1.2). The models in this section were chosen because (i) they have been formulated by renowned researchers (e.g. Pólya, 1957; Schoenfeld, 1985), (ii) they have been used frequently by other researchers (e.g. Pólya's model and Schoenfeld's model), or (iii) they have important elements different from the other models (e.g. Wallas' creativity model and Height's investigation model).

(a) Problem-Solving Model by Pólya (1957)

George Pólya began a revolution in problem solving with the publication of his book *How to Solve It* in 1945 and a revised edition in 1957. There are four phases in his problem-solving model (Pólya, 1957):

(1) Understanding the problem
(2) Devising a plan
(3) Carrying out the plan
(4) Looking back.

When students first encounter a problem, they should make sense of the problem by looking for the given information, visualising the information and organising it. They should then devise a plan to solve the problem by using various heuristics, such as drawing a diagram, making a systematic list and looking for patterns. They should then implement the heuristic chosen to solve the problem. If they still fail to solve it, they will have to go back to either the first phase to see if they have left out any given information, or to the second phase to think of another plan. This suggests that problem solving is not a linear process. If the students manage to solve the problem, they should check the solution, and try to improve on the method used. They should also reflect on whether the result or the method can be applied to other problems.

(b) Problem-Solving Model by Mason, Burton and Stacey (1985)

Mason, Burton and Stacey (1985) formulated a problem-solving model (which will be called Mason's model) with three phases that include the four main mathematical thinking processes:

(1) Entry
(2) Attack (including specialising, conjecturing, justifying and generalising)
(3) Review.

During the Entry Phase, the students should try to understand the problem, which is similar to the first phase of Pólya's model. The major activity in Mason's model is the Attack Phase, which is similar to a combination of the second and third phases of Pólya's model, except that Mason et al. included the four main mathematical thinking processes: specialising, conjecturing, justifying and generalising. They believed that specialising and generalising are two sides of the same coin: the purpose of trying examples (specialising) is to generalise. However, a general result thus obtained might not be true: it is only a conjecture to be proven or justified. The four main mathematical thinking processes will be described in more detail in Section 2.2.3. The last phase of Mason's model, the Review Phase, is similar to Pólya's fourth phase.

(c) Mathematical Discovery Model by Lakatos (1976)

Lakatos (1976) discussed the development of mathematical knowledge through discovery, which results in proving or disproving conjectures, but he did not present a schematic representation of the discovery process. Davis and Hersh (1981) decided to simplify Lakatos' mathematical discovery in the form of a model (see Fig. 2.5). After making a conjecture, there are two possibilities: (i) students can do some naïve testing using empirical data to see if they can refute the conjecture; if the conjecture is not refuted, the students can proceed to prove the conjecture; (ii) students can go straight into proving the conjecture without doing naïve testing; during the proving, refutation may also occur. If refutation happens, the conjecture needs to be reformulated. Again, there are two possibilities: (i) if the refutation is the result of a local counter example, then the conjecture needs to be modified or revised; (ii) if the refutation is the result of a global counter example, then the conjecture is false and a new conjecture will have to be formulated. The model shows that mathematical discovery is not linear.

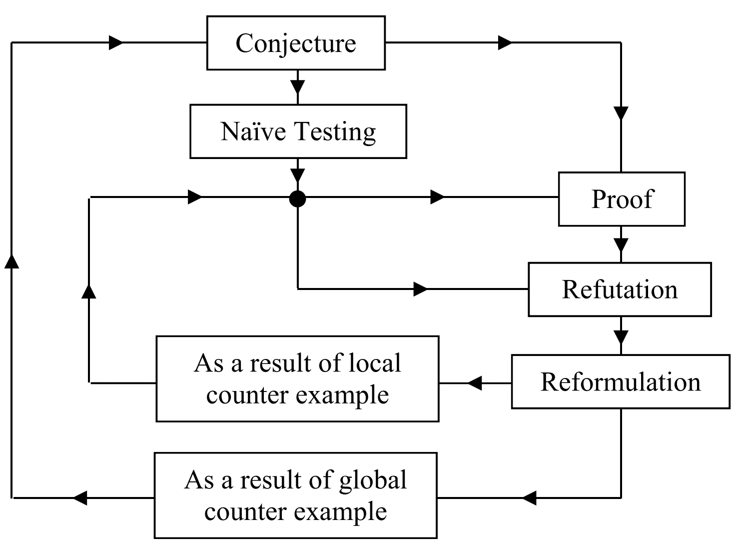


Figure 2.5 Simplified Lakatos' Mathematical Discovery Model

(d) Problem-Solving Model by Schoenfeld (1985)

Another type of problem-solving models, which is not about the problem-solving process but the components essential to solve problems successfully, will be examined. Schoenfeld (1985) realised that having a wide repertoire of heuristics, as described in Pólya's model, is not enough to make one student a better problem solver than another. So he identified four categories necessary for effective problem solving:

(1) Resources
(2) Heuristics
(3) Control
(4) Belief Systems.

To solve a problem, students need to use basic mathematical knowledge, which Schoenfeld called 'resources', because thinking does not exist in a vacuum. Students also need to be familiar with a broad range of heuristics. However, some students are more apt to select and deploy the resources and heuristics at their disposal than others. This is the issue of control or metacognition, which will be discussed in detail in Section 2.3. Whether students will persevere to solve the problem depends very much on their belief systems. If the students think that a mathematical problem is something that can be solved within 10 minutes, then they might give up if they cannot solve the problem within that time frame. But what happens if the students still fail to devise a suitable plan to solve a problem? This question will be answered by looking at the next model.

(e) Creativity Model by Wallas (1926)

The 'synthesis' category in the Bloom Taxonomy (Bloom et al., 1956) includes the "ability to integrate results of an investigation into an effective plan or solution to solve a problem" (p. 170). Since this ability to synthesise a plan to solve a problem is not purely an analytical skill but a synthetic or creative one as well (Lenchner, 1983), some literature on creativity will be reviewed to see what can be learnt from it. Wallas (1926, as cited in Pope, 2005) proposed that there are four stages of creativity:

(1) Preparation

(2) Incubation

(3) Illumination

(4) Verification.

To be creative, students need to prepare by arming themselves with knowledge, and in the case of problem solving, heuristics as well. Then students need time to think. This is called the incubation period where the students take a break and relax their minds: think about the problem in a more relaxed state and environment and let the images from the subconscious surface. This is a useful strategy when students fail to devise a suitable plan to solve a problem or are stuck. For example, Einstein said that he got some of his best ideas while shaving (Fabian, 1990), and Archimedes discovered a way to find the volume of an irregular object, such as the King's crown, while taking a bath (The MacTutor History of Mathematics Archive, 1999). The incubation stage is crucial for illumination. The problem with modern society is that there are so many things to do that students have very little time to incubate. The only stage in creativity that the students have no control over is illumination: whether an idea comes to them does not depend on the students themselves. They may have prepared themselves well with resources and heuristics, and have plenty of time to incubate, but they still may not reach the illumination stage. If they do, the last stage is to fill in the details and test out the idea to make sure that it really works. This is rather similar to Pólya's third and fourth phases of carrying out the plan and looking back. The above review suggests that students should undergo the creative incubation process if they are stuck during problem solving.

(f) Multi-Cyclic Mathematical Investigation Model by Height (1989)

Since investigation, like problem solving, is not a linear process (Burton, 1984), Height (1989) proposed a multi-cyclic strategy for investigation consisting of eight stages (see Fig. 2.6). In the first stage, students need to understand the investigative task and set up a situation to investigate by defining an initial model of the problem. The second stage involves collecting information about the problem, e.g. trying out specific examples, which is similar to the process of specialising in Mason's model. The information is then organised so that the students can search for patterns (third stage) which may lead to the formulation of a hypothesis (fourth stage). In the fifth stage, the testing of a hypothesis is based on empirical data, which is similar to the naïve testing of a conjecture in Lakatos' model. In the sixth stage, the initial model of the problem is refined or extended to increase its boundary, which might lead to more pattern searching and so the cycle continues. In the seventh stage, the students will try to justify the hypothesis formed using a more rigorous proof, and in the last stage, the presentation of a written report. Height (1989) described his model as multi-cyclic, meaning that students might go through the cycle many times. His model is one of the few models that has a cycle and an end. Most models are either cyclic without an end (e.g. Lakatos' model), or linear with an end (e.g. Frobisher's model in next part).

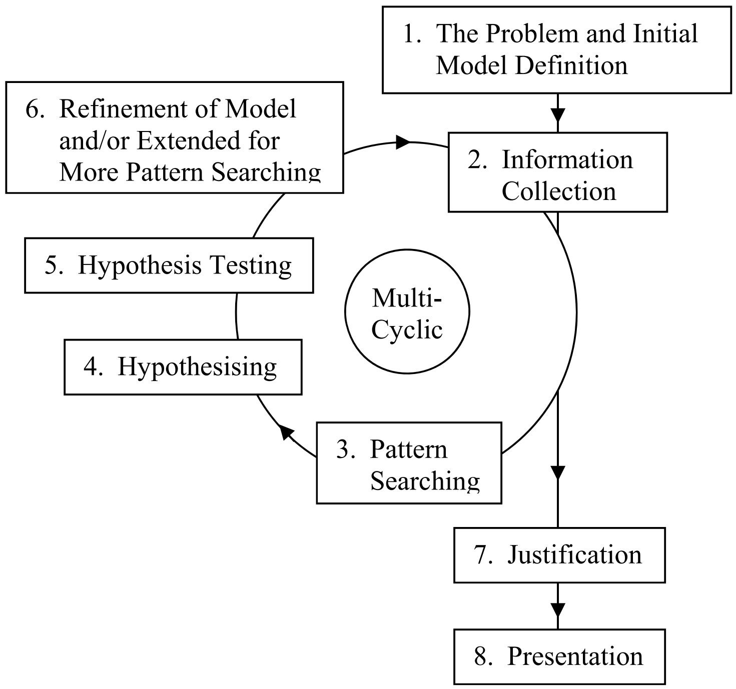


Figure 2.6 Height's Multi-Cyclic Mathematical Investigation Model

(g) Mathematical Investigation Model by Frobisher (1994)

Frobisher (1994) believed that mathematical processes interact with one another when students engage in investigation and he suggested a model for their interactions (see Fig. 2.7). Some of these processes, such as pattern searching, conjecturing and proving, are similar to those advocated in other models described earlier. However, Frobisher distinguished between conjecturing and hypothesising, but he did not explain the difference. Cambridge Dictionaries Online (Cambridge University Press, 2012) defined a conjecture to be 'a guess about something based on what it *seems* [emphasis mine]' but a hypothesis is 'an idea or explanation for something that is based on known *facts* [emphasis mine] but has not yet been proven'. So conjecturing comes first, and then after testing a conjecture based on empirical data (which is similar to naïve testing of a conjecture in Lakatos' model), the conjecture will become a hypothesis based on more reliable facts. According to Frobisher's model, the hypothesis is then subject to further testing. However, most models do not include so many rounds of testing, and they either used the term 'conjecture' (e.g. Lakatos' model) or 'hypothesis' (e.g. Height's model). Since most researchers (e.g. Lampert, 1990; Mason et al., 1985) do not distinguish between a conjecture and a hypothesis, this thesis will use only one term: conjecture.

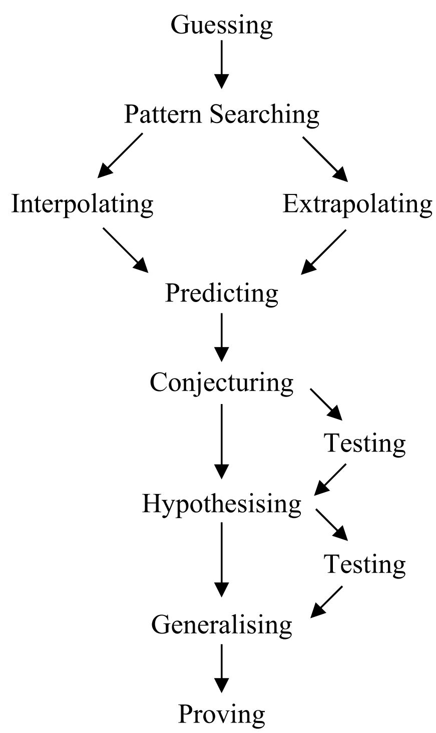


Figure 2.7 Frobisher's Mathematical Investigation Model

(h) Summary of Literature Review on Models of Cognitive Processes

To summarise, there are generally three phases in a model of cognitive processes in mathematical investigation: Entry, Attack and Review. The stages in the Entry Phase include the main processes of understanding the task and problem posing, while the stages in the Attack Phase consist of the four main mathematical thinking processes:

specialising, conjecturing, justifying and generalising. The issue of how problem-solving heuristics fit into the Attack Phase will be discussed later in part (c). The Review Phase consists of the main processes of checking and extension. Then the cycle repeats. Although each stage of the investigation model is linked to a main cognitive process, there are various sub-processes within each stage or main process. Thus there is a need to examine these sub-processes in the next section.

2.2.3 Processes and Outcomes in Mathematical Investigation

Table 2.1 shows the three phases and eight stages of mathematical investigation as explained in the previous paragraph. Each stage is named after the main process(es) that it contains. In general, each stage contains one main process, except for the stage of Specialising and Using Other Heuristics (S/H) which contains two main processes. Since a main process may contain some smaller processes, these smaller processes will be called sub-processes in this thesis. In other words, a process can be a main process or a sub-process. Moreover, it is found necessary to include outcomes when studying processes in mathematical investigation because the outcomes might affect how the processes interact with one another and vice versa. Examples of outcomes are: discovered a pattern and formulated a conjecture.

Table 2.1 Stages of Mathematical Investigation

Phases	Stages / Main Processes
Entry	Stage 1: Understanding the Task (U)
	Stage 2: Problem Posing (P)
Attack	Stage 3: Specialising and Using Other Heuristics (S/H)
	Stage 4: Conjecturing (C)
	Stage 5: Justifying (J)
	Stage 6: Generalising (G)
Review	Stage 7: Checking (R)
	Stage 8: Extension (E)

This section will review literature on the processes and outcomes in the eight stages and apply them to the two types of investigative tasks (Type A and Type B) discussed earlier in Section 2.1.4. An example of each type of investigative tasks is given below. These are the same tasks used in the posttest for the present study.

Task 7 (Type A): Add Sum of Digits to Number (or Kaprekar Sequences[1])

Choose any number. Add the sum of its digits to the number itself to obtain a new number. Repeat this process for the new number and so forth. Investigate.

Task 8 (Type B): Sausages

I need to cut 12 identical sausages so that I can share them equally among 18 people. Investigate.

(a) Stage 1: Understanding the Task

The first stage in mathematical investigation is the main process of Understanding the Task (U) in the *Entry Phase*. Pólya (1957) advocated asking questions such as, "What is the unknown? What are the data? What is the condition?" (p. xvi) To do these, the students might have to re-read the task a few times, paraphrase the task statement, or highlight key information. Pólya also encouraged visualising the given information by drawing a diagram. On the other hand, Mason et al. (1985) suggested the use of a few random examples to make sense of the task. Although Mason et al. labelled this sub-process of trying examples to understand the task as the first type of specialising, this thesis will *not* classify this sub-process under specialising because there is a need to distinguish between trying examples in the first stage of Understanding the Task, and

[1] The students are not expected to know that the sequences generated are called 'Kaprekar Sequences', but this term will be used in this thesis for ease of discussion.

trying examples to search for patterns in the third stage of Specialising, which will be discussed later in part (e). For Type A tasks such as Task 7 (Kaprekar), students are more likely to try some examples to understand the task than for Type B tasks such as Task 8 (Sausage). For the latter, students may engage in the sub-process of drawing a diagram to visualise the information. The outcomes in this stage are (i) understood the task correctly, (ii) misinterpreted the task and did not recover, and (iii) misinterpreted the task but recovered from the misinterpretation.

(b) Stage 2: Problem Posing

The second stage in mathematical investigation is the main process of Problem Posing (P) or Goal Setting in the *Entry Phase*. As explained earlier in Section 2.1.4, students can just search for any pattern in Type A tasks without posing any specific problem to solve, and this is called the posing of the *general* problem, "Is there any pattern? If yes, what is the pattern?" From the perspective of goal setting, this is called setting the *general* goal to search for any pattern. For Task B tasks, there is a need to set a *specific* goal by posing a *specific* problem to solve. To use Task 8 (Sausage) as an illustration, students usually cannot just search for any pattern but they need to pose specific problems such as:

- How do I cut 12 identical sausages so that I can share them equally among 18 people?
- How many cuts will there be?
- Is there a least number of cuts? If yes, what is it? And how do I cut the sausages so that I can get the least number of cuts?

In other words, there are two possible outcomes in problem posing: pose the general problem of searching for any pattern, or pose a specific problem. The main issues to address next are to identify the different types of sub-processes in problem posing and to discuss the quality of problems posed by examining existing literature.

Types of Sub-Processes in Problem Posing

Silver (1994) observed that "[problem] posing can occur before, during, or after the solution of a problem" (p. 19). For investigative tasks, problem posing occurs even before the students start to think of a solution because they need to pose their own problems to solve (Cai & Cifarelli, 2005). This is the first type of problem posing. The second type occurs as a problem-solving heuristic when a difficult problem is reformulated as a related problem which can be solved more easily, or as a smaller related problem to solve, before attempting to solve the original difficult problem (Pólya, 1957). This can happen before or during the solution of the original difficult problem. The third type of problem posing is similar to the first type except that it occurs at the end of the solution of a problem when the students pose more problems to solve or extend the original task by changing the given (Orton & Frobisher, 1996). Brown and Walter (2005) discussed two categories of problem posing. The first category is to accept the given in the task statement while the second category is to change the given in the task statement. The latter will be dealt with in Section 2.2.3(h) under extension. In this section, only the first category will be discussed. Brown and Walter have collected a long list of problems generated in one of their classes, some of which are given below:

- Is there a pattern? If yes, what is the pattern?
- What does it remind you of?
- Is there a formula? If yes, what is the formula? (pp. 30-31)

The next problem is to classify these questions into different types of problem-posing sub-processes. The first type is suggested by the following question from the list above, “Is there a pattern? If yes, what is the pattern?” This will be called the sub-process of searching for any pattern in this thesis. The second type of problem-posing sub-processes is suggested by the following question from the list above, “What does it remind you of?” Kilpatrick (1987) used the term ‘association’ to describe this type of problem posing. He cited cognitive scientists (e.g. Calfee, 1981) who proposed a variety of models of human knowledge which tend to portray knowledge as a hierarchical network of associated ideas, and in particular, Novak and Gowin (1984) who have used concept maps to represent the organisation of concepts. Thus students might associate some key elements in the task statement to other concepts in their minds in order to pose a problem to solve, or they may use a discovered result as a springboard for more problems to pose. The third type of problem-posing sub-processes is suggested by the following question from the list above, “Is there a formula? If yes, what is the formula?” Kilpatrick (1987) described this type of problem posing as ‘generalisation’. Mason et al. (1985) also discussed posing problems to generalise by examining specific examples (specialising).

There is a fourth type of problem-posing sub-processes. Krutetskii (1976) argued that there is a problem that ‘naturally follows’ from the given task. The most common problem that ‘naturally follows’ from Type A tasks is the general problem of searching for any pattern. What is of interest are the types of specific problems that 'naturally follow' from Type B tasks. The following shows some examples of specific problems for Task 8 (Sausage):

- How do I cut the 12 identical sausages so that I can share them equally among the 18 people?
- How many cuts will there be?
- Is there a least number of cuts? If yes, what is it? And how do I cut the sausages so that I can get the least number of cuts?
- Can I extend all the above problems to generalise for *n* identical sausages and *m* people?

Although the above problems are in some kind of logical order, it does not mean that the students must pose these problems in this order. For example, a student might pose the third problem of finding the least number of cuts without posing the first two problems; and if most students do that, then the third problem will become the 'most natural problem' among the first three problems for students to pose. Krutetskii (1976) found that high-achieving students were able to pose this kind of problems that 'naturally follow' from the given task, but low-achieving students were not able to do so unless they were given hints. We have discussed in Section 2.1.4 how to obtain Type B tasks by removing the original intended problem from mathematical problems (Frobisher, 1994). In fact, Task 8 was obtained from a mathematical problem from Mason et al. (1985), and the original intended problem, "Find the least number of cuts", is the third problem listed above. In other words, the original intended problem is supposed to be a problem that should 'naturally follow' from the task.

Quality of Problems Posed

The next issue to discuss is the quality of the problems posed. Although there are no right or wrong problems posed, Brown and Walter (2005) discovered that those "who have not studied the subject, or perhaps consider themselves 'weak' in mathematics, tend to come up with more robust questions" (p. 22) than those who have most recently been exposed to the subject, and they suggested more research on "the nature of the mathematical mind of … a good problem poser" (p. 170). They cautioned against applying some questions in certain circumstances because they may not be appropriate, but on the other hand, "a nonsensical-sounding question might apply if we were willing to modify what might be our own rigid mindset" (p. 32). Mason et al. (1985) observed that "questions that I know I can answer are generally less interesting than questions I am unsure about" (p. 143), thus differentiating problems that are more interesting from those that are less interesting. On either end, it may be easy to agree that some problems posed are more interesting or of a higher quality, while others are less interesting or of a lesser quality. But in the middle, there will always be grey areas. The important issue is to look at inter-coder reliability when deciding on the quality of a problem posed (see Section 5.4).

Mason et al. (1985) also discussed the problem of students posing only questions that they can answer easily. In traditional classrooms that look down upon getting stuck on a problem and demand the correct answer, many students may think that it is better to pose an easier problem that they can solve than to pose a difficult problem that they cannot solve. That is why Mason et al. kept emphasising that "being stuck is honourable" (p. 144) and that "it cannot be avoided, and it should not be hidden" (p. 49). During the developing lessons in the present research study, great care will be taken to convince students that it is alright to pose problems that they cannot solve, and that they should refrain from posing *only* the easier problems that they can solve although they can still pose such problems (see Section 3.6.3b). But it does not mean that they should pursue the difficult problems that they have posed because the solutions might be beyond their abilities. This issue will be addressed in Section 3.2.2 under a metacognitive process called analysing the feasibility of the goal or problem.

(c) Stage 3: Specialising and Using Other Heuristics

The third stage in mathematical investigation is Specialising and Using Other Heuristics (S/H), which occurs at the start of the *Attack Phase* or problem solving process. As explained in Section 2.1.2(d), there are generally two approaches to solving a problem: (i) the inductive approach which involves specialising (which is the same as the investigation process), and (ii) the deductive approach which uses other heuristics such as reasoning. This is the only stage with two main processes: S and H.

Type A tasks, such as Task 7 (Kaprekar), generally require students to start attacking the problem by trying examples to search for any pattern, which is the inductive approach. The trying of specific examples or special cases is called specialising, which is one of the four main mathematical thinking processes (Mason et al., 1985). However, specialising is also a problem-solving heuristic (Pólya, 1957; Schoenfeld, 1985). Type B tasks, such as Task 8 (Sausage), usually require students to solve by using other heuristics such as reasoning, which is the deductive approach. For example, in Task 8, students can find a method to share the sausages by using some simple reasoning:

> "To share 12 sausages equally among 18 people, each person will receive 12/18 = 2/3 of a sausage. This means that if I cut each sausage into 3 parts, then it is possible for each person to receive 2 parts of a sausage. Thus I am able to share the 12 sausages equally among 18 people by cutting each sausage into 3 parts."

This kind of reasoning does not involve trying examples (specialising) to look for patterns. Thus the term 'other heuristics' will be used in this thesis to refer to heuristics that do not involve specialising. However, it is possible to solve a specific problem posed for a Type B task by specialising. For example, for Task 5 (Investigate Handshakes) discussed earlier in Sections 2.1.2(d) and 2.1.4(b), students can pose the specific problem of finding the total number of handshakes, and then solve it by examining the number of handshakes for smaller numbers of participants in order to find a pattern to generalise. Sometimes, students may use both specialising and other heuristics. For example, if they use other heuristics such as 'establishing subgoals' or 'reformulating the original problem posed as a simpler problem', then it still depends on what the students do with the subgoal or the simpler problem. If the students examine specific examples for the subgoal or the simpler problem, then the investigation pathway will be from the main process of other heuristics to the main process of specialising. However, if the students use reasoning to solve the subgoal or the simpler problem, then it will still be using other heuristics without specialising.

Role of Problem-Solving Heuristics in Mathematical Investigation

The main issue to address next is the role of problem-solving heuristics in investigation. Heuristics are often associated with methods to *solve* problems rather than methods to *discover* patterns or mathematical structures in investigation. Pólya (1957) explained that 'heuristic' is the name of a certain branch of study on "the methods and rules of discovery and invention" (p. 112), and the word "heuristic, as an adjective, means 'serving to discover'" (p. 113). In fact, Cambridge Dictionaries Online (Cambridge University Press, 2012) defined the word 'heuristic' as 'a method of teaching allowing students to learn by discovering things themselves and learning from their own experiences rather than by telling them things'. Pólya (1957) then described what he called 'modern heuristic' which seeks to understand the mental operations typically useful in the process of solving problems. But how is this modern heuristic, which is about processes involved in problem solving, related to the old meaning of heuristic, which is about discovery?

Davis and Hersh (1981) discussed Pólya's (1957) heuristics in problem solving together with Pólya's (1962; new edition in 1981) mathematical discovery without differentiating between them. Lakatos (1976) discussed proofs and refutations in problem solving and yet he said that this is the logic of mathematical discovery. One viewpoint of problem solving is just to solve problems, but Pólya (1957) and Lakatos (1976) looked at it from the viewpoint of discovering. Pólya (1957) phrased this very clearly when he described what he called 'heuristic reasoning' as "reasoning not regarded as final and strict but as provisional and plausible only, whose purpose is to *discover* [emphasis mine] the solution of the present problem" (p. 113).

If problem solving is viewed as just solving problems, then the thinking processes are analytical. But if problem solving is viewed as discovery, then it will involve creative thinking processes, such as illumination and insight (Lakatos, 1976). In fact, creativity in problem solving has already been discussed in Section 2.2.2(e), where the "ability to integrate results of an investigation into an effective plan or solution to solve a problem" (Bloom et al., 1956, p. 170) is classified under the 'synthesis' category in the Bloom Taxonomy. Synthesis means to put together or create. Since problem-solving heuristics can be viewed from the perspective of mathematical discovery, then heuristics do play a big role in investigation, which is about discovering underlying patterns or mathematical structures.

(d) Stage 4: Conjecturing

The fourth stage in mathematical investigation is Conjecturing (C), which is also one of the four main mathematical thinking processes (Mason et al., 1985). Conjecturing is to make a statement, called a conjecture, which appears reasonable but whose truth has not been established. To use Task 7 (Kaprekar) as an example, students may search for patterns and observe that ‘all the sequences are increasing’. This observed pattern is only a conjecture to be proven or refuted. This thesis distinguishes between the term ‘observed pattern’ which is a pattern that has been observed but may not be true, and the term ‘underlying pattern’ which is the actual pattern. Therefore, the conjecturing stage contains the sub-process ‘searching for patterns’ and the outcome ‘formulated conjecture’. On the other hand, conjectures can also be formulated from using other heuristics such as reasoning. Using Task 8 (Sausage) as an illustration, students can reason as follows:

> “To share 12 sausages equally among 18 people, each person will receive 12/18 = 2/3 of a sausage. If I cut each sausage into 18 parts, then I will need to make $12 \times 17 = 204$ cuts. But if I cut each sausage into 3 parts so that each person will receive 2 parts, then I will only need to make $12 \times 2 = 24$ cuts. Therefore, the least number of cuts is 24.”

This is only a conjecture since the students have not proven it. In fact, this conjecture is false. On the other hand, it is possible to solve a problem posed during investigation by using other heuristics without formulating any conjecture. For example, for Task 8, a student may pose the problem, "How do I cut the 12 identical sausages so that I can share them equally among the 18 people?" and then solve it by using reasoning (see solution in part (d) earlier), which is rigorous enough. Thus it is possible to bypass the conjecturing stage in investigation, depending on the nature of the problems posed.

(e) Stage 5: Justifying

The fifth stage in mathematical investigation is Justifying (J), which refers to convincing yourself and others that your conjecture is true (Mason et al., 1985). But you will not know beforehand whether a conjecture is true or false. That is why other educators (e.g. Bastow et al., 1991; Frobisher, 1994) prefer to use the term 'testing of conjecture' instead of 'justifying of conjecture'. However, "justification … is at the heart of a true mathematical ethos" (Hatch, 2002, p. 138) and Mason et al. (1985) even put it as one of the four main mathematical thinking processes. Thus this thesis will use the term 'justifying' to refer to the process of testing conjectures. If a conjecture turns out to be false, then it is refuted; if it turns out to be true, then it is proven or justified. The word 'justified' refers to the outcome of a conjecture being proven correct, while the term 'justifying' refers to the process of testing a conjecture which might not lead to justification in the end if the conjecture turns out to be false.

According to Lakatos' (1976) mathematical discovery model described in Section 2.2.2(c), one way to test a conjecture is called naïve testing: examining empirical data to see if the conjecture can be refuted first. If the conjecture can withstand the test of empirical data, it does *not* mean that it is proven or justified: there is still a need to prove the conjecture. There are generally two methods of proving. The first method is to use a non-proof argument (Mason et al., 1985; Stylianides, 2008) while the second method is to use a formal proof, e.g. one that involves algebra (DeMarois, McGowen & Whitkanack, 1996). To illustrate the difference between the two methods of proofs, let us consider Task 7 (Kaprekar). The students might observe that the sequences generated are all increasing. This is only a conjecture based on the observed pattern because the observed pattern might not continue. The following shows two methods to prove this conjecture:

> *Non-Proof Argument:* Assume that the first term of a Kaprekar sequence is positive[2]. Since the digits of the first term in a Kaprekar sequence are always non-negative and at least one of the digits is always positive (namely, the first digit), then the sum of the digits of a term will always be positive. Thus the next term, which is the preceding term plus the sum of its digits, will always be greater than the preceding term, so every Kaprekar sequence is increasing.

> *Formal Proof:* Assume that the first term of a Kaprekar sequence is positive. Let the first term in a Kaprekar sequence be x and its digits be a_1, a_2, a_3,… a_n, where a_i is the i-th digit of the term. Then $a_1 > 0$ and $a_i \geq 0$ for all $i \geq 2$. Thus the next term $x + a_1 + a_2 + a_3 + \ldots + a_n$ is always greater than x since $a_1 > 0$ and $a_i \geq 0$ for all $i \geq 2$. Therefore, the next term will always be greater than the preceding term, so every Kaprekar sequence is increasing.

Tension between Formal Proof and Non-proof Argument

Some researchers (e.g. Mason et al., 1985; Stylianides, 2008) believe that it is good enough to justify a conjecture using a non-proof argument, while others (e.g. Holding, 1991; Tall, 1991) insist on a formal proof. There are two approaches to a formal proof. The first approach is what Lakatos (1976) called the deductivist approach

[2] This is the usual definition of a Kaprekar sequence: the first term is positive. If the first term is 0, every term in the sequence will just be equal to 0. If the first term is negative, there is a slight complication, which could be investigated at a later stage.

where formal proofs are presented out of the blue. Pólya (1957) and Lakatos (1976) were against such proofs that come out of nowhere because it seems impossible that anyone should ever have guessed them. They advocated another approach to formal proofs. Pólya (1957) called this "heuristic reasoning [which] is often based on induction, or on analogy" (p. 113). He argued that "we need heuristic reasoning when we construct a strict proof as we need scaffolding when we erect a building" (ibid.). He strongly believed that it is bad to confuse heuristic reasoning with rigorous proof, but it is even "worse to sell heuristic reasoning for rigorous proof" (ibid.). As explained in Section 2.1.2(c), their idea of a heuristic approach to formal proofs is similar to an inductive approach using investigation processes, such as specialising, conjecturing, justifying and generalising. Nevertheless, some educators (e.g. Tall, 1991) would still prefer the rigour found in a formal proof, though such a proof involving algebra[3] might be beyond the level of most lower secondary school students. To summarise, the justifying stage contains a few sub-processes:

- Naïve Testing: examine empirical data to refute (not prove) a conjecture by counter examples
- Justify by Non-proof Argument
- Justify by Formal Proof: use algebra to justify a conjecture.

On the other hand, as explained earlier in part (d) of this section, it is possible to solve a problem posed during investigation by using other heuristics, such as reasoning, without formulating any conjecture to justify. Since reasoning appears in various stages or sub-processes in mathematical investigation, it is important to distinguish between them by using different terminologies:

[3] Axiomatic proofs are definitely beyond the level of secondary school students.

- Deductive reasoning refers to the use of reasoning (other heuristics) to solve a problem, with or without formulating any conjecture;
- Justify by non-proof argument refers to the use of reasoning in a non-proof argument to justify a conjecture;
- Justify by formal proof refers to the use of a more formal kind of reasoning, such as algebra, to justify a conjecture.

Another important issue to clarify is the sub-process of trying examples because it could occur in three different investigation stages:

- Understanding Stage: try examples randomly to make sense of the task;
- Specialising Stage: try examples systematically to search for patterns;
- Justifying Stage: try examples to refute a conjecture (naïve testing).

Some interesting issues for the present study to examine are: (i) Do students justify their conjectures, or do they wrongly accept their observed patterns as true without testing or based on naïve testing? (ii) Do students justify their conjectures by using a non-proof argument or a formal proof? These issues will be addressed in the data analysis in Sections 7.2.5, 7.3.5 and 7.3.9.

(f) Stage 6: Generalising

The sixth stage in mathematical investigation is Generalising (G), which is also one of the four main mathematical thinking processes advocated by Mason et al. (1985). They explained that generalising means "detecting a pattern leading to WHAT seems likely to be true (a conjecture); WHY it is likely to be true (a justification); WHERE it is likely to be true, that is, a more general setting of the question (another question!)" (p. 24). Thus there are three distinct features of generalising. The first one occurs when a pattern is detected leading to a conjecture, which is actually the process of conjecturing. The problem with this kind of 'general pattern' is that it may be false. This then leads to the second feature: the process of justifying the 'general pattern'. If justification is successful, then the 'general pattern' is the actual pattern. In other words, 'generalisation' only occurs when the 'general pattern' has been justified. There are also different levels of generalising because a 'general pattern' may just be a special case in a more general setting, and so a 'general pattern' can act as a springboard for further generalising, which is the third feature. However, this definition of generalising overlaps with conjecturing and justifying.

Wheeler (1983) described generalisation as 'mathematisation' which is "the process by which mathematics is brought into being" (p. 290), or 'structuration' which is "the act of putting a structure onto a structure" (Wheeler, 1982, p. 47) because the whole idea of generalisation is to find the underlying mathematical structure or pattern. Jaworski (1994) believed that "investigation [is] just a vehicle for other learning … this other learning might be seen as learning to be mathematical" (p. 4) and she explained that the idea of 'learning to be mathematical' is the same as what Wheeler (1983) has called 'mathematisation'. Many educators believe that generalisation or structuration is very important in mathematics. For example, Mason et al. (1985) claimed that "generalisations are the life-blood of mathematics" (p. 8).

An interesting question is whether it is possible to generalise without specialising. For example, Krutetskii (1976) wrote about a gifted 11-year-old who could generalise from one instance in mathematics. When asked to represent the general form of numbers that leave a remainder of 5 when divided by 7, the child argued from the *even more general case* of numbers that leave a remainder of z when divided by y. The child was able to see that the numbers must be of the form $xy + z$, and so for the given case, the answer is $7x + 5$. Thus it is possible for some gifted children to generalise without specialising. In this case, the child solved the problem using other heuristics, such as reasoning, without formulating and justifying any conjecture, and yet produced a generalised result at the end.

To summarise, since the process of generalising overlaps with the conjecturing and justifying processes, in order not to confuse the three processes, the thesis will use the term 'generalisation' to describe the outcome when a conjecture has been justified and led to a general result. Sometimes, the justification of a conjecture may not lead to a generalisation. For example, in Task 8 (Sausage), the proving of the conjecture that 'the least number of cuts for sharing 12 sausages among 18 people is 12' will not lead to a generalisation. On the other hand, it is possible to solve a problem using other heuristics, such as reasoning, which leads to a generalisation without going through specialising, conjecturing and justifying, as the gifted child described in the preceding paragraph has demonstrated. Therefore, in the generalising stage, which is the last stage in the *Attack Phase*, the issue is whether the final outcome is a generalisation.

(g) Stage 7: Checking

The seventh stage in mathematical investigation is Checking, which happens at the start of the *Review Phase*. Checking is denoted by R (R for review), and not C since C has been used for the Conjecturing stage. Many educators (e.g. Pólya, 1957; Mason et al., 1985) believe in the importance of checking the solution. Students can check all the working step by step, or they can just check the essential steps. Very often, students might just glance through the solution without really checking it thoroughly. A caveat is necessary here. Students can check their working (or, in fact, should check their working occasionally) in the previous stages even before finishing solving a problem. But the Checking stage in the *Review Phase* is for after the students have finished solving a problem. T higher level of reviewing the solution to see if the goal of the task is met is dealt with under metacognition in Section 2.3.2.

(h) Stage 8: Extension

The eighth and last stage in mathematical investigation is Extension (E). The idea of extension is to change the given in order to pose more problems to solve (Orton & Frobisher, 1996). This stage is similar to the second stage of problem posing, except that the second stage does not involve changing the given while the last stage of extension requires the changing of the given. The first category of problem posing proposed by Brown and Walter (2005), which is to accept the given, has been examined in Section 2.2.3(b) on the second stage. This section will discuss the second category of problem posing which is to challenge or change the given using the 'What-If-Not' strategy. Brown and Walter explained the rationale for using the strong word 'challenge' instead of just 'change' because the given is sometimes accepted as the truth when in fact it is not. For example, for about two thousand years, Euclid's (1956) fifth postulate, formulated in about 300 BC, that 'there is exactly one line passing through a given point and parallel to a given line' was accepted as a mathematical fact until it was challenged. The alternatives to this postulate give rise to non-Euclidean geometries, such as geometry on a curved surface and hyperbolic geometry (Cederberg, 2001).

The idea of challenging or changing the given is to take something that is a constant and let it vary (Kilpatrick, 1987). For example, some of the attributes of Pythagoras' theorem are: (i) the theorem deals with a right-angled triangle, (ii) the theorem deals with areas, and (iii) the variables in the theorem are related by an equal sign. Using the 'What-If-Not' strategy, students may pose these questions: "What if the triangle is not right-angled?", "What if it is not area but volume?" and "What if it is not an equal sign but an inequality?" Kilpatrick also cited Jim Kaput who communicated to him personally about the what-if-more approach, e.g. what if the indices in the algebraic formulation of Pythagoras' theorem are not 2 but more than 2, which will give rise to Fermat's Last Theorem under certain conditions. Orton and Frobisher (1996) used the 'What if' question to extend a problem by changing the given, which is exactly the same method as the 'What-If-Not' strategy.

Another way to extend an investigative task is to use 'association' (Kilpatrick, 1987), which has been described earlier in Section 2.2.3(b) as a problem-posing sub-process. Students might associate some key elements in the task statement or a discovered result with other concepts in their minds to pose further problems to investigate. If the further problems posed involve changing the given, this will come under the stage of extension. Kilpatrick also discussed the use of 'analogy' to extend the result of an investigation. He cited Pólya (1954, 1981) who showed that analogies could be a fertile source of new problems, e.g. after establishing Pythagoras' theorem, students could ask what the analogous proposition might be in solid geometry. However, there seems to be an overlap between association and analogy: an association might not be an analogy, but an analogy is an association.

The main purpose for changing the given in Type B tasks is to generalise. For example, students are expected to extend Task 8 (Sausage) in order to generalise: "How do I share *n* identical sausages equally among *m* people?" This means that during the extension, the students would have to try other examples (specialising) to search for patterns so as to formulate a conjecture in order to generalise, which is unlike the process of using other heuristics, such as reasoning, used to solve the original task. However, students would have to solve each example they generate using other heuristics. In other words, specialising and using other heuristics overlap during the extension of Task B tasks, and the processes for Type B tasks during extension (specialising, conjecturing, justifying and generalising) are similar to the usual processes for Type A tasks.

On the other hand, there is usually no need to extend Type A tasks because it is possible to pose more problems to solve without changing the given: after discovering a pattern and proving that the observed pattern is the underlying pattern, the students should search for more patterns for the original task. In fact, changing the given in a Type A task will usually result in changing the entire structure, which means that the underlying patterns will be very different from those of the original task. For example, if the word 'sum' in Task 7 (Kaprekar) is changed to 'product' as in Task 9 below, then this is a completely different task because the underlying patterns are totally different.

> **Task 9: Add Difference of Digits to Number**
>
> Choose any number. Add the *product* of its digits to the number itself to obtain a new number. Repeat this process for the new number and so forth. Investigate.

To summarise, the criterion to decide whether a problem posed for an investigative task is an extension, is to look at whether the given in the original task has been changed. Hence, there are three possibilities after solving a problem and checking the solution in mathematical investigation:

- pose more problems to solve without changing the given
- pose more problems to solve by changing the given (the last stage of extension)
- end the investigation.

Some interesting issues for the present study to examine are: (i) Do students know how to extend Type B tasks in order to generalise? (ii) What other kinds of problems do students pose to extend Type B tasks? These issues will be addressed in the data analysis in Section 7.3.6(a).

2.2.4 Empirical Studies on Cognitive Processes

"Research into the effectiveness of process-based teaching … is, however, limited, partly because process-based mathematical learning environments are extremely rare in schools." (Boaler, 1998, p. 42) As explained earlier in Section 2.1.5, empirical studies on mathematical investigation are even fewer than those on mathematical problem solving. Since there is an overlap of processes in problem solving and investigation (see Section 2.1.2), this section will also include the review of empirical studies on the nature and development of cognitive processes in problem solving as well as in mathematical investigation.

(a) Case Study of Mathematical Exploration by Cifarelli and Cai (2004)

Cifarelli and Cai (2004) examined two students engaging in mathematical exploration using a computer microworld called SNOOK, which simulates the path of a cue ball on a billiard table. They described the first student's activity as hypothesis-driven exploration (he made hypotheses and then proceeded to test them) and the second student's activity as data-driven exploration (he examined empirical data to search for patterns). They presented a model to describe the general structure of mathematical exploration: sense-making, formulation of goals (problem posing) and achievement of goals (problem solving). But they understood that the processes involved in the actual exploration were much more complicated (see Fig. 2.8 which shows the complex pathways of the second student's actual processes).

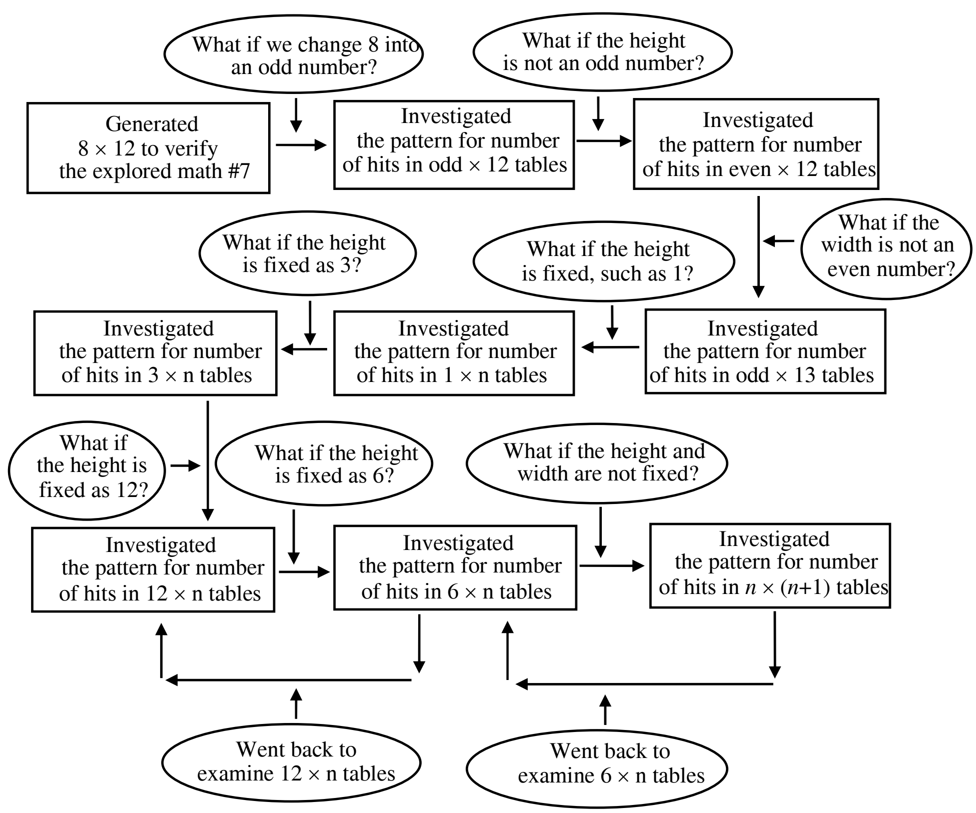


Figure 2.8 Model of Student's Mathematical Exploration by Cifarelli and Cai

(b) Teaching Experiment on Heuristics by Schoenfeld (1985)

Schoenfeld (1985) cited some empirical studies (e.g. Smith, 1973; Wilson, 1967) to suggest that the teaching of general heuristics did not transfer well to new situations because "problem-solving strategies are both problem- and student-specific" (Begle, 1979, p. 145). Schoenfeld (1985) argued that heuristics were too general to help students solve problems because each general heuristic was actually the label given to a collection of many related sub-strategies, and different problems require different sub-strategies to solve. Therefore, Schoenfeld conducted a study where four upper-division science and mathematics majors from advanced undergraduate courses in mathematics in a university in the USA were given training in the use of five specific heuristics rather than the general heuristics. The control group consisted of three undergraduates with comparable mathematical backgrounds from the same courses. All the students were trained to talk aloud as they solved some mathematical tasks and then they each took a pretest consisting of five tasks using the thinking-aloud method. After that, each student went for five training sessions over a period of two weeks. During each session, the student worked on each of the four practice tasks for up to 15 minutes or until it was solved, whichever was earlier. Then the student looked at a hard copy of the solution and listened to a recorded explanation of the solution.

The first difference between the two groups was that the students in the experimental group were given a list of five specific strategies that was placed in front of them during all the practice sessions and the posttest. The second difference was that the solution in the hard copy for the experimental group contained an additional column showing some questions that the student should ask himself or herself and what heuristic to use. The third difference was that the taped explanation of the solution for the experimental group contained some additional explicit heuristic instructions. Results from the posttest showed that all four students in the experimental group had improved in terms of solving the problems completely or almost completely, but only one student in the control group had improved, though not as much as those in the experimental group. Schoenfeld concluded that it was not just a matter of practising problem solving but that explicit instruction in specific heuristics did make a difference.

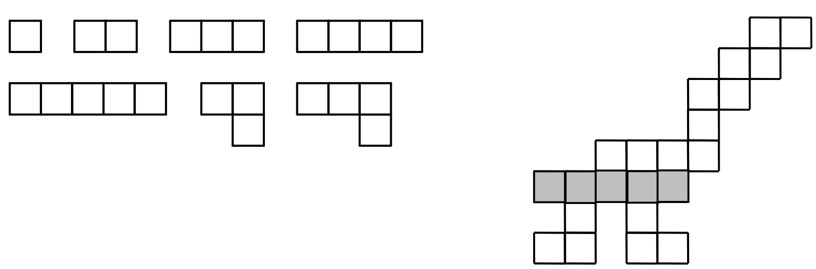
(c) Exploratory Study of Heuristics in Geometrical Problem Solving by Liu (1982)

Liu (1982) investigated the heuristics used by a group of 40 male students chosen randomly from three Form Five (i.e. Grade 11) classes in a Hong Kong school when they solved some geometry problems. The students used the thinking-aloud method to verbalise their thought processes during the problem solving and their protocols were recorded on audio tapes and coded. Of the 22 heuristics that she had identified beforehand, only 16 could be reliably coded. Examples of the 16 heuristics included quoting theorems, using algebraic processes, working forwards, working backwards, reading the question, devising a plan, and understanding the hint implied, although it was observed that there were overlaps of some of these heuristics in her classification. Liu then identified four factors: logical ability, use of strategy, use of goal-oriented syntheses and mathematical attainment. Her factor analysis showed that the dominant factor underlying geometrical problem-solving performance was logical ability which accounted for 40% of the variance. She concluded that logical ability was the basic ability involved, that mathematics attainment only played a minor role, and that mastery of heuristics is essential to success, in geometrical problem-solving.

(d) Research Study of Problem-Solving Processes by Stein and Burchartz (2006)

Stein and Burchartz (2006) investigated students' problem-solving processes when they attempted to solve puzzles that were not solvable. This was part of the Invisible Wall project that had been running in Germany since 1992. In this study, 41 pairs of students from Grades 3 and 4 in 17 state primary schools, and 45 pairs of students from Grades 7 and 8 in 12 lower secondary state schools, were randomly given one big and one small unsolvable puzzle. An example of an unsolvable puzzle is to try to fit seven given tiles into the big puzzle as shown in the diagram below. A small puzzle is just smaller in size with a different set of tiles. The students were informed that some of the puzzles could not be solved. Each pair was videotaped as they worked on the puzzles.

Tiles for Big Puzzle Big Puzzle ("giraffe")



It was found that some students engaged in monster-barring, a phrase that the researchers adopted from Lakatos (1976). These students observed that if they fitted the 5-unit tile into the shaded[4] squares in the above big puzzle, then they would not have enough 3-unit right-angled tiles to fit the "legs of the giraffe". So they happily removed the 5-unit tile and then proceeded with trying to fill the puzzle with the other tiles, without realising that the shaded squares were the only squares in the puzzle that

[4] All the squares were not shaded in the puzzle; the five squares were only shaded for ease of reference.

could fit the 5-unit tile. The students' answers were classified into three categories: complete proof (e.g. "the 5-unit tile and the 4-unit tile have these unique positions, and then we have no space for the 3-unit tile"), demonstration of the way the problem was solved (e.g. "if you cover it this way, you cannot place the 5-unit tile"), and naming isolated facts (e.g. "this can't be done since the parts are too long"). The results show that there were a large number of students in the last category, especially those in primary schools. Although some of these students did understand the logical structure, they were unwilling to give a full explanation or did not see the point in doing so. The researchers realised that "in many cases students say less than they know, and in many other cases valuable insights got lost during the problem-solving process" (p. 81) because many younger students tended to use actions as part of their reasoning. Thus they concluded that they needed to look at the actions accompanying the younger students' spoken words so as not to underestimate their reasoning ability.

(e) Research Study of Knowledge and Strategies in Problem Solving by Kaur (1995)

Kaur (1995) investigated 626 students from one primary and two secondary schools in Singapore using two paper-and-pencil test instruments: a 'Problems Test' and a 'Computations Test'. These students had not been exposed to any explicit mathematical problem-solving instructional programme in school and so they were called novice problem solvers. The results of the tests suggested that there appeared to be a 'gap' between a student's ability to carry out particular mathematical calculations and operations, and the ability to solve non-routine problems employing the same mathematical computations. 139 students, who obtained correct 'Computations items' corresponding to incorrect 'Problems items' employing the same type of computation, were interviewed to identify their difficulties. It was found that the common difficulties were the lack of comprehension of the problem, the lack of strategy knowledge, and the inability to translate the problem into a mathematical form.

Additional data were collected using another paper-and-pencil test instrument for 63 students who did well in the 'Problems Test' in order to identify the common strategies they used to solve these problems. They were also asked to respond to the statement, "When my teacher gives me a mathematics problem to solve, this is what I do…" at the end of the test. The data were then coded using a taxonomy of problem-solving behaviours based on the four categories of Pólya's (1957) model. It was found that the range of problem-solving strategies used by novice problem solvers expanded with increasing age and mathematical maturity, for example, the majority of good novice problem-solvers in Primary 6, Secondary 1 and 2 (i.e. Grades 6 to 8), were able to use strategies such as draw a diagram/picture, work backwards, make a list and make a table, while those in Primary 5 were only able to use the first three strategies. The findings also suggested that good novice problem solvers differed from poor novice problem solvers in their selection and use of strategies during problem solving.

(f) Mathematical Problem Solving for Everyone (M-ProSE) Project by Dindyal, Tay, Toh, Leong and Quek (2012)

Dindyal, Tay, Toh, Leong and Quek (2012) described a mathematics problem solving package, comprising what they called 'mathematics practical lessons', which was trialled in a Singapore school for 159 Grade 8 students in 2009. The school is a special school catering to high-performing students in mathematics and science, and it has a flexible curriculum that can accommodate a new module. The teaching package consisted of 10 lessons, each lasting 55 minutes. The students were taught Pólya's (1957) problem-solving model with an additional emphasis on the control strategy in Schoenfeld's (1985) model (both models had been described in Section 2.2.2). A 'practical worksheet' was designed to guide the students to use Pólya's four stages and problem-solving heuristics to solve mathematical problems. An example of an instruction in the worksheet is: "Write down in the *Control* column the key points where you make a decision or observation, e.g. go back to check, try something else, look for resources, or totally abandon the plan." An assessment rubric was developed based on the application of Pólya's four stages and the heuristics in the worksheet.

The end-of-module assessment consisted of one mathematical problem that all the students attempted in 55 minutes with the use of the practical worksheet. It was found that 69.8% of the students attained the maximum score of 10 for applying the first three of Pólya's stages, but only 7.5% scored the maximum of 6 for the last stage of checking and extending the problem. 74.8% of the students also scored the maximum of 4 for using problem-solving heuristics. After the trial period, the school adopted the module as a compulsory course for all their Grade 8 students.

2.2.5 Summary of Literature Review on Cognitive Processes

The review of relevant literature on cognitive processes in Section 2.2 has helped to clarify the types of processes in mathematical investigation and how they are similar or different from those in problem solving. The four main mathematical thinking processes (i.e. specialising, conjecturing, justifying and generalising) are common to both investigation and problem solving. Other common main cognitive processes include understanding the task, using other heuristics, checking and extension. What is different between investigation and problem solving is that the former includes the main process of problem posing (or goal setting) after understanding the task. The two types of investigative tasks (Types A and B) also seem to elicit somewhat different processes. For Type A tasks, students usually pose the general problem of searching for any pattern, resulting in specialising and conjecturing, and there is usually no need to extend the task because students can discover many patterns in the original task; while for Type B tasks, students usually need to pose more specific problems which can be solved by using other heuristics such as reasoning, with or without formulating any conjecture, but there was usually a need to extend the task in order to generalise.

Models are found to be useful to display the interactions of these processes with one another. But most of these processes and their interactions with one another are based on theory. There is also a lack of empirical research into the actual processes that students use when they attempt investigative tasks, and how these processes can be developed. Therefore, the present research study will try to address this gap in current research by documenting the actual processes engaged by Secondary 2 students and developing these processes in the students if possible. Since these processes include not only the cognitive processes but also the metacognitive ones, there is a need to review relevant literature on metacognition so as to inform the present study.

2.3 LITERATURE REVIEW ON METACOGNITIVE PROCESSES

If research on cognitive processes in mathematical investigation is rare, then research on metacognition in investigation is almost non-existent, despite an extensive search of existing literature. Therefore, the review of literature in this section will have to draw mostly from research studies on metacognition in problem solving. This section will begin by looking at the nature of metacognition, followed by a discussion of the differences between metacognitive and cognitive processes. It will then review three frameworks of cognitive-metacognitive processes, and empirical studies on the nature and development of metacognition. Lastly, this section will end with a summary of the main findings on metacognitive processes in current literature.

2.3.1 Nature of Metacognition

The term ‘metacognition’ was coined by Flavell (1976) to refer to “one’s knowledge concerning one’s own cognitive processes and products or anything related to them, e.g., the learning-relevant properties of information or data” (p. 232). To bring the meaning of metacognition down to the level of teachers who were confused about this term during a conference, Schoenfeld (1987) translated metacognition as “reflections on cognition or thinking about your own thinking” (p. 189) in everyday language. Keiichi (2000) considered metacognition from the viewpoint of education and regarded metacognition as “the knowledge and skills which make the objective knowledge active in one’s thinking activities” (p. 65). According to Brown, Bransford, Ferrara and Campione (1983) and Schraw (2001), most researchers made a distinction between knowledge of cognition and regulation of cognition when they talked about metacognition.

The first category, knowledge of cognition or metacognitive awareness, refers to what a person knows about his or her own cognition abilities. Schraw (2001) described at least three kinds of metacognitive awareness: declarative, procedural and conditional knowledge (Brown, 1987; Jacobs & Paris, 1987; Schraw & Moshman, 1995). Declarative knowledge refers to knowing about things, which includes knowledge about oneself and about what influences one's performance. Procedural knowledge refers to knowing how to do things, which includes heuristics and strategies. Conditional knowledge refers to knowing when to use what types of declarative and procedural knowledge and why. Schoenfeld (1987) observed that children are not very good at describing their own knowledge of cognition, but as they get older, they get better, although it is far from perfect. Schoenfeld emphasised that this is important because "good problem solving calls for using efficiently what you know; if you don't have a good sense of what you know, you may find it difficult to be an efficient problem solver" (p. 190).

The second category, regulation of cognition, refers to how well a person controls his or her own learning behaviour (Schraw, 2001). There is a whole list of regulatory skills described in literature (Schraw & Dennison, 1994), but three essential skills are mentioned in all accounts: planning, monitoring and evaluating (Jacobs & Paris, 1987). Planning involves selecting appropriate strategies and allocating resources that affect performance, e.g. strategy sequencing and allocating time and attention selectively before beginning a task. Monitoring involves taking stock of one's performance, e.g. pausing periodically during problem solving or mathematical investigation to check one's own progress and direction. Evaluating involves assessing the products and effectiveness of one's learning, e.g. reflecting at the end of problem solving or mathematical investigation. Another way to look at regulation in metacognition is a management issue: how well a person manages his or her time and effort as he or she is working on complex tasks (Schoenfeld, 1987). This includes (a) making sure that you understand what the problem requires before hastily attempting a 'solution' which might not even solve the problem, (b) planning, (c) monitoring or keeping track of how well you are progressing during problem solving, and (d) allocating resources, or deciding what to do and for how long, before you decide whether it is worth pursuing further or not.

However, some researchers have categorised metacognition differently. For example, Schoenfeld (1987) included a third category called beliefs and intuitions, although this was not classified under metacognition in his earlier work (Schoenfeld, 1985). He has conducted some research studies to suggest that beliefs and intuitions can influence a student's problem-solving behaviour. "Many students come to believe that school mathematics consists of mastering formal procedures that are completely divorced from real life, from discovery, and from problem solving" (Schoenfeld, 1987, p. 197) so that they fail to use the mathematics that they have learnt to solve genuine mathematical problems. But some researchers (e.g. Brown et al., 1983; Schraw, 2001) have classified Schoenfeld's (1987) third category of metacognition under the first category of metacognitive awareness described earlier in this section.

Others classified metacognition *differently* into two other categories: metacognitive knowledge and metacognitive experiences. "Metacognitive knowledge refers to the part of one's acquired world knowledge that has to do with cognitive (or perhaps better, psychological) matters" (Flavell, 1987, p. 21) or "declarative, memory-retrieved knowledge regarding goals people pursue in cognitive endeavors, persons, tasks and strategies" (Desoete & Veenman, 2006, p. 3). "Metacognitive experiences are conscious experiences that are cognitive and affective. What makes them metacognitive experiences rather than experiences of another kind is that they have to do with some cognitive endeavor or enterprise, most frequently a current, ongoing one.” (Flavell, 1987, p. 24) Desoete and Veenman (2006) defined metacognitive experiences as “the feelings and judgments or estimates regarding cognitive processing [and] in this way, metacognitive experiences serve the monitoring and control of the learning process, and at the same time provide an intrinsic context and future motivation towards learning” (p. 3). Desoete and Veenman also added a third category: metacognitive skills, which are “defined as procedures or strategies that persons actually apply to monitor and control cognition, rather than what the person knows about strategies and the conditions of their applicability” (ibid., p. 3).

There are also other perspectives on metacognition. For example, Keiichi (2000) viewed metacognition as ‘the inner teacher’ from which a student learns. Mason et al. (1985) also talked about metacognition as “your own internal tutor” (p. 115). They advocated developing an internal monitor to regulate problem solving, such as keeping an eye on the execution of a plan to make sure that it does not drift too far off course, evaluating ideas as they come along to see if they are worth pursuing, insight, mulling when you are stuck, changing strategy when it appears to lead to nowhere and prompting you to review your solution before finishing the work. Insights, like intuitions, are “not something you can intentionally bring about [but] you can prepare for it by … doing the spadework of specialising and generalising” (p. 127). Mason et al. suggested that students should conscientiously monitor their thinking until “you become aware *in the moment* of your thinking processes” (p. 118).

It is beyond the scope of this thesis to study how metacognitive knowledge and beliefs affect investigation. The present research will only focus on metacognitive regulation, which consists of the main metacognitive processes of planning, monitoring and evaluating, during mathematical investigation.

2.3.2 Differences between Metacognitive and Cognitive Processes

Other than the different views about what metacognition consists of, another main problem lies in distinguishing what is metacognitive from what is cognitive (Garofalo & Lester, 1985). The first viewpoint is that “cognition is involved in doing, whereas metacognition is involved in choosing and planning what to do and monitoring what is being done” (ibid., p. 164). The second viewpoint is that cognition is thinking while metacognition is “thinking about your own thinking” (Schoenfeld, 1987, p. 189). For example, from the first viewpoint, understanding a task before attempting to solve it is metacognitive because the process involves planning rather than doing (Artzt & Armour-Thomas, 1992). But understanding involves thinking, and according to the second viewpoint, thinking is cognitive and not metacognitive, so understanding a task should be cognitive. What is metacognitive is when a person monitors his or her own understanding. Moreover, renowned educators, like Pólya (1957), Mason et al. (1985) and Schoenfeld (1985), have always considered understanding the problem or the task as a cognitive process, stage or episode.

In problem solving or investigation, when students spend time trying to understand a task instead of jumping straight in to solve the problem, they are aware that they need to understand the task properly first and so they are regulating their understanding (Schoenfeld, 1985, 1987). But if they set off immediately on a wild goose chase after a casual reading of the task, it means that they do not monitor their problem-solving behaviour. Thus even if understanding the task is viewed as cognitive, students are engaged in metacognition if they see the need to regulate their understanding. Similarly, when students check their solution, this is considered a cognitive process. What is metacognitive is when students review their solution to see if it has solved the problem or met the goal of the investigative task. Therefore, this thesis will consider the main processes for mathematical investigation described earlier in Section 2.2.3 as cognitive processes or stages. Within these stages, there are various cognitive and metacognitive sub-processes. For example, in the understanding stage, re-reading the task and highlighting key information are cognitive sub-processes, while monitoring understanding is a metacognitive sub-process; and in the checking stage, checking working is a cognitive sub-process, while reviewing the solution to see if it had met the goal of the task is a metacognitive sub-process.

2.3.3 Models of Metacognitive Processes

Metacognitive processes do not act alone: they usually interact with cognitive processes (Garofalo & Lester, 1985). In this section, three frameworks of cognitive-metacognitive processes from the literature will be reviewed.

(a) Cognitive-Metacognitive Framework by Schoenfeld (1985)

Schoenfeld (1985) identified six types of cognitive stages or episodes during problem solving: read, analyse, explore, plan, implement and verify. He then used a timeline to represent these episodes, with overt signs of metacognitive behaviour indicated by inverted triangles ▼ as shown in Figure 2.9. Schoenfeld focused on decision-making behaviour at what he called the executive level. Examples of such metacognitive processes include management of resource allocation, deciding whether a plan is worth pursuing, and reflecting at various points of implementation of the plan whether you are on the right track. However, it was not possible to reflect all these specific metacognitive processes in the timeline. Schoenfeld also discussed about the transition between two episodes when the pair of students engaged in a discussion. Sometimes, the discussion during this transition period had led to some productive breakthrough, but very often, the discussion was unproductive in his study.

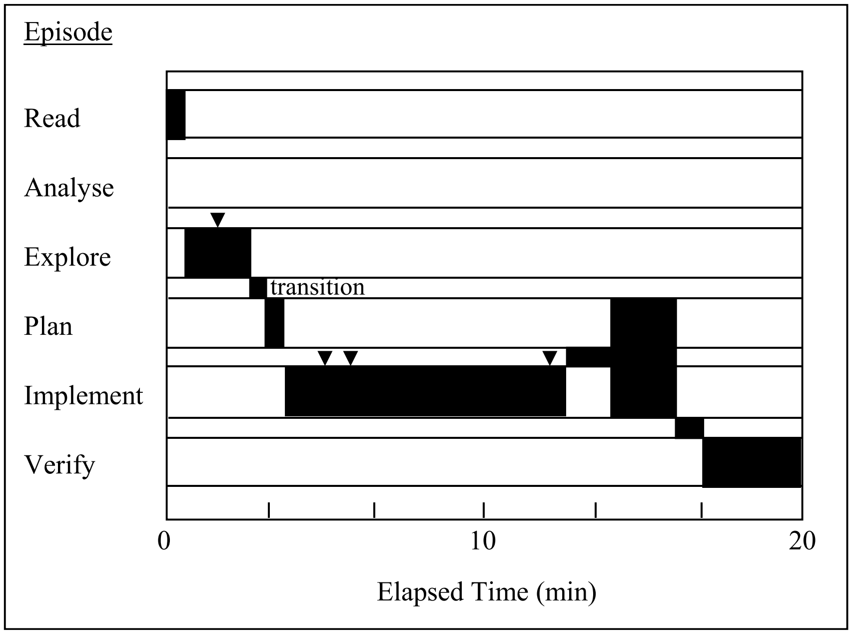


Figure 2.9 Timeline Representation of Metacognition by Schoenfeld

(b) Cognitive-Metacognitive Framework by Garofalo and Lester (1985)

One of the earliest and frequently cited cognitive-metacognitive frameworks in mathematics educational research is from Garofalo and Lester (1985). This taxonomy is based on a combination of the work of Pólya (1957), Schoenfeld (1983, 1984) and Sternberg (1982). It comprises of four cognitive categories: orientation, organisation, execution and verification. Figure 2.10 below shows the distinctive metacognitive processes within each category.

ORIENTATION: Strategic behaviour to assess and understand a problem
A. Comprehension strategies
B. Analysis of information and conditions
C. Assessment of familiarity with task
D. Initial and subsequent representation
E. Assessment of level of difficulty and chances of success

ORGANISATION: Planning of behaviour and choice of actions
A. Identification of goals and subgoals
B. Global planning
C. Local planning (to implement global plans)

EXECUTION: Regulation of behaviour to conform to plans
A. Performance of local actions
B. Monitoring of progress of local and global plans
C. Trade-off decisions (e.g. speed vs. accuracy, degree of elegance)

VERIFICATION: Evaluation of decisions made and of outcomes of executed plans
A. Evaluation of *orientation* and *organisation*
  1. Adequacy of representation
  2. Adequacy of organisational decisions
  3. Consistency of local plans with global plans
  4. Consistency of global plans with local plans
B. Evaluation of *execution*
  1. Adequacy of performance of actions
  2. Consistency of actions with plans
  3. Consistency of local results with plans and problem conditions
  4. Consistency of final results with problem conditions

Figure 2.10 Cognitive-Metacognitive Taxonomy by Garofalo and Lester

(c) Cognitive-Metacognitive Framework by Artzt and Armour-Thomas (1992)

Artzt and Armour-Thomas (1992) built upon the work of Garofalo and Lester (1985) and Schoenfeld (1985). Their main criticism of the cognitive-metacognitive taxonomy of Garofalo and Lester, and Schoenfeld's timeline representation, was that these frameworks did not include specific cognitive processes. Although Schoenfeld (1985) viewed the six episodes of reading, analysing, exploring, planning, implementing and verifying as cognitive, Artzt and Armour-Thomas (1992) believed that some of these episodes are predominantly metacognitive. Therefore, they developed a taxonomy for problem solving in small groups that classifies these episodes as predominantly cognitive or metacognitive. They identified eight types of episodes: read, understand, analyse, explore, plan, implement, verify, and watch and listen. Then they assigned to each episode a predominant cognitive level as shown in Table 2.2.

Table 2.2 Framework Episode Classified by Predominant Cognitive Level

Episode	Predominant Cognitive Level[5]
Read	Cognitive
Understand	Metacognitive
Analyse	Metacognitive
Explore	Cognitive and Metacognitive
Plan	Metacognitive
Implement	Cognitive and Metacognitive
Verify	Cognitive and Metacognitive
Watch and Listen	(level not assigned)

Artzt and Armour-Thomas (1992) explained that episodes of watching and listening during problem solving in small groups were not assigned any predominant cognitive level because the lack of verbalisation from the students during these episodes makes it impossible to infer a level of cognition. They also clarified that some episodes, such as understanding, involve both cognitive and metacognitive processes, but it is impossible to distinguish between them just by observing the verbal comments of students. However, they still classified understanding as predominantly metacognitive because they subscribed to the view of Garofalo and Lester (1985) described earlier in Section 2.3.1 that cognition is about doing and metacognition is about planning and monitoring. Artzt and Armour-Thomas (1992) even developed a model to show the interactions of the various episodes (see Fig. 2.11).

[5] The authors did not call it 'Predominant Cognitive or Metacognitive Level' although the level could also be metacognitive.

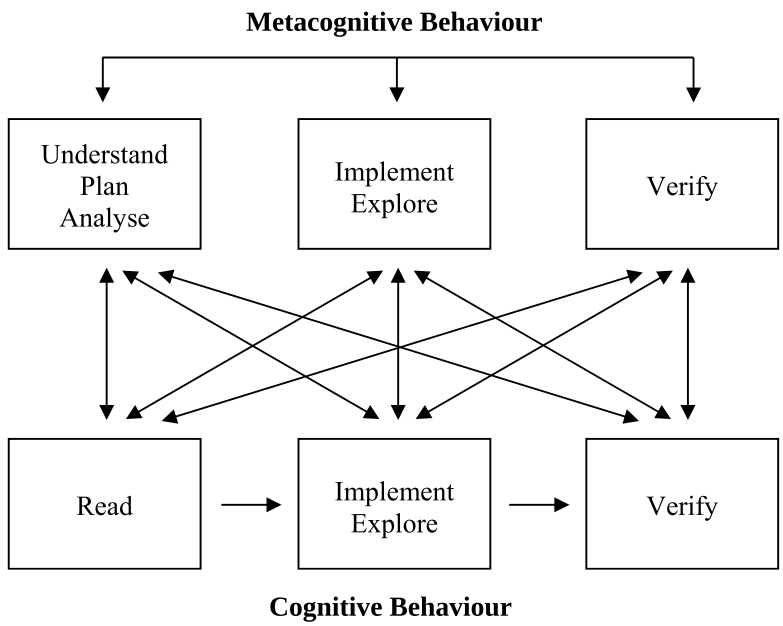


Figure 2.11 Artzt's and Armour-Thomas' Cognitive-Metacognitive Model

What is missing from all the three frameworks described above is the problem-posing stage of investigation because these frameworks were designed primarily for problem solving and not for investigation. Therefore, there is a need to fill in the gap in existing literature when developing a framework for the interactions between cognitive and metacognitive processes in mathematical investigation for the present study by including the metacognitive processes during the problem-posing stage.

2.3.4 Empirical Studies on Metacognitive Processes

As explained at the start of Section 2.3, research on metacognition in investigation is almost non-existent, so the review of empirical studies on metacognitive processes in this section will draw entirely from literature on metacognition in problem solving.

(a) Research Study on Metacognition by Schoenfeld (1985)

Schoenfeld (1985) studied some college freshmen attempting to solve an unfamiliar problem in pairs. Each pair was given 20 minutes and their discussion was recorded on audio tape. One such mathematical problem is to show that it is always possible to construct, with straightedge and compasses, a straight line parallel to the base of a given triangle such that the triangle is divided into two equal parts. The first pair of undergraduates immediately assumed that the required line must be the line joining the midpoints of the other two sides (which is wrong), and they spent considerable amount of time doing the actual construction, instead of giving more thought to whether that was really the required line. Schoenfeld called this an example of bad control that contributed to failure. 'Control' is also called 'regulation of cognition' in other literature on metacognition described earlier in Section 2.3.1. The second pair managed to curtail wild goose chases before they caused disaster, but the executive behaviour (another name for control) did not really help them to solve the problem.

Schoenfeld also gave the problem to a mathematician who had not done plane geometry for many years, but the mathematician managed to solve it by choosing his resources carefully, and exploiting or abandoning the resources appropriately as a result of careful monitoring. His control decisions exerted a positive influence on his problem solving. These findings suggested that control is crucial in determining the success of problem solving. Even though the first two pairs had studied plane geometry in recent years and had the content knowledge to solve the problem, compared with the third subject who had not done plane geometry for many years, the former were unable to solve the problem due to lack of control decisions while the latter was successful.

(b) Research Study on Metacognition by Focant, Grégoire and Desoete (2006)

Focant, Grégoire and Desoete (2006) investigated 42 fifth-grade students randomly selected from six schools in the French-speaking part of Belgium in 2002. The test instruments were goal setting and planning tasks, a standardized mathematical test on arithmetical problem solving, and a control task. In the goal setting and planning tasks, the students had to solve several problems individually and they had to answer these two questions: "What is this problem asking for?" (goal setting) and "You have not solved the problem effectively. But if you should solve it, how would you do this?" (planning). The results showed that almost all the students had mastered goal setting, and the average student was able to plan a two-step arithmetic problem but not a three-step arithmetic problem. The mathematical test was used to measure problem-solving performance, and students who failed in two of the three items were discarded from the study because the mastery of computational ability was necessary for the control task. In the control task, the students had to detect errors in eight arithmetic tasks, out of which four of them had computational errors and the other four used the wrong arithmetic operation (plus, minus, times and divide). The results suggested that most of the students were successful in the control strategy.

(c) Teaching Experiment on Metacognition by Mevarech and Fridkin (2006)

Mevarech and Fridkin (2006) investigated 81 pre-college students in Israel who obtained a low score on the Israeli matriculation exam in mathematics. They were randomly assigned into two groups. One group was exposed to the IMPROVE programme (experimental group) and the other to traditional learning instruction (control group) for 12 hours a week for one month. The IMPROVE method was the acronym of all the teaching steps: Introducing new concepts, Metacognitive questioning, Practising, Reviewing, Obtaining mastery, Verification, and Enrichment and remedial. The test instruments, consisting of mathematical examinations and two questionnaires (general and domain specific metacognitive questionnaires), were administered before and after the intervention. The general metacognitive questionnaire had 52 items to assess the students' knowledge and regulation of cognition. The domain specific metacognitive questionnaire consisted of 24 items that assessed the students' metacognitive knowledge in the area of solving maximum and minimum problems. It was found that, for both questionnaires, the two groups did not differ significantly during the pretest but the difference between the two groups increased significantly during the posttest. The findings of the study showed that students in the experimental group had developed a higher level of metacognition.

(d) Research Study on Metacognition (Keiichi, 2000)

Keiichi (2000) videotaped a lesson in a school in Japan where the students in middle grades were given a word problem to solve. The students worked on it individually before discussing with their classmates. After the lesson, the students were shown the videotape for about two to three minutes at four specific points to stimulate recall: when they were given the task, when they began working on it individually, when they began working on it with their classmates and after they finished working on it. A questionnaire was administered to the students concerning what they had watched on the videotape. Some of the questions were designed to investigate the students' metacognition, e.g. "What kind of ideas occurred to you while you were watching the videotape? Did you remember what your teacher said while you were watching the videotape? Did you have a will to remember the teacher's advice?" The researcher also analysed the students' answer scripts which contained two sides: the left side was for the students to work on the mathematical task, and the right side was for the students to record their thinking processes. It was found that metacognition played a vital role in solving word problems and that the teacher's utterances which impressed most students were those on mathematical reasoning and problem-solving strategies. The videotape also revealed that the teacher often emphasised problem-solving strategies to the exclusion of other aspects of problem solving such as metacognition.

(e) Survey Study on Metacognition in Problem Solving by Wong (1989)

Wong (1989) described a research study on metacognitive strategies in mathematical problem solving for 670 students from nine secondary schools and four junior colleges in Singapore. The students were from different streams (special, express and normal academic), levels (Secondary 2, Secondary 4 and JC 1) and academic tracks (arts, science and general). Students had to fill up a questionnaire based on a 5-point Likert scale asking them about their metacognitive strategies according to the four components in the cognitive-metacognitive framework of Garofalo and Lester (1985): orientation, organisation, execution and verification (see Fig. 2.10 in Section 2.3.3). It was found that there was no significant difference in the mean score for each component between special and express streams, but that the normal stream scored significantly lower than both special and express streams. However, there was no significant difference in the mean score for each component among the three levels and among the three academic tracks. One trend that surfaced among all the different streams, levels and academic settings was that the mean for the verification component was a lot higher than the mean of each of the other three components.

(f) Metacognitive-Heuristic Approach to Problem Solving by Foong (1990)

Unlike the paper-and-pencil survey study in Wong (1989), Foong (1990) used the thinking-aloud methodology and protocol analysis as 57 trainee teachers in Singapore attempted mathematical problem solving individually. The main purpose was to investigate successful and non-successful problem-solving processes. It was found that there were eight behavioural factors that influence the trainees' success or failure in problem solving: solution-oriented metacognitive heuristics, analysis with diagram, domain-specific knowledge, planning, deductive approach, maladaptive emotional, misinterpretation of the problem, and tendency to compute numbers. The first five are positive factors while the last three are negative traits. The second part of Foong's study was to investigate the effectiveness of a small scale teaching experiment for improving problem-solving performance among eight trainee teachers using a metacognitive-heuristic approach. There were six lessons lasting one-and-a-half hours each. The findings suggest that it is possible to teach heuristics and metacognition explicitly to trainees, and that the trainees can learn how to use these effectively in problem solving.

2.3.5 Summary of Literature Review on Metacognition

The review of relevant literature on metacognitive processes in Section 2.3 has helped to clarify the construct of metacognition. It is beyond the scope of the present study to investigate the effect of metacognitive knowledge and beliefs on the proficiency of the students' mathematical investigation, so the study will focus on the main metacognitive processes that regulate cognition: planning, monitoring and evaluating. The boundary between metacognitive and cognitive processes is not distinct, so there may be some difficulty in classifying a sub-process as metacognitive or cognitive. In general, sub-processes associated with 'doing' and 'thinking' are considered cognitive, while sub-processes that involve 'thinking about thinking' or 'monitoring' are categorised as metacognitive. Empirical studies on metacognition in investigation are almost non-existent, but empirical studies on metacognition in problem solving have suggested that metacognition could be developed. Moreover, the review of empirical studies on both cognitive and metacognitive processes has uncovered various research methodologies that may help to capture students' thinking processes more effectively for the present research study on investigation.

2.4 LITERATURE REVIEW ON RESEARCH METHODOLOGIES

It is important to be clear about the types of data to be collected for the present research before deciding on the methodology because the methodology chosen should be suitable to capture the necessary information for the research, rather than selecting a methodology first and then deciding what types of data can be collected using the research design (Kelly & Lesh, 2000a). As the present study researches the nature and development of processes in mathematical investigation, there is a need to review literature on how to collect data on students' thought processes, including the thinking-aloud method, and on how to analyse such data. Then the research designs for developing students' processes in investigation will be reviewed, followed by a summary of the relevant literature review on research methodologies.

2.4.1 Methods of Data Collection for Processes

There are various methods of data collection for thought processes. In this section, eight of these methods will be examined, especially from empirical studies that used these methods. The pros and cons of these methods will also be analysed critically with support from current research literature. This is to inform the research methodology needed to study the nature of cognitive and metacognitive processes in the present study.

The first method of data collection is to use a questionnaire to find out whether students engage in certain processes, e.g. the empirical study conducted by Mevarech and Fridkin (2006) described earlier in Section 2.3.4(c). The advantage of this method is that the questionnaire can be used for a large sample. The disadvantages are: (i) the students may not understand some of the terminologies used in the questionnaire since it may be beyond them to comprehend the meanings of such terms just by listening without experiencing them personally (Wong, 1989), and it may not be easy for the person who conducts the survey to explain the terminologies clearly to every student, (ii) it is easier for the students not to be truthful in their answers, e.g. they may claim that they usually verify their answers but they may not do so in reality, and (iii) the questionnaire does not capture the actual thought processes of the students.

The second method is to examine the students' answer scripts on investigation or problem solving, e.g. the empirical study conducted by Keiichi (2000) described earlier in Section 2.3.4(d). Some problem-solving heuristics, such as systematic listing and drawing a figure, can be gathered from the answer scripts, but other heuristics may not be so evident in the final answer because the student may have used these heuristics during the investigation but not in the final presentation of the solution (Jaworski, 1994; Lakatos, 1976). Therefore, the final solution will not reveal many of the thought processes that may have occurred during the investigation.

The third method is to look at the students' rough working. Although rough working may be better than the final solution to reveal the students' thought processes, it is also not good enough because some students may think and plan without writing anything down and so these thought processes will also not be captured.

The fourth method is to use structured task-based interviews (Goldin, 2000). The students will be observed individually as they solve a problem, but hints or heuristic suggestions will be provided to the students if "the child's free problem solving came to such a firm halt or impasse that further progress seemed unlikely" (ibid., p. 522). The interviewer or clinician will have to be specially trained with interview scripts that describe branching sequences of possible questions and interventions so that appropriate assistance can be rendered to the students when they are stuck. This is unlike an unstructured interview where no substantial assistance will be provided to the students. The purpose of such intervention is to bridge the gaps in the students' partially developed heuristic planning competencies, so that the interviewer will be able to observe the students' problem-solving path towards a solution which may not be possible otherwise. However, this method is not suitable for the present study because the purpose of the data collection for this study is to describe and assess students' behaviours during mathematical investigation, not to intervene.

The fifth method is to interview the students individually after they have finished the test. This is called retrospective interview. During the interview, the students will be asked to recall their thought processes during certain episodes of the investigation. In this way, the researcher can find out more about what the students were thinking when they were investigating during the test. However, the main problems are (i) the students may not remember all their thought processes (Liu, 1982), (ii) the students may rearrange their thought processes into a more coherent and logical order, and so the actual sequence of thinking is not observed (Kilpatrick, 1968), and (iii) the students may confuse their current knowledge during the interview with their past knowledge during the test, and thus present to the interviewer what the students think should be the 'correct' thought processes based on current knowledge and not what actually occurred during the test (Newell & Simon, 1972). Nevertheless, retrospective interviews are useful for gathering some types of information, such as error analysis in Newman's (1983) clinical interviews, but they are not as appropriate to document the students' step-by-step thought processes in the current study.

The sixth method is to ask students to write down after the test what they have done when solving a problem or doing an investigation, e.g. the empirical study by Kaur (1995) described earlier in Section 2.2.3(e), and the research work of Focant et al. (2006) in Section 2.3.4(b). The disadvantage compared with the fifth method is that there is no chance to clarify what the students have written if it is not clear. But if the sample size is small, it is possible to combine the fifth and sixth methods where some or all the students will be selected for retrospective interviews to clarify what they have written. However, the method is still not suitable for the present study because it cannot capture the students' step-by-step thought processes during investigation.

The seventh method is to ask the students to write down what they are thinking when they are investigating, e.g. the empirical study by Keiichi (2000) described earlier in Section 2.3.4(d). This method may solve the problem of the students confusing their current knowledge during a retrospective interview with their past knowledge during the test. But the main problem is that thinking may be so rapid that it may be impossible to write it down exactly as it occurs and so there is still a possibility of editing or summarising one's own thought processes (Lui, 1982). The students may also think and write some rough working at the same time, and since it is hard to juggle between writing the rough working and the thought processes simultaneously, the students may end up summarising their thought processes. Moreover, spending time to write down what they are thinking may interrupt the students' train of thought as they investigate. In fact, this was the method used in the initial exploratory study described later in Section 3.3, and it was found that these students either wrote very little or did not write anything at all.

2.4.2 Thinking-Aloud Method

The eighth method for collecting data for processes is to ask the students to say aloud what they are thinking during the investigation, e.g. the empirical study by Schoenfeld (1985) described earlier in Section 2.2.3b, and the empirical study by Foong (1990) discussed earlier in Section 2.3.4(f). This is called thinking aloud and was developed by Duncker (1926, as cited in Foong, 1990). Although thinking may be so rapid that it may be impossible to say it aloud as it occurs (Lui, 1982), it is still a lot easier to say it aloud than to write it down as in the seventh method. Moreover, if the students say aloud what they are thinking instead of writing it down, then they can still concentrate on the investigation by doing some rough working at the same time.

However, verbalising their thoughts may also interfere with their train of thought as they attempt to investigate, especially when they will be videotaped. Schoenfeld (2002) gave the example of how some college students, when videotaped alone to solve the problem of estimating the number of cells in an average-sized human adult body, were under tremendous pressure because they knew that he would be looking over their work, and as a result, they felt they needed to do something mathematical and ended up doing something which he described as 'ridiculously odd'. But when another sample of students was videotaped working in pairs on the same problem, they seemed to manage to dissipate some of the pressure by communicating with each other at the start by saying something like, "This sure is a weird problem." As a result, none of them ended up approaching the problem from the 'ridiculously odd' angle like the former sample of students. This suggests that thinking aloud, especially when videotaped, may add undue pressure on the students, and as a result, affects their way of thinking and their performance. But there is a difference between articulating one's own thought and talking to a team mate because the latter will usually involve reorganising one's own thought before speaking them out. Thus pair work will not be suitable since the focus of the present study is on the students' own thought processes. Moreover, the pressure of being videotaped might be alleviated to some extent if the students become familiar with the thinking-aloud method through practice.

One problem with all the last four methods is the inability of the students to describe or say what they are thinking. For example, Lampert (1990) reported that her students responded to "questions about how they figured something out with phrases such as 'I just know', 'I just thought it', or 'I don't know how I figured it out'" (p. 56); and Stein and Burchartz (2006), whose study was discussed earlier in Section 2.2.4(d), found that “in many cases students say less than they know” (p. 81). But it is still easier for students to just *report* what they are thinking as it occurs by writing it down or saying it aloud than to try to recall and *describe* these thought processes during retrospective interviews, as they just may not have the words to describe their thought processes. Moreover, it may be easier to train students to write down or think aloud their thought processes by letting them go through the experience during investigation than to train them to describe their own thought processes.

From the above review, it was found that thinking aloud is the most useful tool to describe a student’s step-by-step thought processes and it comes closest to reflecting the cognitive and metacognitive processes than retrospective interviews or writing down the thought processes (Ericsson & Simon, 1980; 1993). Clement (2000) believed that research on students’ thinking processes is most fruitfully undertaken using the thinking-aloud method. To address some of the shortfalls of thinking aloud that were described earlier, the students should be given some form of training before data collection. Foong (1990) concluded from her review of literature that thinking aloud was relatively easy to learn, as long as the students were motivated to cooperate. Some important points to emphasise to students during a thinking-aloud training session include:

- the students are to say aloud whatever that comes into their minds, rather than reflect on these thoughts and then rephrase or describe them (Clement, 2000) because introspection or reflection on such thoughts could interfere with the students’ train of thoughts as they attempt to investigate or solve a problem (Nisbett & Wilson, 1977);

- the students are to make an effort to translate thinking that uses nonverbal representations, such as imagery, into language because it is important to gain insight into this aspect of the students' thinking as it might influence their problem-solving or investigation behaviour (Ericsson & Simon, 1993).

However, the thinking-aloud method is only suitable if the sample size is small since it is not feasible to videotape each student thinking aloud separately if the sample size is large. But if there is a need to go for breadth in a research to study how different types of students investigate, depth may have to be sacrificed by asking the students to write down their thought processes or to answer a questionnaire. This may be complemented by selective retrospective interviews where only some students are selected to be interviewed based on what they have written since interviews are also time consuming. However, such methodology is still unable to capture the step-by-step thought processes accurately. But for research study where the sample size is small, the thinking-aloud method is still the most suitable method to track the students' actual thinking processes, despite the shortfalls.

2.4.3 Methods of Data Analysis for Processes

The method used to analyse thinking aloud of cognitive processes is called protocol analysis and was pioneered by Newell and Simon (1972). It refers to analysing a person's step-by-step behaviour which is recorded or videotaped while he or she engages in some cognitive tasks. Protocols are usually verbal and are obtained by asking the person to think aloud during problem solving or investigation. The recorded or videotaped data are then transcribed and coded using a coding scheme, which is usually developed and refined by a pilot study. Van Someren, Barnard and

Sandberg (1994) suggested some guidelines on how to construct a valid and reliable coding scheme for verbal protocols. They emphasised the need to map the psychological model (e.g. the various models described earlier in Section 2.2.2) to the thinking processes that would appear in the protocols. The psychological model would usually describe which thinking processes would occur and also the order of occurrence. The construction of a coding scheme would be based on identifying observable behaviours that correspond to the processes from the model.

According to Meijer, Veenman and van Hout-Wolters (2006), the coding scheme should not have too many categories to capture finer behavioural details because the coding process might end up focusing on unimportant minute details rather than trying to make sense of what is meaningful. Moreover, the coding scheme might become unreliable since different coders might be confused by the subtle differences among similar codes to distinguish finer behavioural details, and thus end up assigning the same protocol to different categories. Thus exemplars of actual verbal protocols should be included to help the coders identify the behaviours correctly. As an illustration, specialising is a process in the investigation model. But how does one know when a student specialises? This happens when one observes a student trying some examples. So 'trying example' is a student's behaviour which can be observed to infer the process of specialising. In other words, a process may or may not be an observable behaviour during thinking aloud. Therefore, there was a need to code only observable behaviours, which in this case is just 'trying example'.

Examples of the development of coding schemes could be found in the empirical study of Foong (1990) described earlier in Section 2.3.4(f), and the research work of Meijer et al. (2006). What is common in the construction of these coding schemes is a combination of a top-down (theoretical driven) and a bottom-up (empirical driven) strategy. Foong (1990) first identified five problem-solving categories (problem-orientation heuristics, problem-solution heuristics, domain-specific knowledge, metacognition, and affective behaviours) with a list of 40 possible behaviours from existing literature. Then the coding scheme was refined based on students' verbal protocols, and the final coding scheme consisted of 28 prominent behaviours in the five problem-solving categories. Similarly, Meijer et al. (2006) constructed a taxonomy of metacognitive processes with six categories: orientation, planning, execution, monitoring, evaluation and reflection. They identified metacognitive behaviours for each category from thinking-aloud protocols in current literature and then revise these behaviours based on empirical data collected before arriving at a final coding scheme.

2.4.4 Method for Development of Processes: Teaching Experiment

To study the development of processes in mathematical investigation, there is a need to review literature on methods used to develop these processes. Clement (2000) believed that research on students' thinking processes is most fruitfully undertaken using the thinking-aloud method and teaching experiments.

(a) Teaching Experiment vs. Classical Experimental Design

A teaching experiment is different from the classical experimental design in various ways. Although both research methods involve teaching, the latter relies strongly on a psychometric pretest and posttest to measure the effect of an intervention programme. "Psychometrics was founded upon the idea that a student's actual score on an item is composed of a ‘true’ score and some amount due to error.” (Steffe & Thompson, 2000, p. 272). Usually, a quasi-experimental design will have an experimental group and a control group (Burns, 2000). The experimental group will be subjected to some new teaching method while the control group will be taught in the traditional way. Both groups sit for a pretest and a posttest with parallel test items. The pretest was to ensure that the performances of both groups in the content tested in the pretest are similar at the start of the experiment. The posttest will measure whether there is any significant difference in the test score for the experimental group compared with the control group after the intervention programme. In other words, a quasi-experiment design is essentially a quantitative research methodology, although in recent research, the pretest and the posttest can contain items that require a qualitative analysis.

However, a teaching experiment is primarily a qualitative research design that does not depend on a psychometric pretest and posttest. A teaching experiment consists of a sequence of teaching episodes (Steffe, 1983). “A teaching episode includes a teaching agent, one or more students, a witness of the teaching episodes, and a method of recording what transpires during the episode.” (Steffe & Thompson, 2000, p. 274) The records can be used to prepare subsequent episodes and to conduct a retrospective conceptual analysis of the teaching experiment. English, Jones, Lesh, Tirosh and Bussi (2002) believed that the focus of teaching experiments should be on development in mathematically-enriched environments such as those involving problem solving and mathematical investigation, and not just in traditional classroom teaching environment. Civil (2002) described the distinguishing characteristics of such a classroom environment in which students solve problems and investigate mathematics like mathematicians: (a) collaboration in small groups on challenging mathematical tasks, (b) the students are encouraged to develop and share their strategies, and to be persistent in the mathematical tasks, (c) mathematical discussions and communication among the students and with the teacher, and (d) the students are responsible for decisions concerning validity and justification.

Sometimes there is a need to observe and record the processes exhibited by students during problem solving or investigation, especially when the purpose of the teaching experiment is to develop such processes (Lesh et al., 2000). A method to track a student's thinking processes is to get the student to think aloud as described earlier in Section 2.4.2. Since it is practically impossible to videotape all the students' thinking processes during a teaching episode as the classroom will be too noisy if every student thinks aloud, it might be necessary to record the thinking processes of each student separately when they think aloud during problem solving or investigation in a pretest and a posttest. However, a teaching experiment, coupled with a pretest and a posttest, appears to be similar to a quasi-experimental design. An important difference is that the pretest and the posttest for the teaching experiment are not psychometric tests to measure the students' scores (Steffe & Thompson, 2000), but a means for studying the students' processes before and after the teaching experiment qualitatively, although it is still possible to use a scoring rubric to assess the processes and analyse the scores quantitatively. Other differences include using records of what transpires in a teaching episode to inform subsequent episodes in a teaching experiment, and the focus of a teaching experiment to develop processes in a mathematically-rich environment, which were described earlier.

(b) Development of Processes

Researchers are divided on whether processes associated with problem solving and investigation can be taught or whether they are only developed through meaningful experiences (Frobisher, 1994), and whether or not processes can be transferred to other situations (Lampert, 1990). Orton and Frobisher (1996) believed that communication processes, such as explaining, can only be developed through repeated usage over a long period of time, and they suggested that students should be introduced to working with different processes a few at a time. But they also believed that other processes, such as recording processes, need to be taught. In the same way, Lampert (1990) also recognised that conventional mathematical tools, including language and symbols, need to be taught because they are useful for communicating mathematics and for reasoning, but she did not ‘force’ them down on her students. Instead, she negotiated their meaning with her students, according to the true spirit of discourse in a mathematical community.

Lampert’s (1990) teaching experiment has already been discussed in Section 2.1.5(d). It has shed light on what a teacher should do in a classroom environment that develops mathematical processes in problem solving and investigation. The teacher is to facilitate and guide the students in their investigation and discussion. If the students are stuck, the teacher will not tell them the answer but prompt them with appropriate questions when necessary. During the class discourse, the teacher’s duty is to encourage the students to participate by telling the class what they have found, even if they are not entirely sure that it is correct. “Until the group arrived at a mutually agreed-upon proof that one or more of the answers must be correct, all answers were considered to be hypotheses” (ibid., p. 40). The teacher will ask the students to give reasons why they think a particular hypothesis is correct or wrong. If the hypothesis is found to be false, the student who gives the answer is free to respond with a revision. The teacher will emphasise that it is alright to make mistakes because no one is perfect. The teacher will tell the students that if they change their minds about their conjecture, they can revise it by saying, "I want to revise my thinking" (p. 52), a phrase that Lampert taught and encouraged her students to use.

Many educators (e.g. Cobb, 1991) believe that the teacher may still have to provide some guidance when necessary because not all students can discover mathematics on their own. The difference is what kind of guidance. As explained in the preceding paragraph, Lampert (1990) guided her students by asking some appropriate questions without telling. Similarly, Tanner (1989), whose study was described in detail in Section 2.1.5(b), found that "the most effective interventions question students rather than dictate to them" (p. 22). On the other hand, some educators (e.g. Noddings, 1990) believe that there may also be a need to tell and explain to the students, but "how much to tell and explain really depends on the teachers' sensitivity to the needs of their students" (Teong, 2002, p. 42). Therefore, it is a fine balance between stimulating mathematical thinking by asking suitable questions when students are stuck, and explaining to the students when really necessary.

2.4.5 Summary of Literature Review on Research Methodologies

The review of relevant literature on research designs in Section 2.4 has suggested that the thinking-aloud method is useful to capture a small sample of students' cognitive and metacognitive processes during mathematical investigation, despite some of its disadvantages which might be alleviated to a certain extent by letting the students practise thinking aloud. The literature review has also informed the present study on the construction of a coding scheme and the use of protocol analysis to study the types and interactions of thinking processes. The review has brought out the importance of an inter-coder reliability test to strengthen the validity and reliability of the coding scheme. A teaching experiment was found to be suitable to develop students' processes in mathematical investigation, and empirical studies that used teaching experiments have offered insights into how to conduct teaching episodes where the teacher acts as a facilitator to guide students in their investigation.

2.5 CONCLUDING REMARKS FOR THIS CHAPTER

Chapter 2 has reviewed the current state of mathematical investigation in research literature. It has examined some key issues and clarified conflicting views between investigation and problem solving, which will help to define the terminologies used in the present study clearly. The review of literature on cognitive and metacognitive processes has aided in identifying the types and interactions of these processes, which will inform the development of two theoretical models to study the nature of thinking processes in mathematical investigation. An extensive search of existing literature has revealed a gap in empirical studies on the processes in investigation, which has served to set the direction of research for the present study in order to address this gap. The review of relevant research methodologies has paved the way for developing a suitable research design to answer the research questions. Chapter 3 will then describe the research methodology for the present study.

PART TWO: RESEARCH METHODOLOGY AND DEVELOPMENT OF DATA ANALYSIS TOOLS

Part Two of this thesis describes the research methodology and the development of data analysis tools for the present study. It consists of three chapters. Chapter 3 explains the research methodology, Chapter 4 outlines the construction of the coding scheme for coding the students' thinking-aloud protocols during investigation, and Chapter 5 describes the design of some data analysis instruments.

CHAPTER 3: RESEARCH METHODOLOGY

The review of relevant literature in Chapter 2 provided the theoretical basis for the conceptualisation of the present research on the nature and development of processes in mathematical investigation for Secondary 2 students. This chapter will explain the research methodology used to carry out the current study. It will describe the theoretical background, two theoretical investigation models, an initial exploratory study that informed the direction of the present research, the research questions and definitions of terms, the research design, the pilot study, the main study and the quality of the present research.

3.1 THEORETICAL BACKGROUND

The review of relevant literature in the previous chapter has provided the theoretical background for the present research on the nature and development of cognitive and metacognitive processes in mathematical investigation. The theoretical basis for including mathematical investigation in school mathematics education is the belief of the importance of mathematical problem solving and academic mathematics. “The idea of investigation is fundamental both to the study of mathematics itself and also to an understanding of the ways in which mathematics can be used to extend knowledge and to solve problems in many fields” (Cockcroft, 1982, p. 73), which is why many school curricula in countries all over the world (e.g. Singapore, USA, UK, Australia and New Zealand) include not only mathematical problem solving but mathematical investigation. Other educators (e.g. Brenner & Moschkovich, 2002; Lampert, 1990) also believe in bringing the practices of academic mathematicians, who investigate and solve genuine problems, into the mathematics classroom (see Section 1.1).

The study of cognitive processes in mathematical investigation is based primarily on the art of problem posing by researchers such as Brown and Walter (2005), Kilpatrick (1987), Krutetskii (1976) and Frobisher (1994); and the four main mathematical thinking processes by Mason et al. (1985), namely, specialising, conjecturing, justifying and generalising (see Sections 2.1.2 and 2.2.3). The theoretical basis for the study of metacognitive processes is based upon the works of Flavell (1976), Schraw (2001) and Schoenfeld (1985, 1987); and the focus for the present study is on the regulation of mathematical investigation behaviours (see Section 2.3). The collection of data on the nature of thinking processes is based on the thinking-aloud method developed by Duncker (1926, as cited in Foong, 1990), which is also used extensively by other researchers (see Section 2.4.2). The analysis of verbal protocols for thinking aloud is based on the works of Newell and Simon (1972), which is used extensively by other researchers as well (see Section 2.4.3). The methodology for developing processes in mathematical investigation is a teaching experiment (see Section 2.4.4). Clement (2000) believed that one of the most important needs in basic research on students' thinking processes is the need for insightful explanatory models of these processes, and that such basic research is most fruitfully undertaken using the thinking-aloud method and teaching experiments.

These theoretical perspectives form the basis underpinning the present study in researching on the constructs of cognitive and metacognitive processes in mathematical investigation, and for developing the research methodologies of the teaching experiment and thinking aloud.

3.2 THEORETICAL MATHEMATICAL INVESTIGATION MODELS

Based on the literature review in Chapter 2, two theoretical investigation models were developed to describe the interactions among the processes. The first model displayed the interactions among the cognitive processes while the second model displayed the interactions among the metacognitive and the cognitive processes.

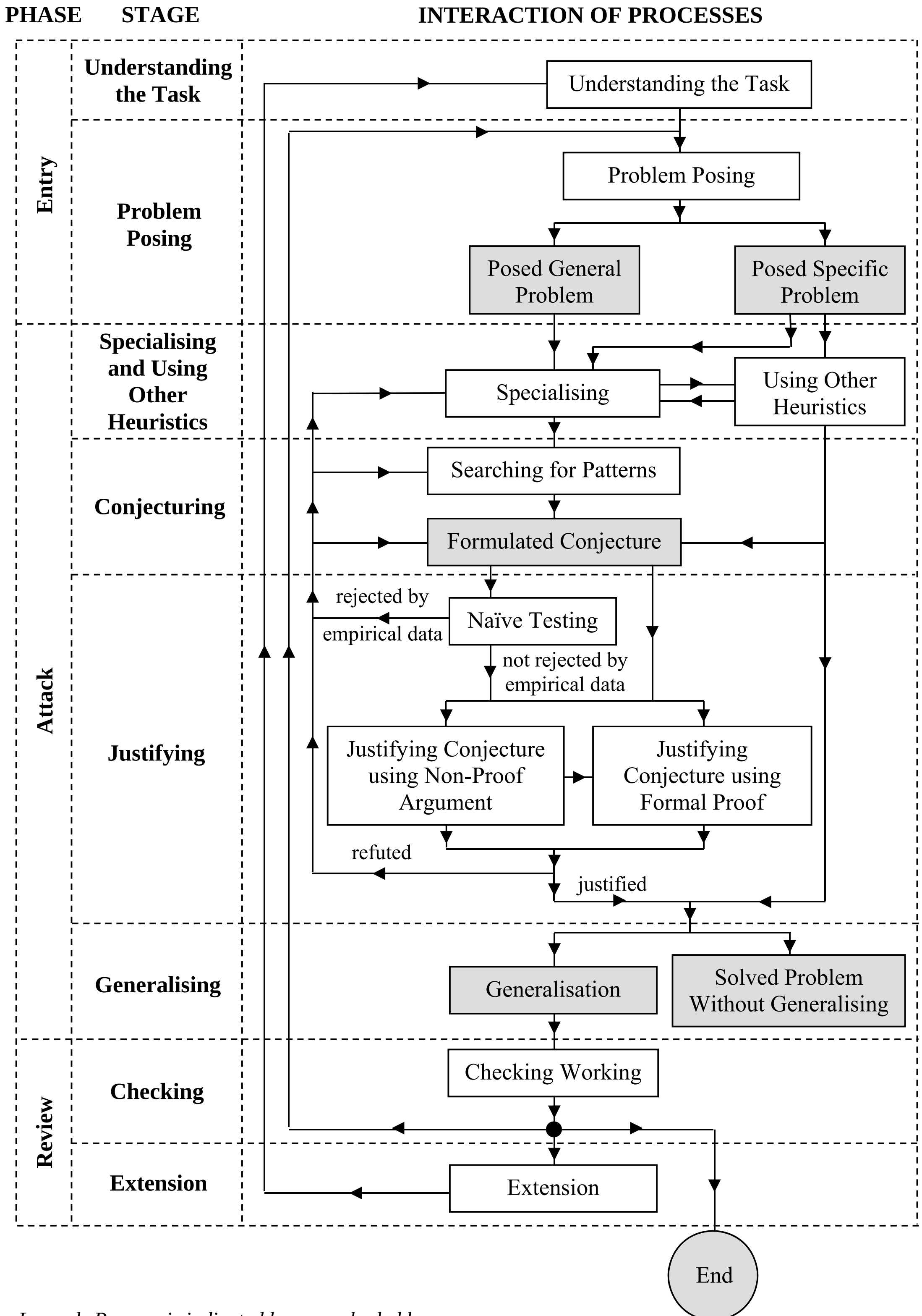
3.2.1 Theoretical Investigation Model for Cognitive Processes

The purpose of the first investigation model was to provide a theoretical framework to study the interactions of the various cognitive processes. The findings of the current study will then be used to refine the model if necessary. The need for such an insightful explanatory model of students' thinking processes is one of the most important current needs in research on these processes (Clement, 2000). This type of explanatory models is often iconic in nature and the purpose of these models is to give satisfying explanations for patterns in observations (Lesh et al., 2000).

The literature review in Section 2.2 has suggested that there are three phases and eight stages in mathematical investigation as shown in Table 2.1 on page 61. Each stage is named after the main process(es) that it contains. A main process may contain smaller processes called sub-processes. As explained earlier in Section 2.2.3, there is a need to include in the investigation models certain important outcomes that may affect how the processes interact with one another. The two types of investigative tasks (Type A and Type B) were discussed in Section 2.1.4.

(a) Description and Explanation of Theoretical Model for Cognitive Processes

Based on the analysis of various problem-solving and investigation models in existing literature in Section 2.2.2, and the review of literature on the processes and outcomes of investigation in Section 2.2.3, a theoretical mathematical investigation model for cognitive processes was developed for the present research (see Fig. 3.1). However, there was a need to modify existing investigation models as they were unable to describe all the processes studied in the current research. For example, most models suggest that the only approach in mathematical investigation is to search for any pattern by going through the inductive pathway of specialising, conjecturing, justifying and generalising. But the prescriptive model for the present study had taken into account that students can bypass the inductive pathway by posing specific problems for Type B tasks, and then go through the deductive pathway of using other heuristics to solve these problems without formulating any conjecture and without generalising (see Section 2.1.4). The model for the current study also included the extension stage which was missing in most investigation models. Another example is that most models either show a linear route with an end point or a cyclic path that goes on forever. But the model for the current study is multi-cyclic in the sense that it has more than one cycle. In fact, problem solving and investigation are not just a cyclic process because it is not just one simple cycle (Love, 1988).



Legend: Process is indicated by an unshaded box
Outcome is indicated by a shaded box or circle

Figure 3.1 Theoretical Investigation Model for Cognitive Processes

In the first phase, the *Entry Phase*, there are two stages. In Stage 1 (Understanding the Task), students should try to understand the task by reading the task statement carefully, highlighting key information and visualising the given information by drawing a diagram when appropriate (Pólya, 1957). In particular, for Type A tasks, as discussed in Section 2.2.3(b), it is often useful to try a few random examples to make sense of the task (Mason et al., 1985).

In Stage 2 (Problem Posing), students should pose problems to solve. Thinking of what problems to pose is a process. At the end of the process are two possible outcomes: posed the general problem of searching for any pattern, or posed a specific problem to solve. Based on the literature review in Section 2.2.3(b), students just need to search for any pattern for Type A tasks although they can also pose more specific problems to solve; but for Type B tasks, there is a need to pose a specific problem to solve. This will in turn affect, to a certain extent, what students will do in the next stage: specialise or using other heuristics.

In the second phase, the *Attack Phase*, there are four stages. In Stage 3 (Specialising and Using Other Heuristics), students generally need to specialise for Type A tasks, i.e. try examples, systematically to search for patterns (Mason et al., 1985). However, as explained in Section 2.2.3(c), students can solve the specific problem posed for Type B tasks by specialising or by using other heuristics; or they can alternate between specialising and using other heuristics, as shown by the corresponding pathways in the theoretical model in Figure 3.1.

Stage 4 (Conjecturing) consists of the sub-process 'searching for patterns' and the outcome 'formulated conjecture'. As explained in Section 2.2.3(c), when students observe a pattern, the outcome ‘observed pattern’ is only a conjecture. But students can also formulate a conjecture by using other heuristics, such as reasoning, without searching for any pattern. Therefore, the model will use the term ‘formulated conjecture’ as an outcome to describe both scenarios, rather than the term ‘observed pattern’ which only applies to the first case of observing a pattern from specialising.

Stage 5 (Justifying) contains three sub-processes: (i) naïve testing, (ii) justifying a conjecture using a non-proof argument, and (iii) justifying a conjecture using a formal proof. As explained in Section 2.2.3(e), naïve testing was advocated by Lakatos (1976) for refuting a conjecture by counter examples. However, students can bypass naïve testing and go straight to justifying a conjecture using a non-proof argument or a formal proof. If a conjecture is refuted, the pathways in Figure 3.1 show that the students will either have to go back to specialise some more, search for new patterns, or reformulate the conjecture.

Stage 6 (Generalising) consists of only two outcomes. As explained in Section 2.2.3(f), generalising overlaps with conjecturing and justifying, so to avoid confusion, this stage is concerned with whether generalisation has occurred. If a conjecture is formulated from specialising and proven correctly, then generalisation has occurred. But if a conjecture is formulated from using other heuristics (see pathway in Fig. 3.1 from ‘Using Other Heuristics’ to ‘Formulated Conjecture’), this will lead to ‘Solved Problem Without Generalising’ when the conjecture is proven. However, students may also use other heuristics to solve a problem without formulating any conjecture. The pathway will then proceed directly from ‘Using Other Heuristics’ to the generalising stage.

In the third phase, the *Review Phase*, there are two stages. In Stage 7 (Checking), the students should check their working after solving a problem. As explained in Section 2.2.3(g), the students should occasionally check their working even before they finish solving a problem. This means that the sub-process ‘Checking Working’ could occur in other stages, but the checking stage in the *Review Phase* is for after the students have finished solving a problem.

As explained in Section 2.2.3(h), there are three possibilities after Stage 7: (i) pose more problems to solve without changing the given, (ii) extend the task by changing the given, which is Stage 8 (Extension), or (iii) end the investigation. For Type A tasks, extending the task would usually result in an entirely new task with different patterns. For Type B tasks, students are expected to extend the task by trying examples (specialising) to search for patterns (conjecturing), and to justify the observed patterns so as to generalise. This means that the processes for Type B tasks during extension (specialising, conjecturing, justifying, generalising) are similar to the usual processes for Type A tasks. But for each example that the students generated during the extension of Type B tasks, they would need to solve it using other heuristics, such as reasoning, just like what they would have done for the original task. In other words, specialising and using other heuristics overlap during the extension of Task B tasks.

(b) Caveats: What the Theoretical Model is Not Trying to Illustrate

It is necessary to highlight a few caveats. First, the inductive and deductive pathways of the theoretical model are not mutually exclusive. Students can move from one pathway to the other pathway as shown by the arrows in the model in Figure 3.1, e.g.

a student may use other heuristics and go down the deductive pathway, but later formulate a conjecture and switch to the inductive pathway. Moreover, as explained in Section 2.2.3(e), the inductive pathway also includes reasoning during the justifying stage when students try to prove their conjecture using a non-proof argument or a formal proof.

Secondly, the theoretical model does not show all the sub-processes within each stage. For example, it does not show all the sub-processes in the understanding stage, e.g. highlighting key information, visualising information and trying examples. Instead, all these sub-processes will be identified in the coding scheme (see Chapter 4 later).

Thirdly, the theoretical model illustrates the *logical* sequences of processes that students may go through in an investigation. For example, students are supposed to check their solution, but they may choose not to do so. Moreover, students can change their minds anytime. For example, the students may be trying examples to search for patterns when they suddenly have an insight and decide to change direction by posing another specific problem. All these other possible types of pathways will not be shown in the theoretical model since students can go from any process to almost any other process in the model. However, the *actual* pathway that a student takes during an investigation will be traced using the Investigation Pathway Diagram (see Section 5.2 later).

Nevertheless, it is still possible that some of the logical pathways may still be missing from the theoretical model despite careful considerations, and there may be other important sub-processes that have not been identified. Therefore, one of the purposes of the present study is to identify the actual interactions among these processes when students engage in mathematical investigation so that the theoretical model can be validated and refined, if necessary, to reflect the actual interactions more accurately.

3.2.2 Theoretical Investigation Model for Metacognitive Processes

The second investigation model describes the interactions between the metacognitive and the cognitive processes. For simplicity, it will just be called the theoretical investigation model for *metacognitive processes*, to distinguish it from the first model which is the theoretical investigation model for cognitive processes. Based on the review of literature on metacognitive processes in Section 2.3, it was found that metacognition consists of knowledge of cognition and regulation of cognition (Brown et al., 1983; Schraw, 2001). As it is beyond the scope of this thesis to study how metacognitive knowledge affects investigation, the present research will only focus on metacognitive regulation, which consists of planning, monitoring and evaluating. Since there was a lack of both theoretical and empirical research on metacognitive behaviours during investigation, I had to hypothesise the types of metacognitive processes that could occur during investigation, and their interactions with the main cognitive processes. Only five metacognitive processes were identified:

- Analysing feasibility of goal (planning)
- Analysing feasibility of plan (planning)
- Monitoring understanding (monitoring)
- Monitoring progress (monitoring)
- Reviewing solution (evaluating)

Analysing the feasibility of the goal or problem is an important metacognitive process for mathematical investigation because students need to set their own goals by posing problems to solve, and so they should analyse whether their goals or problems are worth pursuing, or too trivial or too difficult to pursue. Similarly, they should think of a plan to solve a problem that they have posed and to analyse the feasibility of their plan. Monitoring progress is a common metacognitive process described in the literature review in Section 2.3.1, while monitoring understanding and reviewing the solution to see if it has met the goal of the task have been discussed in Section 2.3.2.

Table 3.1 shows the main cognitive processes and metacognitive processes that could occur in the various investigation stages. As explained in Section 3.2.1(a), the issue during the generalising stage is whether the result obtained after justification is a general one, so there is no process in this stage. Thus there is a need to combine the stages of justifying and generalising in the following table.

Table 3.1 Cognitive and Metacognitive Processes in Mathematical Investigation

Phases	Stages	Cognitive Processes	Metacognitive Processes
Entry	Understanding the Task	Understanding the Task	• Monitoring Understanding
	Problem Posing	Problem Posing	• Analysing Feasibility of Goal
Attack	Specialising and Using Other Heuristics	Specialising	• Analysing Feasibility of Plan • Monitoring Progress
		Using Other Heuristics	• Analysing Feasibility of Plan • Monitoring Progress
	Conjecturing	Conjecturing	• Analysing Feasibility of Plan • Monitoring Progress
	Justifying / Generalising	Justifying / Generalising	• Analysing Feasibility of Plan • Monitoring Progress
Review	Checking	Checking	• Reviewing Solution
	Extension	Extension	• Analysing Feasibility of Goal

Figure 3.2 shows the theoretical investigation model of metacognitive processes. The model displays how the metacognitive processes interact with the main cognitive processes in investigation. It was discovered that the first metacognitive process in chronological order is not a planning process but a monitoring one: 'Monitoring Understanding'. It is followed by a planning process, 'Analysing Feasibility of Goal', which acts on the cognitive process of problem posing, and also on the cognitive process of extension at a later stage. Both 'Analysing Feasibility of Plan' and 'Monitoring Progress' act on the main mathematical thinking processes of specialising, conjecturing, justifying and generalising. The last metacognitive process, 'Reviewing Solution', acts on the checking process.

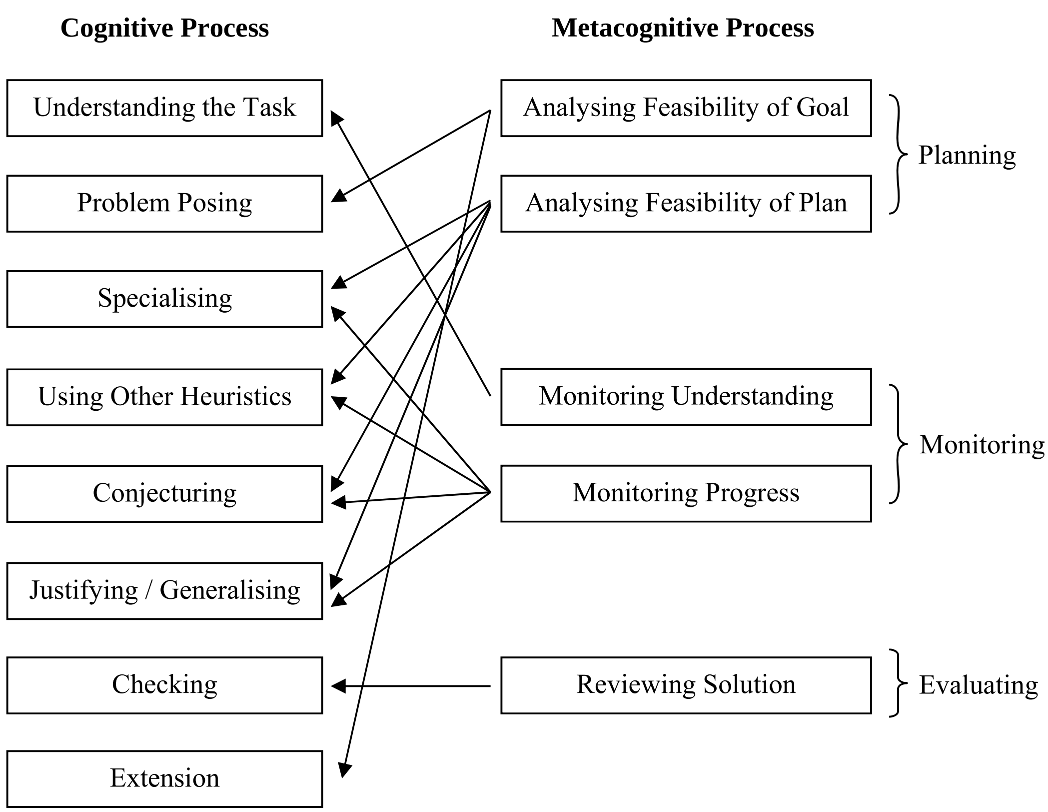


Figure 3.2 Theoretical Investigation Model for Metacognitive Processes

3.3 INITIAL EXPLORATORY STUDY TO SET DIRECTION OF RESEARCH FOR PRESENT STUDY

This section outlines an initial exploratory study (*not* the pilot study) conducted to set the direction of research for the present study. The exploratory study was necessary for the following reasons. The initial plan of the current research was to cover both the breadth and the depth. To cover the breadth, the first part of the plan was to study the thinking processes of a large group of secondary school students with different gender, grade levels and academic abilities using a paper-and-pencil test instrument containing a few investigative tasks, so as to identify the nature of these processes and compare them among the different types of students. To cover the depth, the second part of the plan was to study the development of processes for one class of students using a teaching experiment, a pretest and posttest.

However, there was doubt about the feasibility of the first part of the plan since anecdotal evidence suggests that students do not know what to do when given investigative tasks containing only the word 'investigate', and so they would not exhibit enough processes to shed light on their nature. Due to the lack of empirical research on investigation, there was no concrete evidence to support or disprove the above anecdotes. Thus there was a need to conduct an exploratory study to find out the state of investigation proficiency among the students in Singapore because this will impact on the direction of research for the present study. Jaworski (1994) believed that it is very important to document clearly the reasons for any decision made in a research study.

To plan for the paper-and-pencil test instrument for the first part of the initial plan, *another* small study was conducted with a convenience sample of 21 pre-service (or trainee) secondary mathematics teachers to find out if they understood what it meant to investigate when given an investigative task without any teaching or instruction, and if not, how to modify the test instrument for a large sample. The test instrument consisted of two parts (see Appendix A). The first part was open, and after the pre-service teachers had tried for 15 minutes, it was discovered that most of them did not know *what* to investigate. So the second part of the test instrument was given to them: it was the same task but with the question "Which numbers are polite?" This time, the pre-service teachers knew what to investigate but their answer scripts suggested that most of them did not know *how* to investigate to find out which numbers are polite.

The implication of this small study was that secondary school students may not know what or how to investigate when given an investigative task, so the test instrument for the exploratory study (see Appendix B) was modified to include scaffolding questions for the first task (e.g. find as many patterns as you can about the powers of 9), and subsequently, the other two tasks (one Type A task and one Type B task) were made more open by asking students to pose their own problems to investigate.

The sample for the exploratory study was 29 Secondary 1 (equivalent to Grade 7) students from an intact class from one of the high-performing local secondary schools. In Singapore, all secondary schools are ranked according to the mean subject grades of the General Certificate of Education Ordinary Level (GCE O-Level) examination results every year. The high-performing secondary schools usually attract the higher-achieving primary school students, which are selected for entry to the secondary schools based on their Primary School Leaving Examination (PSLE) results. PSLE is a national exam that all primary school students take at the end of their primary school education, and the subjects tested include mathematics. Thus the sample of Secondary 1 students from one of the high-performing secondary schools could be considered as high-achieving students. The students were not streamed further into different classes according to their PSLE results in Secondary 1 and the intact class was chosen randomly from the whole Sec 1 cohort in that school. If the students could do the investigation, the test instrument would be given to a class of average students from another school, and to a class of low-achieving students from a third school, to see if students with different academic abilities could cope with such investigation.

However, findings from the class of 29 high-achieving students show that they were unable to cope with investigation, even with scaffolding built into the first question. Within five minutes from the start of the test, five students raised their hand and asked me, who was also the invigilator, what they were supposed to do for the first task. Other students did not ask me but some of them (all names are pseudonyms) wrote in the written survey after the test:

> Albert: I am thinking of asking the teacher what 'investigate' means.
>
> Ben: I find it a bit difficult as I do not understand the meaning of investigate.

Analyses of the students' answer scripts for the first task show that 35% of the students did not know how to investigate by specialising, and so they did almost nothing or nothing at all. Although 17% of the students did try something, they were unable to discover a single pattern, not even the simplest patterns, such as, all the powers of 9 are odd, or the last digit alternates between 1 and 9. This suggests that at least half of the students (52%) did not know how to investigate. Although the other 48% did find some patterns, many of the discoveries were very trivial, such as, powers of 9 are divisible by 9, and when a power of 9 is divided by 9, the result is the preceding power of 9. It seems that telling the students to find *any* pattern about powers of 9 when there was no specific problem to solve had confused them. The other two tasks were also badly done. Providing a specific problem in Task 3 (how many matches will there be if there are 20 teams in the tournament?) did not help the students to pose other problems to investigate in the subsequent part. What were more worrying were the very negative comments from most of the students:

> Chris: I feel it is pretty useless and a waste of my time.
>
> Dan: I think I am going to fail. I think that I am going to be scolded as I don't know how to do the questions as it is so #☠!!@ing hard. I think that [the test] was good as it help [*sic*] me to find out that I'm dumb

The exploratory study provided some credence to the anecdotal evidence that the students in Singapore do not know what and how to investigate when given an investigative task, and providing suitable scaffolding in the task statement of the first task did not help them to understand the task requirement of subsequent open investigative tasks. To carry out the large-scale study of the initial plan will also do more harm than good because it will make more students dislike investigation, judging by the negative comments from the students described in the preceding paragraph. Thus it was decided not to carry on with the exploratory study for a second class of average students and a third class of low-achieving students. Hence, the present study will focus only on a small group of students to study the nature and development of their processes in depth. Moreover, the exploratory study had also helped support the decision that the students for the present study must be given a familiarisation lesson *before* doing the pretest to teach them what to investigate, or else there will definitely be an improvement from the pretest to the posttest for the simple reason that the students did not understand the word 'investigate' during the pretest and thus could not proceed at all.

3.4 RESEARCH QUESTIONS

Based on the literature review in Chapter 2 and the direction of research provided by the exploratory study in Section 3.3, three research questions were formulated for the present research study:

RQ1: What is the relationship between the investigation pathways of Secondary 2 students and their outcomes across the two types of investigative tasks?

RQ2: What is the effect of the cognitive and metacognitive processes of Secondary 2 students on the outcomes of their investigation?

RQ3: What is the effect of the teaching experiment on the development of Secondary 2 students' mathematical investigation processes?

RQ1 will examine the nature of the processes from a macroscopic angle by looking at the investigation pathways, which depict the interactions of these processes, and their relationship with the outcomes. RQ2 will explore the nature of the processes from a microscopic viewpoint by observing the interactions between these processes and the outcomes. RQ3 will analyse the development of these processes.

3.5 DEFINITIONS OF TERMS

As the same term can be subject to various interpretations, and different terms can be used to mean the same construct in literature, it is important to define clearly how certain key terms are used in the present study. Thus this section will provide operational definitions for some important terminologies used in the current study based on the review of relevant literature in Chapter 2.

(a) Investigative Task

An 'investigative task' is a mathematical task that does not contain any problem in its task statement and ends with the word 'Investigate'. Thus it is open in the sense that students can pose any problem to solve. The present research will study the processes for two types of investigative tasks: Type A and Type B. The differences between the two types of tasks and the rationales for including both types in the present study have been dealt with in Section 2.1.4. Examples of the two types of tasks are:

> **Posttest Task 1 (Type A): Add Sum of Digits to Number**
>
> Choose any number. Add the sum of its digits to the number itself to obtain a new number. Repeat this process for the new number and so forth. Investigate.

> **Posttest Task 2 (Type B): Sausages**
>
> I need to cut 12 identical sausages so that I can share them equally among 18 people. Investigate.

(b) Mathematical Investigation

A 'mathematical investigation' is an *activity* undertaken by students when they attempt an investigative task (see Section 2.1.3). It includes processes such as understanding the task, problem posing, problem solving and extension. There are generally two approaches to the problem-solving process: the inductive approach using specialising (which is similar to the investigation *process*); and the deductive approach using other heuristics such as reasoning. Investigation, as a *process*, involves the four main mathematical thinking processes of specialising, conjecturing, justifying and generalising (Mason et al., 1985). Thus the present study distinguishes between investigation as an activity and investigation as a process. In general, when the term 'investigation' is used in the thesis, it will always refer to the activity, as contrast to the 'investigation process'.

(c) Processes, Outcomes and Pathways

The 'nature of processes' refers to the *types* of processes and the *interactions* among these processes. There are eight main cognitive processes identified for investigation: Understanding the Task, Problem Posing, Specialising, Using Other Heuristics, Conjecturing, Justifying, Generalising, Checking and Extension. Metacognition refers to the knowledge and regulation of one's own thinking or cognitive processes. As it is beyond the scope of the present research to study the effect of metacognitive knowledge on investigation, the current study will focus on the main metacognitive processes that regulate cognition. The interactions of these processes are displayed using the two investigation models developed for the present study (see Section 3.2). The *outcomes* of an investigation refer to the results obtained, including intermediate results such as problems posed, patterns observed, and conjectures justified. The investigation *pathways* refer to the interactions of the processes and outcomes as indicated by the paths in the investigation model for cognitive processes that go from one process or outcome to another.

(d) Teaching Experiment

In this thesis, a teaching experiment refers to a qualitative research design consisting of a sequence of teaching episodes, where the teacher acts as a facilitator to help the students develop their thinking processes in mathematical investigation. It is different from a quasi-experimental methodology as explained in Section 2.4.4.

3.6 RESEARCH DESIGN

This section will provide an overview of the research design for the present study in the form of a research framework, followed by a discussion of how the sample was selected. It will present the outlines of the familiarisation lesson and developing lessons, and the instructional strategies for the lessons. The rationales of the selection of the investigative tasks for the lessons, the pretest and the posttest will then be examined. Lastly, it will explain the methods of data collection and analysis.

3.6.1 Research Framework

Since the present research was a study on the nature and development of processes in mathematical investigation, a qualitative approach would be more appropriate than a quantitative one. The review of relevant literature on the various methods of data collection for processes and for development of processes was described in Section

2.4. Current research trends are moving towards detailed qualitative studies of individual students in order to gain insight into how the individual thinks or learns, and so the emphasis was not so much on generalising in quantitative studies because no two individuals are the same (Kelly & Lesh, 2000b). Clement (2000) observed that one of the most important needs in basic research on students' thinking processes is the need for insightful explanatory models of these processes, and that such basic research is most fruitfully undertaken using the thinking-aloud method and teaching experiments. In particular, the focus of teaching experiments should be on development in mathematically-enriched environments such as those involving problem solving and mathematical investigation (English et al., 2002).

Therefore, the research methodology of the present study was a teaching experiment (Steffe & Thompson, 2000) to develop the students' cognitive and metacognitive processes in mathematical investigation, and a pretest and a posttest using the thinking-aloud method (Clement, 2000) to study the nature of the students' thinking processes so as to inform the two theoretical investigation models developed for the current research. The proficiency of the students' performance in mathematical investigation during the pretest and the posttest was also measured using a scoring rubric that assessed both the processes and the outcomes.

Figure 3.3 shows the research framework for the present study. The students' initial investigation proficiency was modelled by the two theoretical investigation models for cognitive and metacognitive processes described in Section 3.2, and was measured quantitatively using a scoring rubric designed to evaluate the processes captured using the thinking-aloud method and the outcomes of a written pretest. Then the students underwent a teaching experiment to develop their investigation proficiency. The students' final investigation proficiency was again measured using the same scoring rubric for a parallel posttest. The two tests were also used to inform the theoretical investigation models qualitatively in order to obtain the refined investigation models.

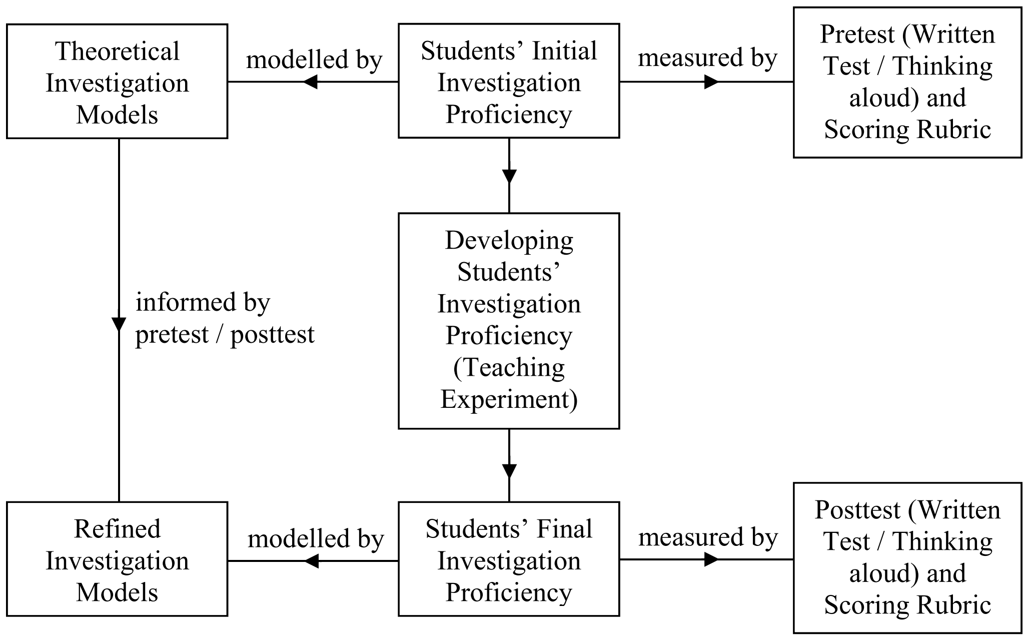


Figure 3.3 Research Framework for Present Study

3.6.2 Sample for Main Study

Since the present research was meant to be an in-depth study of the nature and development of processes, the sample in the main study was a small one consisting of 10 Secondary 2 (equivalent to Grade 8) students from another high-performing local school (different from the school where the initial exploratory study described in Section 3.3 was conducted). As the school could only provide me lower secondary students to choose from, and the main study was conducted in the first term of the year, Secondary 1 (equivalent to Grade 7) students were not selected since they would not have learnt number patterns and enough algebra, which would help them to search for patterns and justify conjectures using formal proofs involving algebra. Thus Secondary 2 students were chosen because they had learnt these topics by the end of Secondary 1. As explained in Section 3.3, students from a high-performing secondary school could be considered as high-achieving students. The reason for choosing high achievers was that they were more likely to exhibit a fuller and richer range of processes in investigation than average or low achievers (Tanner, 1989). In fact, the Final Coding Scheme had to be based on the empirical data obtained from the posttest of the main study because the processes exhibited in the pilot test and the pretest did not cover the whole range (see Section 4.6 later). These data were necessary to inform the theoretical investigation models, which could then be used to study the processes in average and low-achieving students in future research.

The sample for the main study was a purposeful sample. The first criterion for choosing the students from a few Secondary 2 classes was that they should not have any experience in mathematical investigation. The students were given a sample investigative task, which was Investigative Task 1 in Appendix D, and asked to record whether they knew how and what to investigate for this task (they were to answer 'Yes' or 'No'), and whether they had any experience with this type of investigation (they were to answer 'Yes', 'No', 'A little bit' or 'Some'). All students who answered 'No' for both questions were then identified. There were 23 of them. The second criterion was to choose among the 23 students a wide variety of achievement in mathematics based on their academic results. Although all these students scored A* (highest grade) for their Primary School Leaving Examination (PSLE) mathematics, the range of marks for an A* grade was still very wide (their exact PSLE mathematics scores were never made known). The big differences in mathematics achievement within this broad band of high-achieving students were suggested by their Secondary

One Final Examination mathematics scores shown in Table 3.2, which ranged from 47 to 89 marks out of 100 marks.

Table 3.2 Mathematics Examination Scores of Students with No Experience in Mathematical Investigation

	41-50	51-60	61-70	71-80	81-90	Total
Boys	1	1	4	2	4	12
Girls	0	2	0	4	4	11
Total	**1**	**3**	**4**	**6**	**8**	**23**

Table 3.3 Mathematics Examination Scores of Students in Sample

	41-50	51-60	61-70	71-80	81-90	Total
Boys	1	0	1	1	2	5
Girls	0	2	0	1	2	5
Total	**1**	**2**	**1**	**2**	**4**	**10**

As the scores for the 23 students were already skewed to the left and some students did not wish to participate in the research, it was not possible to choose the sample for the present study to contain the same number of students in each mark range. So the next best choice was to choose 10 students such that there was at least a student in each mark range for the sample (see Table 3.3). The purpose of this second criterion was to increase the possibility of observing a wider range of investigation behaviours among the students with different levels of achievement in mathematics. The third criterion was gender: half of them were chosen to be boys and the other half girls. Again it was not possible to get an equal number of boys and girls in each mark range. The purpose of this third criterion was to get a representative sample in terms of gender because there was not much research to suggest how gender would affect the processes in investigation, and so a balanced sample would increase the possibility of observing a wider range of investigation behaviours. However, it is beyond the scope of the present research to study the effect of gender on investigation.

3.6.3 Teaching Experiment

There were two parts to the teaching experiment for the present study. The first part consisted of a two-hour familiarisation lesson conducted by me to expose the students in the sample to mathematical investigation in order to teach them what and how to investigate when given an investigative task *before* the pretest was conducted. The rationale for this decision was described in detail in Section 3.3 where most students in the initial exploratory study did not understand the task requirement and so they were unable to even start attempting. Therefore, if the students in the present study were given the pretest before any lesson on investigation, there would definitely be an improvement from the pretest to the posttest for the simple reason that the students would not have understood the word ‘investigate’ during the pretest and thus could not proceed at all.

The second part of the teaching experiment consisted of five two-hour developing lessons after the pretest was conducted. The main purpose of these lessons was to develop the students’ processes during investigation. As explained in Section 2.4.4(b), many educators are divided whether these processes could be taught or developed. Since “there is much uncertainty whether processes can be taught, or whether they are only assimilated into a pupil’s repertoire through usage over a long period of time” (Frobisher, 1994, p. 161), there was no guarantee that the developing lessons in the current study would develop the students’ processes enough to show an increase in the posttest score. Thus, if the findings show an improvement in the score, it could then be attributed to the success of the developing lessons. This kind of research design, in which a familiarisation lesson was conducted before the pretest, will enhance the validity and reliability of the present study. All lessons were videotaped.

(a) Outlines of Familiarisation Lesson and Developing Lessons

The outlines of the familiarisation lesson (Lesson 1) and five developing lessons (numbered from Lessons 2 to 6 for continuity) are given in Appendix C, and the investigative tasks are given in Appendix D. The purpose of the familiarisation lesson was to teach the students in the sample what to do when given an investigative task: what they were supposed to investigate and how to investigate. But great care was taken not to intervene too much as this was only a familiarisation lesson, e.g. for the Investigative Task 2 (Handshakes) in Appendix D, the students were guided to realise that they could generalise for n workshop participants after finding the total number of handshakes for 20 participants; but it was not emphasised that they should generalise whenever possible since the purpose of the familiarisation lesson was not to develop in students the habit to generalise but to expose them to the possibility of generalisation. Therefore, the main purposes of the familiarisation lesson were:

- to familiarise the students with what to investigate: search for any pattern for Type A tasks, and pose specific problems to solve for Type B tasks;
- to teach the students that they should not accept an observed pattern as true: they must prove that it is the actual underlying pattern or refute it by counter examples;
- to teach the students that they can extend the task by changing the given;
- to teach the students that they can change the given in Type B tasks to generalise;
- to provide the students the opportunity to practise thinking aloud as this was required during the pretest.

Although the students should have had more practice in thinking aloud so that they would be better in articulating their thoughts during the data collection, they did not practise thinking aloud during the first hour of the lesson. This was because this was the first time the students were exposed to such investigation, so thinking aloud from the beginning might have distracted them from focusing on their investigation. The pilot study, which will be described in Section 3.7 later, suggests that practising thinking aloud during the second half of the familiarisation lesson was sufficient to get most students to think aloud during the pretest.

For the developing lessons after the pretest, the focus was in developing in students the various cognitive and metacognitive processes required in investigation. Orton and Frobisher (1996) suggested that students should be introduced to working with different processes a few at a time. Therefore, the students were taught only one or two main processes during each of the developing lessons. The main purposes of the developing lessons were as follow (see Appendix C for the outlines of the lessons):

- to guide the students to investigate the pretest tasks further, focusing on what they could have investigated, e.g. what were some patterns for Pretest Task 1 and how to generalise for Pretest Task 2 (Lesson 2);
- to develop the students' understanding process by reminding them of various strategies that they had learnt during their normal school lessons to understand textbook questions, e.g. read the task carefully, re-read or rephrase the task statement, highlight key information, and visualise given information by drawing a diagram; and to teach them a new understanding process: try examples randomly to make sense of Type A tasks (Lesson 2);

- to develop the students' problem-posing process, especially for Type B tasks since students were expected to just search for any pattern for Type A tasks; and to convince them that it was alright to pose a difficult problem that they could not solve (Lesson 3);
- to teach the students to analyse the feasibility of their goal or problem to see if it was worth pursuing, or too trivial or too difficult to pursue (Lesson 3);
- to develop the students' problem-solving heuristics, especially specialising systematically to search for patterns (conjecturing) and using reasoning to solve problems which may result in the formulation of conjectures (Lesson 4);
- to teach the students how to regulate their investigation using metacognitive strategies such as analysing the feasibility of their plan during specialising, using other heuristics, and conjecturing, to see if the plan was worth pursuing, and to monitor their own progress every 5 minutes or so (Lesson 4);
- to teach the students that certain results were actually conjectures to be proven or refuted, and to develop the students' justifying process, such as refuting conjectures by counter examples (naïve testing), and justifying conjectures using a non-proof argument or a formal proof (Lesson 5);
- to teach the students to analyse the feasibility of their plan during justifying, and to develop in the students the habit of monitoring their own progress (Lesson 5);
- to teach the students to extend Type B tasks to generalise, and to develop in the students the habit to extend and generalise whenever possible (Lesson 6);
- to develop the students' checking process, such as checking their working step by step occasionally, or by working backwards, or by examining whether the answer was reasonable or logical (Lesson 6);

- to teach the students to always review their solution after solving a problem to see if it had met the goal of the task, to evaluate the efficacy of their method of solution, and to look for alternative methods (Lesson 6);
- to teach the students to analyse their plan of attack when they were stuck, and in particular, the need to incubate (Lesson 6);
- to provide the students more opportunity to practise thinking aloud so that they could articulate their thoughts better in the posttest (Lessons 5 and 6).

(b) Instructional Strategies for Lessons

I was the teacher for all the lessons. The witness of the teaching experiment (Steffe & Thompson, 2000) was the school teacher who settled all the administrative work and logistics for the research, and who videotaped all the lessons using a videocam. The general instructional strategies for both familiarisation and developing lessons were about the same. The students were given an investigative task to investigate individually for 10 minutes and they were to record their rough working and observations in the answer script provided. If they had any query during this period, they could ask the teacher, but the teacher would not tell them the answers; instead, the teacher would ask appropriate questions to guide the students to think for themselves. After the individual work, the students would discuss with a partner seated beside them for another 10 minutes before the teacher facilitated the class discourse for 10 minutes. The students would then continue their investigation, either individually or with the same partner, for another 10 minutes, before another round of class discourse for 10 minutes. Finally, for the remaining 10 minutes of the hour, the students would summarise their findings on the answer script provided. This whole procedure was repeated for the second hour of the lesson with another task.

The teacher's duty was to facilitate and to guide the students in their investigation and discussion. The teacher followed what Lampert (1990) did for her classroom discourse, which has been discussed in detail earlier in Section 2.1.5(d) and 2.4.4(b). During the class discourse, the students were encouraged to participate by telling the class what they had observed or found, even if they were not entirely sure whether it was correct or not. The students were then asked to give reasons why they thought a particular conjecture was correct or incorrect. If the conjecture was found to be false, the student who gave the answer was free to respond with a revision. It was emphasised that it was alright to make mistakes because even great mathematicians make mistakes or travel down false trails before they make important discoveries. The students were taught that if they changed their minds about their conjecture, they could revise it by saying, "I want to revise my thinking" (Lampert, p. 52), a phrase that Lampert taught and encouraged her students to use.

3.6.4 Selection of Investigative Tasks for Lessons, Pretest and Posttest

In this section, the rationale for choosing the investigative tasks for the six lessons and the tests will be explained. Since the purpose of the present study was to examine the actual processes engaged by students during investigation, tasks that required students to investigate and do research outside curriculum time, e.g. project work and real-life mathematical modelling tasks discussed in Section 2.1.4(d), would not be suitable as it was not possible to capture and track such processes. Therefore, only the two types of pure-mathematics investigative tasks defined earlier in Section 3.5 will be selected for the present research. The tasks used in the present study were modified from investigative tasks found in academic books and resource books such as Bastow et al.

(1991), Biddle, Savage, Smith and Vowles (1988), Height (1989), Ho and Soon (1998), Holding (1991), Mason (1999), Mason et al. (1985) and Mottershead (1985).

(a) Selection of Investigative Tasks for Lessons

As explained earlier in Section 2.1.4, it was necessary to examine two types of investigative tasks (Type A and Type B) in the current study because the two types of tasks tend to elicit different kinds of processes. In general, Type A tasks involve posing the general problem to search for any pattern, specialise to investigate, and there is no need to extend the task partly because there are many patterns to find in the original task and partly because extension would usually result in a new task with completely different patterns. On the other hand, Type B tasks usually involve posing specific problems, use other heuristics such as deductive reasoning to solve, and there is a need to extend the task to generalise. Thus the lessons should contain a mixture of Type A and Type B tasks. Since there was only enough time in each lesson to attempt two tasks, then one task would be of each type. Appendix D shows the list of all the investigative tasks used in the six lessons.

(b) Selection of Investigative Tasks for Pretest and Posttest

Based on the above explanation, the pretest and the posttest should also contain a mixture of Type A and Type B tasks so as to elicit a full range of processes. As investigation was time consuming, a student would need at least 30 minutes to attempt each task in order to discover something significant. Since the duration of the test should not be too long, the pretest contained two tasks (one Type A and one Type B) and its duration was one hour. Similarly, the posttest contained two parallel items (one Type A and one Type B) with the same duration. Table 3.4 provides a summary of the types of investigative tasks for the pretest and the posttest. Since the students in the present study will not understand the terms ‘Happy Numbers’ in Pretest Task 1 and ‘Kaprekar Sequences’ in Posttest Task 1, the tests will use a different heading for the two tasks, namely, ‘Square Each Digit and Add’ for Pretest Task 1, and ‘Add Sum of Digits to Number’ for Posttest Task 1 (see Appendix D which shows the task statements). But for the discussion in this thesis, these headings are too long and they do not convey immediately what the two tasks are about, so the terms ‘Happy Numbers’ and ‘Kaprekar Sequences’ will be retained in the thesis.

Table 3.4 Types of Investigative Tasks for Pretest and Posttest

Task	Name of Task used in this Thesis	Type of Task
Pretest Task 1	Happy Numbers (or Happy)	Type A
Pretest Task 2	Toast	Type B
Posttest Task 1	Kaprekar Sequences (or Kaprekar)	Type A
Posttest Task 2	Sausage	Type B

The posttest must contain parallel items. Unlike procedural tasks where parallel items just mean changing some numbers while retaining the same method of solution, parallel items for investigative tasks could not be designed just by changing the numbers, or else the tasks in the posttest would just be an exercise for the students with nothing new to discover. Moreover, the students were expected to change the given numbers in Type B tasks in order to generalise. Therefore, there was a need to find tasks with some similarities in their structures to make them parallel to each other for the pretest and the posttest. Appendix E shows the detailed task analyses of the four tasks for the two tests, while Table 3.5 summarises the similarities between the two Type A tasks and the two Type B tasks for the pretest and the posttest.

Table 3.5 Similarities Between the Pretest and Posttest Tasks

Aspects	Pretest Task 1 and Posttest Task 1	Pretest Task 2 and Posttest Task 2
Topic	Both tasks involve Arithmetic: Numbers and Sequences	Both tasks involves Arithmetic: Numbers involving a context
Type of Task	Both tasks are Type A: Pose general problem (search for any pattern), specialise, no need to extend	Both tasks are Type B: Pose specific problems, use other heuristics, need to extend in order to generalise
Task Statement	Both tasks involve digits, addition and repetition of process to obtain new number	Both tasks involve a context about food and two variables: toast 3 slices of bread in a grill that can hold exactly 2 slices for Pretest Task 2; and share 12 sausages equally among 18 people for Posttest Task 2
Structure of Solutions	Both tasks involve two types of sequences each, with no known formula for the general term	Both tasks involve three methods of toasting the 3 slices of bread or cutting the 12 sausages: Method A (Usual Method), Method B (Shortest Method), Method C (Long Method)
Understanding the Task	Both tasks involve the need to understand the meaning of ‘new number’ and ‘repeat this process’ in the task statement (common misconceptions), and the trying of examples to understand the task	Both tasks involve the need to make sense of the context to understand the task, and there is no need to try examples to understand the task
Problem Posing	Both tasks involve posing the general problem to search for any pattern	Both tasks involve the need to pose specific problems to solve, but one key question central to both tasks is to minimise: the time for Pretest Task 2 and the number of cuts for Posttest Task 2
Specialising or Using Other Heuristics	Both tasks involve specialising to look for patterns, rather than using other heuristics	Both tasks involve using other heuristics such as reasoning, rather than specialising to look for patterns
Conjecturing	Both tasks involve conjecturing based on observing patterns from specialising	Both tasks involve conjecturing based on using other heuristics, such as reasoning, for the original task; but specialising to formulate conjectures when generalising for the extension
Justifying and Generalising	Both tasks involve justifying conjectures using non-proof arguments or formal proofs involving algebra, which will lead to generalisation	Both tasks involve justifying conjectures using non-proof arguments or formal proofs involving algebra, which will not lead to generalisation for the original task, but generalisation for the extension
Extension	Not expected to extend both tasks within the duration of the test as there are many patterns to find in the original task, and extension would usually result in a new task with completely different patterns	Expected to extend both tasks to find a general formula involving the two variables (toast m slices of bread in a grill that can hold exactly n slices for Pretest Task 2; and share m sausages equally among n people for Posttest Task 2)

3.6.5 Data Collection

Kelly and Lesh (2000a) emphasised the importance of deciding on the types of data to be collected for a research before deciding on the methodology since the methodology chosen should be suitable to capture all the necessary information for the research. Based on the research questions formulated for the study, the types of data to be collected for the present study are the cognitive and metacognitive processes of students while they were doing mathematical investigation, and the proficiency of their performance in the investigation. The latter could easily be gathered by giving the students a test and then looking at the solutions in their answer scripts if the performance depends only on the outcomes. But since the performance includes both outcomes and processes, the main problem would be how to capture the students' processes and to evaluate their processes. From the discussion on the eight methods to collect data on thinking processes in Sections 2.4.1 and 2.4.2, it was found that the thinking-aloud method comes closest to a reflection of the actual thinking processes engaged by students in a mathematical activity (Ericsson & Simon, 1980; 1993; Lesh et al., 2000), and thus is the most suitable method to capture the students' actual thought processes during investigation, despite certain shortfalls. Therefore, the main instruments for collecting the data for the present study were the pretest and the posttest, and the method was thinking aloud.

To alleviate some of the shortfalls of thinking aloud, there was a need to provide the students the opportunity to practise thinking aloud so that they could articulate their thoughts more accurately while at the same time minimising the interference of thinking aloud with their train of thought to attempt the task at hand. Thus the students practised thinking aloud during the first lesson in the present study before the pretest and during the last two lessons before the posttest. The instructions (see Appendix F), which were read to the students before they started practising thinking aloud during the lessons, were adapted from Foong (1990) with some modifications. Some important points to emphasise to the students in the instructions had been discussed in detail in Section 2.4.2.

The 10 students were separately videotaped thinking aloud using a videocam for the 60-minute pretest during the week after their familiarisation lesson, and for the 60-minute posttest during the week after the last developing lesson. Since the students had lessons in the morning until early afternoon, and they were not free for most afternoons because of other school activities, there was a need to videotape a few students at a time on two separate afternoons. Each student was allocated a classroom and there was a need for an invigilator for each student. Thus I engaged the help of other people to invigilate the students during the test. The invigilators were briefed beforehand by me using the instruction sheet for invigilators (see Appendix G). For example, the invigilators were told what to do if the students stopped thinking aloud. The instruction sheet also contained exact instructions to be read to the students before the test, including administrative instructions for the test and instructions for thinking aloud. However, I would set up and start the video recorder for each student. This means that I had to go around to each classroom to do so.

In addition, the students were told beforehand during the lessons, and also just before the test, to show all their working on their answer scripts, including any false trail, instead of just presenting the final solutions. The video recorder captured each student's answer script as he or she was writing and thinking aloud during the test.

For the posttest, the students could also refer to a checklist which contained a summary of the investigation processes that they had learnt during the teaching experiment (see Appendix H). This checklist was not given to the students during the pretest because they might be confused by certain processes which had not been taught during the familiarisation lesson.

3.7 PILOT STUDY

The pilot study was different from the initial exploratory study described in Section 3.3. Since the exploratory study used a different test instrument for a different purpose as the present study, there was a need to pilot the new test instrument and materials for the main study. The purposes of the pilot study were to:

- trial the materials for the familiarisation lesson and its implementation;
- trial the test items for the pretest;
- trial the students' thinking aloud during the pretest and the equipment for videotaping the students;
- collect the students' actual thinking-aloud protocols for the pretest in order to refine the initial coding scheme constructed for coding such protocols.

The sample for the pilot study was a convenience sample of 10 Secondary 1 students from the same school as the main study because the school was not able to provide Secondary 2 students for me to choose from. Since the pilot study was conducted during the last term of the year, the Secondary 1 students had already learnt number pattern and enough algebra by that time. Moreover, the purpose of the pilot study was not to study the nature of their investigation processes but to trial the test instruments for the main study. Thus a convenience sample of Secondary 1 students was deemed to be adequate for the pilot study.

The 10 students in the sample went through the two-hour familiarisation lesson and sat for the pretest. The school was unable to give me more time to test the materials for the developing lessons and the posttest. Only five of the 10 students were videotaped thinking aloud for the pretest due to the time constraint. The materials for the familiarisation lesson and the pretest were found to be satisfactory, and so there was no need to make any change. The practice of thinking aloud during the second half of the familiarisation lesson was also found to be just enough for most students to think aloud during the pretest, so there was no need to increase the practice time in the main study. The verbal protocols collected were then transcribed and coded to refine the coding scheme, which will be discussed later in Chapter 4.

3.8 MAIN STUDY

The main study was conducted for the 10 Secondary Two students whose profiles were described in Section 3.6.2(a). I taught the two-hour familiarisation and the five two-hour developing lessons according to the outlines and instructional strategies highlighted in Section 3.6.3. The tasks used in the lessons and for the pretest and the posttest were as described in Section 3.6.4, and each student was videotaped writing in their answer scripts as they thought aloud during the tests as described in Section 3.6.5.

3.9 QUALITY OF PRESENT STUDY

Unlike traditional quantitative research in education which emphasises reproducibility and accuracy, in a qualitative study where the goal is to produce a description of a complex system, such as a model of the students' processes in investigation, "truth and falsity may not be at issue as much as fidelity, internal consistency, and other characteristics that are similar to those that apply to quality assessments for photographs, portraits or verbal descriptions" (Lesh et al., 2000, p. 20). Thus the issue of the reliability of the coding scheme will have to be addressed by an inter-coder reliability test where a few experienced mathematics educators will code some samples of videotaped transcripts based on the coding scheme. If the reliability is low, the coding scheme will have to be refined and subjected to another round of reliability testing. If the reliability is high, it will suggest that the coding scheme is reliable. The next chapter will show that the coding scheme designed for the present study has passed the inter-coder reliability test (see Section 4.7).

Silverman (2005) also proposed some other criteria for evaluating the quality of a qualitative study. One of the criteria is whether the study is able to build useful theories or models. Schoenfeld (2002) also discussed the "*descriptive power* … of theories or models to capture 'what counts' in ways that seem faithful to the phenomena being described" (p. 456). He explained that this means if another expert, who is familiar with the coding scheme and the model, views a coded transcript of a videotaped session of a student's behaviour, then the expert will not be surprised by any unexpected behaviour of the student when the expert views the actual videotape. Since the coding scheme has passed inter-coder reliability test, and the coding of the students' thinking-aloud protocols is used to validate and refine the investigation models developed for the present study, then the models will be reliable, and the descriptive power of both the coding scheme and the models will also be high.

Another criterion to evaluate the quality of a qualitative study, as proposed by Silverman (2005), is "to what extent do our preferred research methods reflect careful weighing of the alternatives, or simple responses to time and resource constraints or even an unthinking adoption of the current fashions?" (p. 229) He cited the example of many researchers on health sciences blindly following the trend of using interviews in their studies, but he believed that interviews alone were not sufficient because interviewees might not tell the true stories. For my type of research study, the three common research methodologies are:

(i) conduct a paper-and-pencil study for a big sample to cover the breadth, and an in-depth interview for a small sample;
(ii) conduct a test using the thinking-aloud method for a small sample;
(iii) conduct a teaching experiment for a small sample, with general descriptions of how investigation benefits the students.

I did not just adopt the current fashions without giving careful thoughts to alternatives. Instead, I conducted an initial exploratory study to find out whether a paper-and-pencil study was feasible for this kind of open investigative tasks (see Section 3.3). The findings of this exploratory study indicated that most high-achieving students did not know what to investigate when given this type of investigative tasks. In other words, there was a need for some teaching to be done first. This means that the first two options above were not feasible. The last option was also not appropriate because the research methodology would not be able to capture the students' actual thinking processes. In the end, I had to combine the last two options: a teaching experiment for a small sample to develop the processes, coupled with a pretest and a posttest using the thinking-aloud method to capture the actual thinking processes (see Section 3.6). These were some measures taken to ensure the quality of the present study.

3.10 CONCLUDING REMARKS FOR THIS CHAPTER

Chapter 3 has provided an outline of the research methodology for the present study. A group of 10 Secondary 2 students from a high-performing school had undergone a teaching experiment to develop their cognitive and metacognitive processes in mathematical investigation. In addition, the students were videotaped thinking aloud during the pretest and the posttest in order to capture their thinking processes so as to inform the two theoretical investigation models that describe the interactions of these processes. Chapter 4 will then describe the construction of a coding scheme for coding the thinking-aloud protocols of the students during the pretest and posttest.

CHAPTER 4: DEVELOPMENT OF CODING SCHEME

This chapter will describe the development of a coding scheme for coding students' cognitive and metacognitive thinking-aloud behaviours during the pretest and posttest of the present study. It will begin with preparing the data obtained from the study by transcribing the students' thinking-aloud protocols and other actions captured by the video recorder during the tests. Then the chapter will explain the purpose of the coding scheme, followed by a description of the five phases in the construction of the coding scheme, including an inter-coder reliability test for the Final Coding Scheme. The parsing of protocols into stages and episodes will also be discussed.

4.1 TRANSCRIBING OF THINKING-ALOUD PROTOCOLS

This section will describe the preparation of thinking-aloud data before coding could take place. The two main sources of data collected for the present study were:

- the students' pretest and posttest answer scripts;
- the students' thinking-aloud or verbal protocols, and other non-verbal actions, recorded on videotape during the pretest and posttest.

To analyse the data, the thinking-aloud protocols needed to be transcribed. But the non-verbal actions performed by the students could also be important to fill in the gaps not captured by their verbal protocols or their test answer scripts. For example, a student just said the following:

> "How to use this and get that?" [S1; Pretest 1]

This verbal protocol was meaningless unless it was known what the student meant by 'this' and 'that'. From the videotape, the student was seen pointing at the number 30 and then the number 500 in his answer script. Thus the verbal protocol was transcribed as follows, with non-verbal actions recorded in square brackets:

> "How to use this [point to the number 30] and get that [point to the number 500]?" [S1; Pretest 1]

Sometimes, the students were silent for a few seconds but they pointed their pen at different parts of their answer script as they were thinking. Then they observed a pattern. If the students' actions were not recorded in the transcript, anyone reading the transcript might think that the students discovered something while thinking silently, when in fact they were actually looking at certain parts of their working for a pattern. Thus it was necessary to record in the transcript all the students' actions in order to capture their processes fully. Some students were also able to multi-task: they wrote one thing but thought aloud a different thing. For example, the following student wrote down the numbers 21 to 30, and while he was writing down the sum of the digits for all the numbers 21 to 30, e.g. he wrote + 2 + 1 for the number 21, he was able to multi-task and think of something else:

> "Ok … let us try now starting from [start writing: 21, 22, 23 until 30] 21, 21, 25, 26, 27 [stop writing]. Well, while writing, I am thinking [start writing: + 2 + 1 for the number 21; and similarly for the other numbers until 30] whether I am going to the right place … because it seems that, hmm, this thing is pretty hard to prove and … actually, in fact, I don't really know what am I doing [stop writing]." [S9; Posttest 1]

If only his verbal protocols were transcribed, it would not be evident that he was actually adding to each number the sum of its digits. As a result, it was necessary to view the videotape to see what the students were doing and then describe their actions in writing in addition to transcribing their thinking-aloud protocols.

Since most students did not say aloud what they were writing all the time, it was also necessary to record what they were writing in the transcript, or else anyone reading the transcript might not understand what the students were doing because of the gaps in their thinking-aloud protocols which did not include what they were writing. Even if a person reads the transcript with the corresponding answer script, it is still very tedious to fill in the gaps in the verbal protocols because it is not easy to match a certain part of the answer script to the corresponding part of the protocols. Therefore, it was necessary to include in the transcript what the students wrote but did not say aloud. However, the students' handwriting was occasionally too small to be read on the videotape, or sometimes the students had unintentionally blocked their answer scripts from being captured fully by the video recorder. Thus the actual answer script had to be used during transcribing: from the position of what the student was writing in the videotape and sometimes also from their verbal thinking-aloud protocols, it was possible to identify from the actual answer script what the student was writing. This was then recorded in the transcript to ensure that everything was captured properly.

Furthermore, for ease of reference when describing in the transcripts what the students were doing during the tests, there was a need to label each page, specific problem posed, example tried, pattern observed, and conjecture formulated, with a number, in order to keep track of the students' working since most students did not label them. For example, when a student turned to a previous page or pointed at a previous example as seen in the videotape, there was a need to record which page and which example the student was pointing at respectively, and if all the pages and examples were labelled with a number, it would be easier to refer to.

Hence, the transcripts contained not only the students' verbal protocols but descriptions of their actions as well, including what they had written in their answer scripts without saying them aloud. In this way, the two main sources of data complemented one another to ensure that all the essential data or processes were captured and recorded in the transcripts. In other words, anyone reading the transcripts should not be surprised by any unexpected behaviour of the student when he or she views the actual videotape (Schoenfeld, 2002). Therefore, the transcripts had replaced the videotapes, and the latter were no longer necessary for data analysis. But the students' answer scripts were still helpful because they showed the students' working and solutions in their entirety, unlike the transcripts. Although the transcripts included what the students wrote in their answer scripts, the transcripts contained the students' thinking-aloud protocols interspersed with parts of their working here and there, so the details of the students' working could be 'lost' among the myriad of their protocols (e.g. see Line 20 in the sample transcript in Appendix I). Thus the actual answer scripts were still helpful in giving an overall picture of their solutions (see the corresponding working in the first column in the answer script in Appendix J). Hence, the end products at this stage were the answer scripts and the transcripts.

In general, the transcripts were used to inform the nature of the processes engaged in by the students during mathematical investigation, and the answer scripts to inform the outcomes of the investigation, e.g. the student had discovered a pattern. But even though the students had been told to show all their working, sometimes they would just say that they had discovered a particular pattern, and then they thought about it, decided that it was incorrect and so did not write it down. Therefore, the outcomes of an investigation were gathered from multiple sources as well: their answer scripts and their verbal protocols if necessary. After the preparation of the data, the next step was to code the protocols in the transcripts before the data could be analysed to inform the investigation models developed for the current study.

4.2 PURPOSE OF CODING SCHEME

The theoretical basis for protocol analysis used to analyse the cognitive and metacognitive processes in the present study has been discussed in detail during the literature review in Section 2.4.3. To analyse the students' thinking-aloud protocols during mathematical investigation, there was a need to develop a valid and reliable coding scheme to code the transcripts of their protocols. The theoretical investigation models described in Section 3.2 had posited that students would go through a logical sequence of cognitive processes in eight stages, and their metacognitive processes would interact with their cognitive processes in complicated manners. As such, a coding scheme would provide a means to identify these processes. The coded data would then aid in the interpretation of how these processes interact with one another so as to inform the theoretical investigation models.

The development of the coding scheme comprised five phases, which follow a combination of a top-down (theoretical driven) and a bottom-up (empirical driven) strategy described in the literature review in Section 2.4.3. The first phase was the construction of the Initial Coding Scheme based on current literature. The second phase was the application of the Initial Coding Scheme to four transcripts of thinking-aloud protocols from the pretest of the pilot study to obtain a Revised Coding Scheme. The third phase was the application of the Revised Coding Scheme to another four transcripts of protocols from the pretest of the pilot study to fine-tune the coding scheme. Since it was found that there was no further change to the coding scheme, the product at the end of the third phase was still called the Revised Coding Scheme so as not to use too many names. As investigation was not easy for students who had no prior experience, it was unlikely that they would exhibit the full range of processes during the pretest of the pilot study. Therefore, the fourth phase was the use of four transcripts of posttest protocols from the main study to further refine the coding scheme. The last phase was the inter-coder reliability test, and the product at the end of the last phase was the Final Coding Scheme.

4.3 PHASE 1: DEVELOPMENT OF INITIAL CODING SCHEME BASED ON THEORY

The review of relevant literature in Chapter 2 had revealed that there was practically no empirical research in mathematical investigation where students used the thinking-aloud method. However, there were quite a number of empirical studies on problem solving using thinking aloud, such as those cited in Sections 2.2.4 and 2.3.4. Since many processes in problem solving were similar to those in investigation, I started with the coding schemes described in some of these empirical studies and then modified them for investigation based on the theoretical research on the processes of investigation described in Section 2.2.3. Thus the Initial Coding Scheme was formulated based on two sources of current literature:

- the coding schemes designed in empirical studies on problem solving,

- the processes identified during theoretical research on investigation.

Using the guidelines created by van Someren et al. (1994) and Meijer et al. (2006) described in Section 2.4.3, possible students' behaviours that could be observed during thinking aloud were identified and matched to the processes and outcomes described in the two investigation models formulated for the present study in Section 3.2. The classification of whether a process was cognitive or metacognitive followed from the literature review on metacognitive and cognitive processes in Section 2.3.2.

Table 4.1 shows the Initial Coding Scheme. It was divided into two categories. The first category, Category C, shows the codes for cognitive behaviours used to inform the investigation model for cognitive processes. A student's thinking-aloud behaviour could be a process or an outcome. Non-italic codes indicated processes while codes in italics indicated outcomes. Since the model divided the processes and outcomes into eight stages, the codes for the cognitive behaviours were also divided into the eight stages. But there were four behaviours that could occur in any stage, so they were classified under 'Others' in Category C. The second category, Category M, shows the codes for metacognitive behaviours used to inform the investigation model for metacognitive processes. Since most of these processes could occur in more than one stage, they were not divided into the stages. There were a total of 36 behaviour codes: 31 cognitive codes and 5 metacognitive codes. Most codes were self-explanatory; if otherwise, short explanations for the codes were given in the coding scheme. More detailed descriptions of the behaviours represented by these codes could be found in the literature review of the processes and outcomes of investigation described in Section 2.2.3. All Category M codes began with the letter M for metacognitive behaviours. The first letter of a Category C code was the abbreviation for the investigation stage, e.g. U1 to U6 were cognitive behaviours for the stage of Understanding the Task.

Table 4.1 Initial Coding Scheme

Legend: Non-italic codes represent processes; codes in italics represent outcomes.

Category C: Cognitive Behaviours

Stage 1: Understanding the Task (U)
First Reading of Task (U1)
Re-reading Task (U2)
Rephrasing Task (U3)
Highlighting Key Information (U4)
Visualising Information (U5)
Understanding Task by Trying Examples (U6)

Stage 2: Problem Posing (P)
Thinking of Problem to Pose (P1)
Posed Problem (P2): Posed problem without changing the given in the original task

Stage 3: Specialising and Using Other Heuristics (S/H)
Specialising (S1): Trying examples to look for pattern
Thinking of Plan to Solve Problem (H1)
Decided on Plan (H2)
Using Reasoning (H3)
Using Algebra (H4)

Stage 4: Conjecturing (C)
Searching for Pattern (C1)
Formulated Conjecture (C2)

Stage 5: Justifying (J)
Thinking of Plan to Justify (J1)
Naïve Testing (J2): Trying examples to refute conjecture if possible
Justifying Conjecture using Non-proof Argument (J3)
Justifying Conjecture using Formal Proof (J4)
Verified Conjecture Correct (J5)
Justified or Proven Conjecture (J6)
Refuted Conjecture (J7)

Stage 6: Generalising (G)
Solved Problem that led to Generalisation (G1)
Solved Problem without Generalising (G2)

Stage 7: Checking (R)
Checking Correctness of Working (R1)

Stage 8: Extension (E)
Thinking of How to Extend (E1)
Posed Problem to Extend (E2): Posed problem by changing the given in the original task

Others
Performing Calculation (X1)
Referring to Given Checklist (X2)
Made Mistake (X3)
Discovered Mistake (X4)

Category M: Metacognitive Behaviours

Monitoring Understanding (M1): By clarifying task requirements, given conditions or meaning of some parts of task
Analysing Feasibility of Goal (M2): Analysing whether a goal or a problem (including extension) was feasible or worth pursuing
Analysing Feasibility of Plan (M3): Analysing whether a plan to solve a problem or to justify a conjecture was feasible or worth pursuing
Monitoring Progress (M4)
Reviewing Solution (M5): Reviewing solution to see if it had achieved the goal or solved the problem, including evaluating the efficacy of a method of solution

4.4 PHASE 2: DEVELOPMENT OF REVISED CODING SCHEME BASED ON EMPIRICAL DATA

In Phase 2, the Initial Coding Scheme developed theoretically in Phase 1 was applied to four transcripts of actual thinking-aloud protocols from the pretest of the pilot study to obtain a Revised Coding Scheme. Two transcripts were for Pretest Task 1 (Type A task) while the other two transcripts were for Pretest Task 2 (Type B task).

4.4.1 Coding of Transcripts of Thinking-Aloud Protocols

There was a need to divide the protocols in the transcripts into lines so that each line could be coded. If the protocols were derived from students' discussions as they worked in groups, it would be very natural to divide the protocols into lines according to different students' protocols and then code each line, although it could still be a problem if one student talked continuously for some time, which might result in a few codes for his or her portion of protocols. However, the latter was a major problem for students thinking aloud individually in the present study because there was only one student talking continuously throughout the duration. Therefore, there was a need to find another way to divide the protocols into lines.

Table 4.2 shows a sample transcript with a column for the behaviour code (or 'Bhvr Code' in short). The remarks column was for recording additional points such as the number for specific problem posed (see Line 06) and mistake made (see Line 07) for easy reference as explained earlier in Section 4.1. According to Ericsson and Simon (1993), one way to separate the protocols into lines was to try to assign the codes to the protocols and dividing the protocols into lines at the same time. As I read the protocols, I assigned a code to the protocols until I came to a point where there was a need for a different code. This was then the point to separate the previous protocol from the rest of the protocols. The previous protocol would then be considered a protocol line and it was coded accordingly. The process was then repeated until the end of the transcript.

Table 4.2 Sample Transcript Coded using Initial Coding Scheme

Note: The time for this Pretest Task 2 started at 33:06 because it was continuous videotaping after Pretest Task 1.

Line	Time	Protocols	Bhvr Code	Remarks
01	33:06	Three slices of bread are to be toasted under a grill. The grill can hold exactly two slices. Only one side of each slice is toasted at a time. It takes 30 seconds to toast one side of a slice of bread.	U1	
02	33:30	It takes 30 seconds to toast one side of a slice of bread.	U2	
03	33:36	5 seconds to put a slice in or to take a slice out, and 3 seconds to turn a slice over. Investigate. Okay.	U1	Continue first reading
04	33:50	[Starts drawing bread shape] First bread, first bread. Bread, draw the bread out. [Stop drawing]. [Write in bread drawing: ①] First bread. Go into the grill [draw an arrow from the bread and write after the arrow: grill]. Take 3 seconds [write beside arrow: take 3 seconds]	U5	
05	34:12	First, what must I investigate about?	P1	
06	34:14	[Write: 1st] I think I'll find [start writing] the total time taken, the total time taken to [stop writing]. To investigate [cancel: 1st] [write: to investigate]. [Continue writing] the total time taken to, to toast the three slices of bread, slices of bread [stop writing]. Finish.	P2	Posed Specific Problem 1
07	34:59	So first bread go into the grill [point pencil at bread ① and then at the word 'grill'] take 3 seconds …	X3	Mistake 1: Should be 5 seconds
08	35:06	Then at the same time [draw second bread] at the same time [write in bread drawing: ②] second bread also go into the grill [draw an arrow from the bread and write after the arrow: grill] same time [draw curly bracket for both slices and write: same time]. Take another 3 seconds [write beside arrow: take 3 seconds].	U5	
09	35:29	So 6 seconds already, to put 2 slices of bread [point to bread ①, then bread ②] under a grill [point to the words 'same time'].	H3	
10	35:33	[Pause for 4 seconds]	?	No suitable code
11	35:37	Then the grill [point to task statement] it takes 30 seconds to toast one side of a slice of bread …	U2	
12	35:47	Okay, so 30 seconds taken to toast [write: 30 seconds to toast] these two sides: this side [shade bread ②] and this side [shade bread ①]. 30 seconds [box up '30 seconds'] to take, to toast these two sides.	H3	

In general, each protocol line should be assigned only one code. Sometimes, it was possible to observe two behaviours in a protocol line. For example, in Line 08 in Table 4.2, the student was visualising the information (U5) and using reasoning (H3) at the same time, so it was not possible to split this protocol line further. However, frequently assigning more than one code to a protocol line would make the transcript look cluttered. Therefore, for simplicity, only the dominant behaviour in a protocol line would be coded. In Line 08, the dominant behaviour was visualising the information, compared with Line 09 when the student was just using reasoning (H3) without visualising the information. Thus Line 08 was coded as U5 only. Sometimes, it was also found that there were no suitable codes in the coding scheme to code some behaviours, such as pausing for 4 seconds in Line 10.

Occasionally, as I read further, I would discover that one of the previous protocols was coded wrongly because I had misinterpreted what the student was doing until I read the subsequent protocols. Therefore, there was a need to read a bigger chunk of the protocols in order to get a sense of what the student was trying to do, before going back to code the previous protocols and divide them into lines at the same time (Silverman, 2005). After coding the entire transcript, I had a better overall picture of what the student was doing, and I would then check through the entire transcript from the beginning to the end to ensure that each protocol line was coded correctly. If it was found during the checking that two consecutive protocol lines ended up with the same code after an amendment, then the two lines were combined to become one line.

4.4.2 Some Issues with Initial Coding Scheme

The coding process was not easy. Quite often, it was difficult to decide which code to assign to a protocol line because either (i) there was more than one code that appeared to fit the behaviour, or (ii) none of the codes in the coding scheme were suitable enough to describe the behaviour in the protocol line. Thus there was a need to define some of the codes more precisely so that they were more clearly observable, and to include enough representative sample protocols for these codes to identify the behaviours easily. All the problems discovered so far with the Initial Coding Scheme in Phase 2 concerned the codes in Category C for cognitive behaviours. There was no problem with the metacognitive codes in Category M, probably because the four transcripts did not contain enough metacognitive behaviours as all the protocols for the students in the pilot test were for the pretest only. Thus issues with metacognitive behaviours will be discussed later on in Phase 4. This section will focus on the problems with the codes in Category C for the Initial Coding Scheme.

(a) Problematic Codes

From the four transcripts, it was found that there were some behaviours which could happen in more than one investigation stage, or could not be coded reliably.

Re-reading Task (U2) and Rephrasing Task (U3)

In the Initial Coding Scheme, based on the theory derived from the literature review in Chapter 2, students were supposed to re-read or rephrase (or paraphrase) the task or parts of the task to understand the task during the first stage of investigation. But it was found from the students' protocols that students re-read or rephrase the task very often in subsequent stages when they were stuck, and they did it for other reasons, such as to find a problem to solve or extend, to think of a plan to solve a problem, or to monitor progress. For example, in the sample transcript discussed earlier in Section 4.4.1, the student re-read the task (U2) in Line 02 to understand the task, but she did it again in Line 11 in the third stage of using other heuristics to help her think of how to solve the problem posed in Line 06. Thus these two behaviours of re-reading the task and rephrasing the task could occur in more than one stage, and so they should not be classified under the first stage of investigation only.

Visualising Information (U5)

Similarly, it was found from the students' protocols and answer scripts that they visualised the given information by drawing a diagram, not only in the first stage of understanding the task but also in subsequent stages. For example, in the same sample transcript discussed earlier in Section 4.4.1, the student visualised the information (U5) by drawing the slices of bread in Line 04 to understand the task, but she did it again in Line 08 in the third stage of using other heuristics to help her think of how to solve the problem posed in Line 06. It was also possible for other students to visualise the given information to help them justify a conjecture in the fifth stage of justifying. Therefore, visualising information was another behaviour that could appear in more than one stage, so it should not be classified under the first stage only.

Thinking of Problem to Pose (P1) and Thinking of How to Extend (E1)

It was found from the students' protocols that it was not easy to distinguish between posing problems without changing the given (P1) and posing problems to extend by changing the given (E1). After the students had solved a problem and checked the solution, they could choose to pose another problem to solve under the umbrella of the original task without changing the given, or pose another problem to extend by changing the given. Very often, the students did not know what to pose, so there was a period when they were just thinking of what problem to pose. In the end, they could end up posing a problem with or without changing the given. Sometimes, they did not even pose any problem after thinking of what problem to pose, so it was not possible to decide whether to code this period of thinking P1 or E1. Therefore, these two behaviours were combined to 'Thinking of Problem to Pose or Extend', and it could happen in the second stage of problem posing or the last stage of extension.

Trying Example to Understand Task (U6), Specialise (S1) or for Naïve Testing (J2)

When students were trying an example, it was not easy to decide from their verbal protocols whether they were trying an example to understand the task (U6), trying an example to specialise (S1), or trying an example to see if a conjecture could be refuted by a counter example (J2), because they usually did not verbalise their intention. Thus it was decided to code the behaviour simply as 'Trying Example'. In other words, the codes U6, S1 and J2 would have to be removed from the Initial Coding Scheme. It would be left to the parsing of protocols into the various stages of investigation to interpret what the students were trying the example for (see Section 4.4.4 later).

(b) New Codes

From the four transcripts, it was found that some behaviours could not be coded using the Initial Coding Scheme, so there was a need to create a few new codes.

Observed Pattern and Rejected Observed Pattern

As posited by the investigation model for cognitive processes developed for the present study, when students observed a pattern, the pattern was only a conjecture to be proven or refuted. However, it was found from the transcripts that there was usually a time gap between observing a pattern and considering the pattern as a conjecture. This was because when the students first spotted a pattern, they were usually unsure that this was the pattern. So the students were observed trying more examples to be more certain of the pattern first, before they accepted it as a conjecture and then tried to prove it. Sometimes, they found counter examples to reject the observed pattern even before formulating it as a conjecture. Thus there was a need to distinguish between observing a pattern and formulating a conjecture, and so two new codes 'Observed Pattern' and 'Rejected Observed Pattern' were created. As this was a new significant finding for the present study, it will be examined in more detail using actual protocols during the data analysis in Section 7.2.4.

Pausing and Hesitating

Another problem was that the students kept silent during the pretest very often despite being told to think aloud. This was to be expected since all of us needed time to think of something. If the students paused for 3 seconds or less, the pause would be represented by three dots '...' in the transcripts since it was unlikely that the students would think of something significant during such a short pause. But if the students paused for more than 3 seconds, they had been reminded by the invigilator to think aloud (see Instructions for Invigilator in Appendix G). However, despite the reminder, some students continued to pause for more than 3 seconds occasionally. If such pauses were ignored and the student ended up thinking of something, it would appear that the student thought of the something immediately when in fact he or she took a long pause to think about it. Thus pauses longer than three seconds would be recorded in the transcripts with the duration separately, e.g. [Pause 5 seconds], and a new code was needed to code this. The students also hesitated a lot during the pretest. This was different from behaviours such as 'Thinking of Problem to Pose' (P1) or 'Thinking of Plan to Solve Problem' (H1). For example, consider the following two protocols:

> "So how should I solve the problem? I think I will try a new example."
>
> "What should I do is ... what I should do is ... um ... I don't know what to do."

For the first protocol, the student was trying to think of a plan to solve the problem and his plan was to try a new example. For the second protocol, the student was stuck and did nothing constructive to solve the problem. Thus there was a need to code this frequent behaviour of the students hesitating because they were stuck.

Other Miscellaneous Behaviours

It was found from the protocols that the students engaged in other miscellaneous behaviours very often, e.g. they re-read what they had written, such as their working or problem posed, quite frequently. This was coded as 'Re-reading What was Written' since it was not possible to read what was in their minds. Some of the students also rewrote part of the solution from the previous page onto a new page for ease of reference, while others took a long time to rewrite the full solution properly. At other times, some students went back to organise their examples by labelling them or boxing them up separately. Thus there was a need to code such recording and organising processes that were not part of the thinking processes (Frobisher, 1994). There were also other off-task behaviours, such as when the invigilator told the students to speak louder or to continue to think aloud when they were silent for more than 3 seconds. Although it was beyond the scope of the present study to investigate the effect of affective variables on mathematical investigation, there was still a need to code such behaviours when they occurred during the tests, e.g. some students sighed or said things such as “Why is this problem so difficult?” or “I give up!”

Unable to Code

Despite the addition of new codes to the Initial Coding Scheme, there were still thinking-aloud protocols which were not possible to code because it was unclear what the students were thinking about. For example, consider the following excerpt:

> “And then … um, let’s see [flip to p. 2] … um … 37, 21.” [S10, Pretest 1]

It was not clear whether the student was ‘hesitating’ or ‘searching for patterns’, even by examining the surrounding protocols. Thus this kind of protocols was classified as ‘Unable to Code’. Other instances include students mumbling, which was transcribed as [unintelligible words] and classified as ‘Unable to Code’.

(c) Labelling of Codes

The use of numbers, e.g. C1-C2, to label codes in the Initial Coding Scheme had caused quite a number of problems. First, whenever a new code was invented, the numbering for existing codes might have to be changed. This was because it was sometimes more appropriate to insert the new code between two existing codes than to put the new code at the end of the existing codes as there was some kind of order among the codes. For example, the new code ‘Observed Pattern’, which was described earlier in this section, should come between Searching for Pattern (C1) and Formulated Conjecture (C2), so this would affect the numbers of subsequent codes. Similarly, removing a code could also change the numbers of the codes that follow. Secondly, it was difficult for anyone, even for someone familiar with the codes such as me, to associate a number with a particular code in order to remember what the code represented. Therefore, there was a need for a better system.

4.4.3 Revised Coding Scheme

Some Important Decisions Made

Based on the issues discussed in the previous section, some important decisions were made when modifying the Initial Coding Scheme to obtain the Revised Coding Scheme. First, a new system of labelling the codes was introduced: the use of parts of the acronyms of behaviours as labels for the codes. For example, ‘Re-read Task’ was denoted by RR and ‘Trying Example’ by TE. But all five metacognitive codes would still begin with the letter M so that it would be easier to associate these codes with metacognitive behaviours, e.g. MR stood for the metacognitive behaviour of Reviewing Solution. Of course, it would still be difficult for anyone not familiar with the codes to remember what the codes stood for, but after working with the codes for some time, anyone would remember what the codes represented more easily than if the codes were labelled using numbers without meaning.

This new system of labelling the codes was not without problem. Sometimes, two behaviours might end up with the same code. For example, both 'Thinking of Problem to Pose or Extend' and 'Thinking of Plan' initially had the same code TP, so one of them had to be changed. Since the letter P was used as the abbreviation for the Problem Posing stage, it was decided that the code for 'Thinking of Problem to Pose or Extend' would be changed to PT, where the first letter also signified problem posing. Another example would be 'Made Mistake', which logically should be coded as MM. But it would be confused with a metacognitive code which always began with the letter M as explained in the preceding paragraph. Since 'Made Mistake' was not a metacognitive behaviour, it was changed to 'Made Error or Mistake' so that it could be assigned the code EM, and not ME which could still be confused with a metacognitive code. To be consistent, the behaviour 'Discovered Mistake' was also changed to 'Discovered Error or Mistake' so that it could be assigned the code ED.

Secondly, the codes in Category C would not be grouped according to the stages since, as explained in Section 4.4.2(a) earlier, quite a number of cognitive behaviours could occur in more than one stage. But the codes were still listed in some kind of order. For example, behaviours for the first stage of understanding the task should come before behaviours for the next stage of problem posing. On the other hand, it was sometimes necessary to group similar codes together for easy reference. For example, Thinking of Problem to Pose or Extend (PT), Posed Problem (PP) without changing the given, and Posed Problem to Extend (EP), were grouped together because they were closely related although they occurred in the different stages of Problem Posing (P) and Extension (E).

Thirdly, if it was possible to assign every behaviour to a unique investigation stage, then the parsing of protocols into the various stages of investigation could easily be done just by looking at the behaviour codes. However, it was not possible to do so since quite a number of cognitive behaviours could appear in more than one stage. Thus there was a need for a system to parse the protocols into the various stages more reliably. The fourth important decision of including a stage code in addition to a behaviour code for each protocol line will be discussed in detail in the next section.

Fourthly, there was a need to create a third category, i.e. Category X for Other Codes, to include behaviours that could not be classified in the other two categories, such as Pausing (XP), Hesitating (XH), Re-reading What was Written (XR), Re-writing or Recording Processes (XW), Off-Task Behaviours (XO), Affective Behaviours (XA) and Unable to Code (XU) described in Section 4.4.2(b) earlier.

Fifthly, there was a need to define some of the codes more precisely so that they could be more easily identified. Thus each code was given a detailed description, and if it appeared similarly to another code, a clarification of the difference was also highlighted in the Revised Coding Scheme. Representative samples of protocols were chosen from the four transcripts and included in the Revised Coding Scheme to identify the behaviours more easily. But some behaviours were not observed from the four transcripts and so there were some codes in the Revised Coding Scheme without any exemplar of protocols. The codes in the Revised Coding Scheme were also not complete. Therefore, due to space constraint in this thesis, the Revised Coding Scheme shown in Table 4.3 did not include the detailed explanations and exemplar of protocols for each code (these would only be shown in the seven-page Final Coding

Scheme in Section 4.6 later). There were a total of 39 codes in the Revised Coding Scheme: 26 codes for cognitive behaviours, 5 codes for metacognitive behaviours (all began with the letter M for metacognitive behaviours), and 8 codes for other behaviours (all began with the letter X).

Table 4.3 Revised Coding Scheme

Legend: Non-italic codes represent processes; codes in italics represent outcomes.

Category C: Cognitive Behaviours

First Reading of Task (FR)	Using Algebra (AL)
Re-reading Task (RR)	Searching for Pattern (SP)
Rephrasing Task (RT)	*Observed Pattern (OP)*
Highlighting Key Information (HI)	*Rejected Observed Pattern (RP)*
Visualising Information (VI)	Formulated Conjecture (FC)
Thinking of Problem to Pose or Extend (PT)	*Verified Pattern or Conjecture Correct (VC)*
Posed Problem (PP): Posed problem without changing the given in the original task	*Justified or Proven Conjecture (JC)*
Posed Problem to Extend (EP): Posed problem by changing the given in the original task	*Refuted Conjecture (RC)*
Thinking of Plan (TP)	*Solved Problem that led to Generalisation (SG)*
Decided on Plan (DP)	*Solved Problem without Generalising, i.e. No Generalisation (NG)*
Trying Example (TE)	Checking Correctness of Working (CW)
Performing Calculation (PC)	*Made Error or Mistake (EM)*
Using Reasoning (RE)	*Discovered Error or Mistake (ED)*

Category M: Metacognitive Behaviours

Monitoring Understanding (MU): By clarifying task requirements, given conditions or meaning of some parts of task

Analysing Feasibility of Goal (MG): Analysing whether a goal or a problem (including extension) was feasible or worth pursuing

Analysing Feasibility of Plan (MF): Analysing whether a plan to solve a problem or to justify a conjecture was feasible or worth pursuing

Monitoring Progress (MP)

Reviewing Solution (MR): Reviewing solution to see if it had achieved the goal or solved the problem, including evaluating the efficacy of a method of solution

Category X: Other Behaviours

Pausing (XP)	Re-writing or Recording Processes (XW)
Hesitating (XH)	Off-Task Behaviours (XO)
Referring to Given Checklist (XC)	Affective Behaviours (XA)
Re-reading What was Written (XR)	Unable to code (XU)

4.4.4 Parsing of Protocols into Stages and Episodes

As explained in the previous section, if it were possible to assign every cognitive behaviour to a unique investigation stage, then the parsing of protocols into the various stages of investigation could easily be done just by looking at the behaviour codes. But since it was not possible to do so because quite a number of cognitive behaviours could appear in more than one stage, there was a need for a system to parse the protocols into the various stages more reliably. The purpose of parsing was to help in the study of the interactions of the various processes within and across different stages so as to inform and refine the investigation models developed for this study. This section will discuss how parsing was done while the inter-coder reliability test for the parsing will be done later in Section 4.7.

(a) Stage Codes

Table 4.4 shows a sample transcript with a new column for 'Stage Code' in addition to the column for 'Behaviour Code' (or 'Bhvr Code' in short). To help to parse the protocols into the different investigation stages, a summary of possible behaviour codes for each stage was first prepared (again, due to space constraint, only the final one would be shown in Table 4.7 later). I would look at the behaviour code for the first protocol line and refer to this table of summary in order to narrow down to the few stages that the behaviour code could appear in. I would then read the protocol in that line to determine whether the behaviour code should occur in which investigation stage before assigning the corresponding stage code, e.g. U for Stage 1: Understanding the Task, which was the same as the abbreviation for each stage described in the investigation models developed for the present study. Since Stage 3

(S/H) consisted of two main processes, namely, Specialising (S) and Using Other Heuristics (H), it was decided to code each of the processes separately so that it was clear from the stage code whether the student was specialising *or* using other heuristics. For example, it was evident from the transcript in Table 4.4 that the student was Using Other Heuristics (H) from Line 07 onwards, and not specialising.

However, when the student paused for four seconds (see Line 10), it was not possible to read what was inside her mind. Although subsequent protocol lines (Lines 11-12) might suggest that she was still in the stage of using other heuristics, it might be possible that she was thinking of something else during the pause, such as trying to understand the task further or even thinking of another problem to pose. The latter actually happened when another student paused for some time during the stage of using other heuristics and then posed another problem to solve. Thus it was decided to assign a new code X for the stage code if the corresponding behaviour code belonged to Category X (Other Behaviours)

In other words, there were 10 possible stage codes to choose from although there were only eight stages in the investigation model for cognitive processes, since Stage 3 (S/H) was split into the stage codes S and H, and there was the new stage code X. Metacognitive codes would usually follow the investigation stage that they occurred in. But if they happened during the transition between two stages and it was not easy to determine which of the two stages that they fell in, then they would be assigned the stage code X. Schoenfeld (1985) discussed two types of transitions between stages, one which was productive, and the other not productive (see Section 2.3.3a for more details).

Table 4.4 Sample Transcript showing Parsing of Protocols into Stages using Revised Coding Scheme

Note: The time for this Pretest Task 2 started at 33:06 because it was continuous videotaping after Pretest Task 1.

Line	Time	Protocols	Stage Code	Bhvr Code	Remarks
01	33:06	Three slices of bread are to be toasted under a grill. The grill can hold exactly two slices. Only one side of each slice is toasted at a time. It takes 30 seconds to toast one side of a slice of bread.	U	FR	
02	33:30	It takes 30 seconds to toast one side of a slice of bread.	U	RR	
03	33:36	5 seconds to put a slice in or to take a slice out, and 3 seconds to turn a slice over. Investigate. Okay.	U	FR	Continue first reading
04	33:50	[Start drawing bread shape] First bread, first bread. Bread, draw the bread out. [Stop drawing]. [Write in bread drawing: ①] First bread. Go into the grill [draw an arrow from the bread and write after the arrow: grill]. Take 3 seconds [write beside arrow: take 3 seconds]	U	VI	
05	34:12	First, what must I investigate about?	P	PT	
06	34:14	[Write: 1st] I think I'll find [start writing] the total time taken, the total time taken to [stop writing]. To investigate [cancel: 1st] [write: to investigate]. [Continue writing] the total time taken to, to toast the three slices of bread, slices of bread [stop writing]. Finish.	P	PP1	Posed Specific Problem 1
07	34:59	So first bread go into the grill [point pencil at bread ① and then at the word 'grill'] take 3 seconds …	H	EM1	Error 1: Should be 5 seconds
08	35:06	Then at the same time [draw second bread] at the same time [write in bread drawing: ②] second bread also go into the grill [draw an arrow from the bread and write after the arrow: grill] same time [draw curly bracket for both slices and write: same time]. Take another 3 seconds [write beside arrow: take 3 seconds].	H	VI	
09	35:29	So 6 seconds already, to put 2 slices of bread [point to bread ①, then bread ②] under a grill [point to the words 'same time'].	H	RE	
10	35:33	[Pause for 4 seconds]	X	XP	
11	35:37	Then the grill [point to task statement] it takes 30 seconds to toast one side of a slice of bread …	H	RR	
12	35:47	Okay, so 30 seconds taken to toast [write: 30 seconds to toast] these two sides: this side [shade bread ②] and this side [shade bread ①]. 30 seconds [box up '30 seconds'] to take, to toast these two sides.	H	RE	

The process of assigning a stage code to each protocol line was not easy because some behaviours could appear in more than one stage, e.g. visualising information (VI) in Lines 04 and 08. Therefore, there was a need to read the surrounding protocols to infer that the student was visualising information to understand the task in Line 04 because she had not posed any problem to solve yet (the latter happened only in Lines 05-06). On the other hand, it could be inferred from the surrounding protocols that the student was visualising information to solve the problem in Line 08 because she had posed the problem earlier in Line 06, started solving in Line 07 and continued to solve in Line 09. Sometimes, there were grey areas. Therefore, there was also a need for an inter-coder reliability test for stage codes (see Section 4.7 later).

(b) Episodes

After all the stage codes had been assigned to the entire transcript, the protocols would be parsed into episodes at what was called an aggregate level by Newell and Simon (1992). A natural method to group the protocols would be according to the investigation stages. In other words, an episode is a cluster of related behaviours which are usually from the same investigation stage, but with certain exceptions which will be explained later in this section. For example, the first episode in the transcript in Table 4.4 consisted of Lines 01-04 as these lines corresponded to the stage of Understanding the Task (U), while the second episode consisted of Lines 05-06 as these lines corresponded to another stage, namely, Problem Posing (P). The third episode consisted of Lines 07-12 as it corresponded to the stage of Using Other Heuristics (H). A *double line* was then used in the transcript to separate the protocols into episodes.

Although the third episode in the transcript in Table 4.4 contained Line 10, which had the stage code X as it could not be determined what the student was thinking when she was silent, Lines 07-12 were grouped together as one episode based on the dominant behaviour of Using Other Heuristics (H). It would be very messy if such an episode was broken down into many smaller episodes if the student paused very often in between. Similarly, if there was a metacognitive behaviour (Category M) in between a cluster of related behaviours, then the metacognitive behaviour would be grouped together with the related behaviours as one episode. But if there was a cluster of metacognitive behaviours (Category M) and/or other behaviours (Category X) in between two different stages, they would be grouped together as a transition episode (labelled as X because all the stage codes were X), just like the transition between stages in Schoenfeld's (1985) Timeline Representation described in Section 2.3.3(a).

Sometimes, it was possible for an episode to contain only one protocol line. For example, if a student had posed a specific problem (PP) immediately after the stage of Understanding the Task (U), then the episode for Problem Posing (P) would consist of only one protocol line. However, it was found that the students went into a cycle of specialising and conjecturing very often when they tried examples (specialising) to search for patterns (conjecturing). For example, the transcript in Table 4.5 shows an episode consisting of the stage of Specialising (S) and the stage of Conjecturing (C) interspersed together in such a way that it was difficult to separate these two stages. If an episode could only correspond to one stage, then the transcript in Table 4.5 would consist of 15 episodes, with most episodes containing only one protocol line each. Thus it was more logical to group the stages of Specialising (S) and Conjecturing (C) as one single episode containing related behaviours of Specialising and Conjecturing (S/C), rather than separating each line into an episode.

Table 4.5 Sample Transcript showing Episode of Specialising and Conjecturing

Line	Time	Protocols	Stage Code	Bhvr Code	Remarks
90	12:30	So I try ... um, 3 ... I try [start writing] 36 ... 36 is 3 + 6 = 9; so 36 + 9 = 45 [stop writing].	S	TE3	
91	12:48	45 can be divided by 3.	C	OP2	Pattern 2
92	12:50	So [continue writing] 4 + [stop writing] ... so 4 + ... um [continue writing] 5 = 9; 45 + 9 = 54 [stop writing].	S	TE3	
93	13:01	These numbers … can be divided by 9.	C	OP3	Pattern 3
94	13:06	[Continue writing] 5 + 4 = 9; 54 + 9 = 63; 6 + 3 = 9 [stop writing] …	S	TE3	
95	13:18	I find that all the … sums of the digits [underline all these sums] are 9 ...	C	OP3	Still Pattern 3
96	13:25	So [continue writing] 63 + 9 = 72 [stop writing].	S	TE3	
97	13:29	72 can be divided by 9 —	C	OP3	
98	13:31	— is [continue writing] 7 + 2 = 9; 72 + 9 = 81 [stop writing and underline the previous 9].	S	TE3	
99	13:39	These also can be divided by 9 ... It's 9 itself.	C	OP3	
100	13:44	So [continue writing] 81 is, um ... 8 + 1 = 9; 81 + 9 = ... 81 + 9 = 90 [stop writing].	S	TE3	
101	13:56	Also can be divided by 9 [draw a line below Example 3] ...	C	SP	
102	14:00	So ... now this one can be [start writing below Example 3: divided by 9] divided by ... divided by 9 [stop writing] ...	C	OP3	
103	14:10	Now I try ... now I should try ... another number ...	S	DP	
104	14:18	Example is ... 3 ... um ... 4 ... no, is ...	X	XH	
105	14:26	84 ... 84 is ... [point pen at 12 in following statement] 84 is 8 + 4 = 12, so [point pen at following numbers] 96, um, 15 ... 111 and 3 and 114 and 6 and ... 6 and 120 and 3 and 123 [stop pointing] ...	C	SP	Refer to Example 2

Similarly, for the stages of Justifying (J) and Generalising (G), the latter stage usually consisted of only one protocol line to determine whether the justification had led to a generalisation (SG) or no generalisation (NG). The working for this protocol line was very often an integral part of the previous working in the justifying stage, so all the working should be treated as one single episode consisting of related behaviours of justifying and generalising. Therefore, it was decided that an episode would usually correspond to an investigation stage, but with the following exceptions: Specialising (S) and Conjecturing (C) would be grouped as an episode; and similarly for Justifying (J) and Generalising (G).

4.5 PHASE 3: REFINEMENT OF REVISED CODING SCHEME BASED ON MORE EMPIRICAL DATA

The Revised Coding Scheme was put to the test by using another four transcripts of thinking-aloud protocols from the pretest in the pilot study. The four transcripts were coded for both the behaviour codes and the investigation stage codes, and it was found to be satisfactory. Since no new behaviour was found and there was no other problem with the Revised Coding Scheme, there was no change to the coding scheme. In order not to use too many different names, the coding scheme at the end of Phase 3 was still called the Revised Coding Scheme. But it was found from all the eight transcripts of pretest protocols that some behaviours in the Revised Coding Scheme had still not been observed, e.g. using algebra (AL) and analysing feasibility of the plan (MF), so it was not possible to include actual protocols of such behaviours as exemplars inside the Revised Coding Scheme. I then checked through the remaining two pretest transcripts from the pilot study but still could not observe these 'missing' behaviours, since it was not expected that the students would exhibit a full range of processes during the pretest because investigation was not easy for students who had no prior experience in it. As the students might display a richer range of behaviours during the posttest after the developing lessons, some posttest transcripts from the main study would be used to refine the Revised Coding Scheme.

4.6 PHASE 4: DEVELOPMENT OF FINAL CODING SCHEME BASED ON NEW DATA FROM POSTTESTS IN MAIN STUDY

The use of the posttest data from the main study to fine-tune the Revised Coding Scheme further was for two purposes: (i) to find actual protocols for behaviours in the

Revised Coding Scheme that had not been observed from all the pretest transcripts from the pilot study, and (ii) to find out if there was any more new behaviour that could not be coded using the Revised Coding Scheme. In order to code a wider range of behaviours so as to inform the Revised Coding Scheme, I had to look through the posttest transcripts of thinking-aloud protocols for all the students in order to sieve out four transcripts of 'richer' protocols which hopefully contained a fuller range of behaviours[6]. Two transcripts of protocols were for Posttest Task 1 (Type A task) and the other two transcripts for Posttest Task 2 (Type B task).

4.6.1 Fine-tuning the Revised Coding Scheme

During the coding process using the four posttest transcripts from the main study, behaviours that had previously not been observed in the pretest samples from the pilot study, such as using algebra (AL) and analysing feasibility of the plan (MF), had all been observed in the posttest samples. This had made it possible to include exemplars of protocols for these behaviours in the Final Coding Scheme.

Furthermore, a new metacognitive behaviour was found among the posttest samples that was not included in the theoretical investigation model for metacognitive processes developed for the present study (see Section 3.2.2) and so it could not be coded using the Revised Coding Scheme. Some students were observed to have some kind of proactive metacognitive awareness, resulting in constantly being aware or conscious of what they were doing, including the ability to sense something amiss when it happened, which caused them to pause and check. Sometimes, the student

[6] It was not possible to use all the posttest protocols to fine-tune the Revised Coding Scheme because it was necessary to do the inter-coder reliability test for the Final Coding Scheme before the latter could be applied to code all the transcripts.

might discover a mistake, but at other times, the result might turn out to be correct. So this kind of metacognitive behaviour was different from the outcome 'Discovered Error or Mistake' (ED). In literature on metacognition reviewed earlier in Section 2.3.1, most researchers (e.g. Brown et al., 1983; Schraw, 2001) made a distinction between two types of metacognition: 'knowledge of cognition' and 'regulation of cognition'. Although the present study focused on the 'regulation of cognition' during investigation, it was found that some students were able to *actively* apply their 'knowledge of cognition', which was also called 'metacognitive awareness'. Thus there was a need to code this new behaviour as MA. As this was a new significant finding for the present study, it will be examined in more detail using actual protocols during the data analysis in Chapter 7 (see, e.g., Section 7.2.3).

4.6.2 Final Coding Scheme

The Final Coding Scheme was obtained from the Revised Coding Scheme by adding one new code, and some exemplars of protocols for behaviours not observed earlier using the pretest data from the pilot study. Table 4.6 shows the Final Coding Scheme. As this was the Final Coding Scheme[7], the table includes some explanations and exemplars of actual students' thinking-aloud protocols for each behaviour. There were a total of 40 codes: 26 codes for cognitive behaviours, 6 codes for metacognitive behaviours, and 8 codes for other behaviours. Since some behaviours could occur in more than one stage, a summary of the possible cognitive and metacognitive behaviours for each stage is shown in Table 4.7.

[7] Actually, the product at the end of this phase (Phase 4) was only the Preliminary Coding Scheme, not the Final Coding Scheme. But since there was no change to the Preliminary Coding Scheme in the last phase (Phase 5) which will be discussed in the next section, the Preliminary Coding Scheme was the final one. In order not to use too many different names, the product at the end of Phase 4 was simply called the Final Coding Scheme.

Table 4.6 Final Coding Scheme

Legend: *There are three categories: Category C, Category M and Category X.*
Non-italic codes represent processes; codes in italics represent outcomes.
Exemplars of actual student thinking-aloud protocols are in quotes and italics, while my comments are in square brackets.
'...' in thinking-aloud protocols means a short pause lasting three seconds or less.

Category C: Cognitive Behaviours

Code	Explanation	Exemplars of Protocols
FR: First Reading of Task	• Occurred only once but different from re-reading task	
RR: Re-reading Task	• Re-reading task or parts of the task for various reasons: to understand the task; to find a problem to solve or extend; to think of a plan to solve a problem or to justify a conjecture; or to monitor understanding or progress • Include missing out some words while re-reading; or preceded by the phrase "the question says"	• *"I cut 12 identical sausages … so that I share them equally among 18 people."* [student missed out three words while re-reading original task statement which is "I need to cut 12 identical sausages so that I can share them equally among 18 people"] • *"The question says choose any number."* [Original task statement is: "Choose any number."]
RT: Rephrasing Task	• Rephrasing or paraphrasing task or parts of the task for various reasons: to understand the task; to find a problem to solve or extend; to think of plan to solve a problem or to justify conjecture; or to monitor understanding or progress • Paraphrase might contain words from original task statement, but it must be preceded by phrases such as "in other words" or "so I must [do this or that]"	• *"I have 12 identical sausages and ... I must share them among 18 people so that … each of them has an equal amount."* [different from original task statement in above cell] • *"So I must add the sum of its digits to the number."* [Original task statement is: "Add the sum of its digits to the number."]
HI: Highlighting Key Information	• Highlighting key information in task statement by underlining, circling, or boxing up some numbers or words	• *"[Underline the phrase: 12 identical sausages]"* • *"[Circled the number 18]"*
VI: Visualising Information	• Visualising information by drawing a diagram for various reasons: to understand the task; to think of a way to solve a problem or to justify a conjecture	• *"Example [start drawing a rectangle] this is a sausage."*
PT: Thinking of Problem to Pose or Extend	• Thinking of what problem to pose without changing the given, or to extend by changing the given • Did not distinguish between thinking of problem to pose and thinking of problem to extend because student can be thinking of any problem	• *"Now my ... problem is that …"* • *"So ... investigate ... what?"* • *"Now can I extend this question?"* • *"What is there to extend?"*
PP: Posed Problem	• This outcome described the situation where the student had posed either the general problem to search for any	• *"My task is to find the pattern for these numbers."* [PP0, i.e. posed general problem]

Code	Explanation	Exemplars of Protocols
	pattern (labelled as PP0), or a specific problem (labelled as PP1, PP2, etc.) without changing the given in the original task • Different from *Posed Problem to Extend (EP)*	• *"I find the sum of digits first."* • *"So, now I will just investigate how long and what is the fastest way to get these 3 slices of bread to be toasted."*
EP: Posed Problem to Extend	• This outcome described the situation where the student had posed a problem to extend the task by changing the given	• *"Now I should extend this problem. My question is that what if there are 12 people and 18 sausages?"* [original task was 18 people and 12 sausages]
TP: Thinking of Plan	• Thinking of a plan for various reasons: to understand the task; to solve a problem; to search for patterns; or to justify a conjecture, etc.	• *"So how should I solve the problem?"* • *"And to split 6 sausages among 10 people, you can … uh, split it … you can … can … ok, I'll try thinking …"* • *"After these 30 seconds … I will … what should I do ah? Should I take out both? Or just turn one and take out another one?"* • *"So how I prove it? So … as you can … it's a little bit hard to prove … because from what I see here … there is no really, no link, there is no real link …"*
DP: Decided on Plan	• Decided on a plan for various reasons: to understand the task; to solve a problem; to search for patterns; or to justify a conjecture, etc. • Plan includes understanding the task by trying examples; and solving a problem by various means such as specialising, reasoning, algebra, etc.	• *"I must try some examples to understand the task first."* • *"So … I'll just choose randomly."* [to understand task] • *"Let me try using algebra to solve the problem."* • *"How to prove is that I can draw a diagram."* • *"I think I shall search for a new pattern instead."*
TE: Trying Example	• Trying an example (labelled as TE1, TE2, etc.) for various reasons: to understand the task; to search for patterns; to find a counter example to reject a pattern or a conjecture (naive testing)	• *"I try 21 [write: 21 → 2 + 1 = 3] is 3, right? Then [start writing] 21 + 3 = 24 [stop writing]; 2 + ... 4 = 6; then 21 ... 24 + 6 = 30."*
PC: Performing Calculation	• Performing calculation (except during trying examples), e.g. to search for patterns during conjecturing • If performing calculation during trying examples, classify under Trying Example (TE)	• *"[Does long division for 55 ÷ 3]"* [to search for patterns during conjecturing, not trying examples]
RE: Using Reasoning	• Using reasoning for various reasons: to solve a problem with or without formulating a conjecture; or to justify a conjecture	• *"There will be 36 pieces. Yeah, so [start writing] 36 ÷ 18 = 2 [stop writing]. So everyone will have … 2/3, is it?"*

Code	Explanation	Exemplars of Protocols
AL: Using Algebra	• Using algebra for various reasons: to solve problem with or without formulating a conjecture; or to justify conjecture	• *"[Start writing] a + z + y [stop writing] … [cancel: a + z + y] … uh … [start writing] a + z [stop writing]. Eh [cancel: a + z]."*
SP: Searching for Pattern	• Searching for any pattern among the examples generated, including not finding any pattern	• *"What's the pattern for these numbers?"* • *"So you see this one is 28 [point to first 28 in Example 2] 28, um [point to the following sums of digits in Example 2] 5, 10, 11 [stop pointing]. No pattern what."*
OP: Observed Pattern	• This outcome described the situation where student had observed or discovered a pattern (labelled as OP1, OP2, etc.), including modifying an observed pattern which could be considered as observing a new pattern • Different from Formulating Conjecture (FC) because subsequent protocols suggest student was still trying examples to be more certain of the pattern, and not yet trying to prove that the observed pattern is the underlying pattern	• *"I think that the sum of digits can be divided by 3. But I try another number first."* [first sentence was coded as OP because second sentence, coded as DP (Decided on Plan), suggested that student was not so sure of the pattern yet, so she continued to try another example to be more certain of the pattern before treating it as a conjecture at a later stage]
RP: Rejected Observed Pattern	• This outcome described the situation where student had rejected an observed pattern based on counter example or any other reason (labelled as RP1, RP2, etc., where the number tallies with the number in OP above, not in order of rejection) • Different from *Refuted Conjecture (RC)* during justifying	• *"So cannot, 11 cannot be divided by 3."* [i.e. student rejected observed pattern that number can be divided by 3 because of counter Example 11]
FC: Formulated Conjecture	• Formulated conjecture (labelled as FC1, FC2, etc., where number follows order of formulating, not according to OP), including modified a conjecture which could be considered as formulated a new conjecture • Different from *Observed Pattern (OP)* because subsequent protocols suggested student went on to prove conjecture	• *"Ok, the conjecture [write: Conjecture:] for me now is that … um [start writing] all 2 digits numbers will add up to a new odd number [stop writing]."* • *"Now it starts from 16 again ... so ... so ... it will continue to do that until the next 16 ... But ... is it always the same?"* [student did not say this was a conjecture but he was more sure of the pattern as he obtained 16 *again*, and subsequent protocols showed that he went on to try to prove it]
VC: Verified Pattern or Conjecture Correct	• This outcome described the situation where student had verified that the pattern or conjecture was correct after naïve testing using more examples (this is not a proof) • Need to look at surrounding protocols	• *"Eh? So, 85 [write: 85]. So the whole thing repeats again. [Draw a vertical line from 85 in E.g. 13 downwards] It's the same."* [verified repeating pattern correct, and surrounding protocols suggest that she did not accept pattern as

Code	Explanation	Exemplars of Protocols
	that student did not accept the pattern as true or the conjecture as proven • If student accepted the pattern as true or the conjecture as proven based on naïve testing using more examples, it would be coded as EM (Made Error)	true because she later formulated it as a conjecture to be proven]
JC: Justified or Proven Conjecture	• This outcome described the situation where student had justified or proven a conjecture (labelled as JC1, JC2, etc., where number tallies with the number in FC above) • Need to look at surrounding protocols: JC must happen after student had formulated conjecture and then proved it correctly via two ways: using reasoning (RE) or algebra (AL)	• *"Ok, so it proves that my conjecture is correct …"*
RC: Refuted Conjecture	• This outcome described the situation where student had refuted a conjecture • Need to look at surrounding protocols: RC must happen after student had formulated conjecture and then proved it wrong via three ways: using reasoning (RE), algebra (AL), or counter example (TE) • Different from *Rejected Pattern (RP)* during conjecturing	• *"So this one is not working [cancel Conjecture 2]"*
SG: Solved Problem that led to Generalisation	• This outcome described the situation where student had correctly solved a problem that led to generalisation • Need to look at surrounding protocols: SG must happen after student had solved a problem via two ways: justified conjecture that was a general result; or solved problem without formulating any conjecture but still led to generalisation • Different from *Justified or Proven Conjecture (JC)* in two ways: student could justify conjecture without generalising; or student could justify conjecture that led to generalisation but still had not solved the problem yet	• *"Ok, so it proves that my conjecture is correct …"* [this correct conjecture was a general result: coded as both JC (Justified Conjecture) and SG] • *"Same for numbers that can be divided by 9. So this is my finding."* [this correct solution, without formulating any conjecture, was a general result: coded as SG only, not JC]
NG: Solved Problem without Generalising, i.e. No Generalisation	• This outcome described the situation where student had correctly solved a problem without generalising • Need to look at surrounding protocols: NG must happen after student had solved a problem via two ways: justified conjecture that was not a general result; or solved problem without formulating any conjecture and without generalising	• *"That means the person must cut one sausage into three identical pieces so that each person gets 2/3 of it."* [this correct solution, without formulating any conjecture, was not a general result]

Code	Explanation	Exemplars of Protocols
CW: Checking Correctness of Working	• Checking correctness of working step by step, just briefly glancing through it, or working backwards • Could happen in any stage, not just in the Checking Stage	• *"I check the numbers can be divided by 3 … Yes, these numbers can be divided by 3."* [student checked previous working to confirm this]
EM: Made Error or Mistake	• Made mistake such as calculation error, faulty reasoning, solved problem wrongly, misinterpreting the task, etc. • Include treating observed pattern or conjecture as true without proving, or wrongly accepted conjecture as true based on naïve testing or faulty reasoning • Abbreviation EM stood for Error Made (did not use 'Made Mistake' as it would give rise to the code MM, but codes with first letter M were reserved for metacognitive behaviours)	• *"[Start writing] 33 + 3 + 3 … = 36 [stop writing]."* [calculation mistake: should be 39] • *"So each person gets 18 parts."* [faulty reasoning: should be 12 parts] • *"So the new number is 4."* [misinterpreted task: this was only the sum of digits] • *"From the examples, we can see that my conjecture is correct."* [wrongly accepted conjecture as true based on naïve testing]
ED: Discovered Error or Mistake	• This outcome described the situation where the student had discovered a mistake	• *"Oh, this is wrong!"*

Category M: Metacognitive Behaviours

Code	Explanation	Exemplars of Protocols
MU: Monitoring Understanding	• Monitoring student's own understanding of task by clarifying task requirements, given conditions, or meaning of some parts of task • Should only occur in the first stage of Understanding the Task, but students could stop anytime in subsequent stages to go back to the first stage	• *"Did I interpret the task correctly?… Yes, I think so."* • *"So let me try another one to further understand what the task is trying to tell me."*
MG: Analysing Feasibility of Goal	• Analysing whether a goal or problem (including extension) was feasible or worth pursuing, e.g. is the problem interesting, too difficult or too trivial?	• *"But this one, how to have formula? … Um, this one is find the LCM first. Actually no need. Yah. Eh? …"* [student analysed whether feasible to find formula]
MF: Analysing Feasibility of Plan	• Analysing whether a plan was feasible or worth pursuing	• *"But I don't think there is any relationship this way … Maybe I can use some other ways of finding the relationship like, um … maybe like adding the digits up."* [analysed different plans to find relationship] • *"I already reach 1000, so I don't think I should continue further as I am unable to find … that will be difficult. Narrow my range down to below 1000 …"* [analysed feasibility of trying beyond 1000]

Code	Explanation	Exemplars of Protocols
MP: Monitoring Progress	• Monitoring progress by reviewing whether the plan was on the right track, including deciding to continue or to change the plan without analysing its feasibility • If reviewing progress *after* solving the problem in the *Review Phase* to see if it had achieved the goal, this would be coded as Reviewing Solution (MR) • If analysing *feasibility* of current plan while monitoring progress, the analysis would be coded as Analysing Feasibility of Plan (MF)	• *"Am I going on the right track? … I don't think I know … Maybe I'll try for a while more and see how it goes …"* • *"I think I'm stuck. I think I'm not doing correctly."* • *"I think I'm going nowhere … Should I think of a new approach instead?"* • *"Yes, I think I am in the right way but now I can't find the pattern yet."*
MR: Reviewing Solution	• Reviewing solution after solving a problem or part of a problem to see if it had achieved the goal or solved the problem, including examining whether the answer was reasonable or logical, evaluating the efficacy of a method of solution, and looking for alternative methods • Different from simply Checking Correctness of Working (CW) which was cognitive • Checking whether *partial* solution was in line with the goal comes under monitoring progress (MP) • If the goal was to find different methods, then looking for alternative methods is not under MR but under problem posing	• *"Ok, actually, I have only answered part of the what the question wants me to do …"* • *"Ok, I am referring back … That seems not the real task is wanting me to find. Ok … so I think that my second conjecture is a little bit off topic."* [student reviewed solution after proving second conjecture] • *"Ok, this [method of dividing the sausage or the solution] is fair."* [student found solution reasonable] • *"I can use another method."* [after solving problem using one method]
MA: Metacognitive Awareness	• The ability of the student to apply their knowledge of cognition proactively in investigation, resulting in constantly being aware or conscious of what he or she was doing, including the ability to sense something amiss when it happened, as well as new information that evolved • Need to look at subsequent protocols: MA usually happened when student obtained a questionable result or outcome, which caused him or her to pause and check; or when student was aware that he or she had obtained a new piece of information • Sometimes MA was accompanied by Discovered Error (ED), but MA was different from ED because the student might not discover the error, or it might turn out that there was actually nothing wrong	• *"Hmm ... this number appears again."* [but this number was on the previous page and the student was aware that it had appeared before without even referring to it] • *"16. Eh ... this number has not appeared before?"* [student was aware that she had obtained a number which had not appeared before and she turned to the previous pages just to confirm] • *"Now, now, this is strange, right?"* [after getting a result] • *"But the problem is …"* [student sensed something amiss because he realised that it was a problem, then he discovered the error] • *"So if he get other 6 pieces —"* [student sensed something amiss, so she paused for four seconds, re-read the task and discovered the error]

Category X: Other Behaviours

Code	Explanation	Exemplars of Protocols
XP: Pausing	• Student was silent for more than 3 seconds, including any sound like 'uh' or 'um' in between the pause; shorter pause lasting 3 seconds or less was indicated as '...' in the transcript and not coded separately	• *"[Pause 5 s]"* • *"[Pause 8 s, including an 'uh' in between]"*
XH: Hesitating	• Hesitating or doing nothing constructive, suggesting student might be stuck • Protocols might contain sounds like 'uh' or 'um', but if the sound was part of a pause lasting more than 3 seconds, it would be coded separately as Pausing (XP) • Protocols might contain repetitions of words which contained nothing new, so this was different from Thinking of Plan (TP) to solve a problem	• *"So what should I do is to ... what I should do is to ... um ... I don't know what to do."* • *"Because the ... odd, odd number ... odd number is ... odd number is ... um ..."* • *"So ... as for one, one piece of sausage, sausage ... then ... example is, um, totally is ..."*
XC: Referring to Given Checklist	• Referring to given checklist (see Appendix H), including reading from it when stuck or thinking what to do next	• *"[Refer to checklist p. 1 for 9 s]"* • *"[Read from given checklist] Understand the task first [stop reading]."*
XR: Re-reading What was Written	• Re-reading what the student had written himself or herself, such as their working, solution, problem posed, or conjectures formulated	• *"So if the original number can be divided by 3 or 9, the new numbers obtained also can be divided by 3 or 9."* [student re-read conjecture that she had written]
XW: Re-writing or Recording Processes	• Recording or organising processes that were not part of the thinking processes, e.g. rewriting part of solution from previous page onto new page for ease of reference; writing full solution properly after solving a problem; or organising examples by labelling or boxing them up separately	• *"[Student rewrites full solution after solving problem.]"* • *"[Student turns to new page and rewrites previous working.]"* • *"[Student boxes up different examples.]"*
XO: Off-Task Behaviours	• Behaviours that were not task-related, e.g. invigilator told student to speak louder or to continue to think aloud; or student just said some irrelevant things	• *"[Invigilator tells student to speak louder]"* • *"[Invigilator tells student that she has one minute left]"*
XA: Affective Behaviours	• Affective behaviours such as student expressing frustration, joy or opinion about the test (although beyond scope of present research to study, affective behaviours still needed to be coded)	• *"[Student sighs]"* • *"Why is this problem so difficult?"* • *"I give up!"* • *"Hmm, that is pretty interesting."*
XU: Unable to Code	• Protocols where it was not possible to infer what the student was doing or thinking about, including student mumbling and unintelligible words	• *"And then … um, let's see [flip to p. 2] … um … 37, 21."* [Unable to decide whether it was XH or SP, so coded as XU] • *"[Unintelligible words]"*

Table 4.7 Summary of Possible Behaviours in Each Investigation Stage

Legend: Non-italic codes represent processes; codes in italics represent outcomes.
Other behaviours from Category X, and codes that could occur in any stage (e.g. Made Error or Mistake), are not included in this table.

Phase	Stage	Possible Cognitive Behaviours	Possible Metacognitive Behaviours
Entry	**Stage 1: Understanding the Task (U)** Trying to make sense of the task, including trying examples and clarifying task requirements	• FR: First Reading of Task • RR: Re-reading Task (to understand the task or to monitor understanding) • RT: Rephrasing Task (to understand the task or to monitor understanding) • HI: Highlighting Key Information • VI: Visualising Information (to understand the task) • TE: Trying Example (to understand the task) • TP: Thinking of Plan (to understand the task) • *DP: Decided on Plan (to understand the task)*	• MU: Monitoring Understanding
	Stage 2: Problem Posing (P) Posed the general problem of searching for any pattern, or trying to pose a specific problem to solve	• PT: Thinking of Problem to Pose • RR: Re-reading Task (to find a problem to pose or to analyse feasibility of pursuing the problem or the goal) • RT: Rephrasing Task (to find a problem to pose or to analyse feasibility of pursuing the problem or the goal) • *PP: Posed Problem*	• MG: Analysing Feasibility of Goal
Attack	**Stage 3: Specialising and Using Other Heuristics (S/H)** Trying examples with the intention of searching for patterns, or using heuristics other than specialising to solve a problem	• TE: Trying Example (with the intention to search for patterns) • TP: Thinking of Plan (to solve a problem) • *DP: Decided on Plan (to solve a problem)* • RR: Re-reading Task (to think of a way to solve a problem or to monitor progress) • RT: Rephrasing Task (to think of a way to solve a problem or to monitor progress) • VI: Visualising Information (to help in solving a problem) • RE: Using Reasoning • AL: Using Algebra	• MF: Analysing Feasibility of Plan • MP: Monitoring Progress
	Stage 4: Conjecturing (C) Searching for patterns, including observing or rejecting a pattern, and formulating a conjecture	• SP: Searching for Pattern • TP: Thinking of Plan (on how to search for pattern) • *DP: Decided on Plan (on how to search for pattern)* • RR: Re-reading Task (to think of a way to find a pattern or to formulate a conjecture, or to monitor progress) • RT: Rephrasing Task (to think of a way to find a pattern or to formulate a conjecture, or	• MF: Analysing Feasibility of Plan • MP: Monitoring Progress

Phase	Stage	Possible Cognitive Behaviours	Possible Metacognitive Behaviours
		to monitor progress) • VI: Visualising Information (to search for pattern or to formulate conjecture) • *OP: Observed Pattern* • *RP: Rejected Observed Pattern* • *FC: Formulated Conjecture*	
	Stage 5: Justifying (J) Trying to justify a conjecture by a non-proof argument using reasoning or a formal proof using algebra, including refuting a conjecture or naïve testing	• TP: Thinking of Plan (to justify a conjecture) • *DP: Decided on Plan (to justify a conjecture)* • RE: Using Reasoning • AL: Using Algebra • RR: Re-reading Task (to think of a plan to justify a conjecture, or to monitor progress) • RT: Rephrasing Task (to think of a plan to justify a conjecture, or to monitor progress) • VI: Visualising Information (to help in justifying a conjecture) • TE: Trying Example (to find counter example to refute a conjecture in naïve testing) • *VC: Verified Pattern or Conjecture Correct* • *JC: Justified or Proven Conjecture* • *RC: Refuted Conjecture*	• MF: Analysing Feasibility of Plan • MP: Monitoring Progress
	Stage 6: Generalising (G) Solved problem which may or may not be a general result	• *SG: Solved Problem that led to Generalisation* • *NG: Solved Problem without Generalising, i.e. No Generalisation*	
Review	**Stage 7: Checking (R)** Checking working and reviewing solution	• CW: Checking Correctness of Working • RR: Re-reading Task (to review whether solution had achieved the goal) • RT: Rephrasing Task (to review whether solution had achieved the goal)	• MR: Reviewing Solution
	Stage 8: Extension (E) Trying to extend the task by changing the given	• PT: Thinking of Problem to Extend • RR: Re-reading Task (to find a problem to extend or to analyse feasibility of pursuing the problem or the goal) • RT: Rephrasing Task (to find a problem to extend or to analyse feasibility of pursuing the problem or the goal) • *EP: Posed Problem to Extend*	• MG: Analysing Feasibility of Goal

4.7 PHASE 5: INTER-CODER RELIABILITY TEST FOR FINAL CODING SCHEME

As explained earlier in Section 2.4.3, there was a need to ensure that the coding scheme was reliable in a qualitative study like the present research (Lesh et al., 2000). One way to do this was to perform an inter-coder reliability test. If the inter-coder reliability was high, it would suggest that the coding scheme was reliable. Since the coding would be used to inform the investigation models developed for the current study, then the models would also be reliable and the descriptive power of the models would be high (Schoenfeld, 2002). Four transcripts of thinking-aloud protocols were selected for two coders to code. The transcripts were chosen from the posttest tasks in the main study because the students exhibited a lot more processes after the teaching experiment. In fact, the combination of the four transcripts included 38 of the 40 behaviour codes and all 10 stage codes. Two of the transcripts were for Posttest Task 1 (Type A task) and the remaining two for Posttest Task 2 (Type B task), since the two types of tasks would tend to elicit different processes. The two coders were mathematics educators with some experience in coding.

(a) Preparation of Transcripts and Test Answer Scripts for Coders

Table 4.8 shows part of a sample transcript given to the two coders. It contained two empty columns, one for stage code and the other for behaviour code. The Remarks column contained only some necessary objective information, such as the page number of the students' answer scripts, and the example number for the examples that the students had tried (see Lines 01 and 05). As explained earlier in Section 4.1, it was necessary to label the pages of the students' answer scripts and the examples generated by the students because the students usually did not do so, and sometimes there was a need to refer to a particular page or example when transcribing the actions of the students. This *objective* information served to help the coders make sense of what the students were doing so that they would be in a better position to assign more appropriate codes to the protocols. Unlike the full coded transcript in Appendix I, the sample transcripts given to the coders did *not* contain other subjective information in the Remarks column, or double lines to separate the protocols into episodes as described in Section 4.4.4(b), so as not to influence their judgment.

Table 4.8 Sample Transcript of Student's Protocols Given to Two Coders for Inter-Coder Reliability Test

Line	Time	Protocols	Stage Code	Bhvr. Code	Remarks
01 (p.1)	00:00	Choose any number. Add the sum of its digits to the number itself to obtain a new number. Repeat this process for the new number and so forth. Investigate.			Answer Script Page 1
02	00:11	[Teacher tells her to speak louder]			
03	00:12	Choose any number. Add the sum of its digits to the number itself to obtain a new number. Repeat this process for the new number and so forth. Investigate.			
04	00:23	So ... now I must, uh, try some examples first ...			
05	00:28	For example, I choose 12 [write 12].			Example 1: p. 1 column 1 in answer script

In addition, each coder was given a copy of the following:

- Students' answer scripts for all the four samples for easy reference of their working (the examples in the answer scripts were also separated and labelled clearly with a number, as shown in the sample answer script in Appendix J);
- Final Coding Scheme (see Table 4.6);
- Summary of Possible Behaviours in Each Investigation Stage (see Table 4.7).

(b) Instructions for Coders

The two coders were to assign both the behaviour code and the stage code for each protocol line in the four transcripts independently, and they did not meet each other. I first briefed the two coders on separate occasions on the different codes in the Final Coding Scheme (see Table 4.6), and then explained how to use the Summary of Possible Behaviours in Each Investigation Stage (see Table 4.7) and the students' answer scripts to help in coding the transcripts. I then walked through the coding of two short excerpts of transcripts (which were different from the four transcripts used for the inter-coder reliability test) with the coders to help them understand the coding process. During the actual coding by the two coders, they were free to discuss with me if they had further doubts about the meanings of the codes in the coding scheme. I would only explain the codes, but I would not suggest any code to code any protocol line, or say anything that might influence the coding of the two coders.

The inter-coder reliability test did not take into account the numbering for observed patterns (OP) or formulated conjecture (FC) because the coders might disagree with me that a student had observed a pattern or formulated a conjecture at a particular protocol line, and so their subsequent numbering of observed patterns and formulated conjectures would definitely be different from my numbering. The issue was not the numbering of subsequent observed patterns or formulated conjectures, e.g. whether it should be OP2 or OP3, but whether the coders agreed with me that those were observed patterns and so should be coded as OP.

(c) Results of the Inter-Coder Reliability Test

Table 4.9 shows the number and percentage of codes for Coder 1 and Coder 2 that were the same as my codes for the four sample transcripts.

Table 4.9 Inter-Coder Reliability Test for Coding Schemes

Transcript	Types of Codes	Total Number of Codes	Coder 1	Coder 2
Sample A	Stage Codes	157	$\frac{142}{157} = 90\%$	$\frac{146}{157} = 93\%$
	Behaviour Codes	157	$\frac{146}{157} = 93\%$	$\frac{143}{157} = 91\%$
Sample B	Stage Codes	172	$\frac{165}{172} = 96\%$	$\frac{164}{172} = 95\%$
	Behaviour Codes	172	$\frac{164}{172} = 95\%$	$\frac{168}{172} = 98\%$
Sample C	Stage Codes	174	$\frac{161}{174} = 93\%$	$\frac{154}{174} = 89\%$
	Behaviour Codes	174	$\frac{162}{174} = 93\%$	$\frac{151}{174} = 87\%$
Sample D	Stage Codes	120	$\frac{111}{120} = 93\%$	$\frac{115}{120} = 96\%$
	Behaviour Codes	120	$\frac{110}{120} = 92\%$	$\frac{104}{120} = 87\%$
Sub-Total	Stage Codes	623	$\frac{579}{623} = 93\%$	$\frac{579}{623} = 93\%$
	Behaviour Codes	623	$\frac{582}{623} = 93\%$	$\frac{566}{623} = 91\%$
Average of the Coders	Stage Codes	623	$\frac{579}{623} = 93\%$	
	Behaviour Codes	623	$\frac{574}{623} = 92\%$	
	Total	1246	$\frac{1153}{1246} = \mathbf{93\%}$	

The inter-coder agreement of 93% for the Final Coding Scheme for the present study was higher than the minimum standard of 80% for inter-coder reliability for media messages proposed by Riffe, Lacy and Fico (1998), thus suggesting that the Final Coding Scheme was reliable enough and hence there was no need to refine the coding scheme any further.

There were two reasons why it was possible to achieve such a high inter-coder agreement. The first reason was that there were many rounds of improving the Initial and Revised Coding Schemes substantially to strengthen its validity and reliability. Ambiguous codes, which were not easily observable or appeared to be similar, were either removed or re-defined more precisely so that they were more clearly recognisable. Detailed descriptions and representative exemplars of protocols were included for each code. Clarification of the differences between similar codes was also highlighted in the coding scheme so as to make it easier for any coder to distinguish between them. Possible behaviours that could occur in each investigation stage were also identified and summarised in a table form (see Table 4.7) in order to help any coder to identify the correct stage more easily. This process of meticulously refining the codes thoroughly was one of the most difficult and challenging tasks in the present research. Such careful deliberation had finally paid off in developing a reliable Final Coding Scheme that successfully passed the inter-coder reliability test.

The second reason was the choice of the two coders for the inter-coder reliability test. If the coders were not familiar with the coding process or had difficulty understanding the codes in the Final Coding Scheme, then it might not even be possible to pass the inter-coder reliability test. However, Coder 1 for the inter-coder reliability test for the present study was an experienced coder, while Coder 2, though less experienced, was very fast in understanding the codes in the Final Coding Scheme and distinguishing between codes that appeared similar just by listening to a short explanation. Therefore, the above reasons explained why the inter-coder agreement for the Final Coding Scheme was so high.

4.8 CONCLUDING REMARKS FOR THIS CHAPTER

Chapter 4 has explained how the coding scheme for the students' thinking-aloud protocols during investigation was developed from a five-phase combination of top-down (i.e. theoretically driven) and bottom-up (i.e. empirical driven) strategy. Starting with the Initial Coding Scheme constructed theoretically based on current literature, it was revised using empirical data from the pilot study and then richer posttest data from the main study to obtain the Final Coding Scheme, which had successfully passed the inter-coder reliability test for both the behaviour and stage codes. Chapter 5 will then describe the design of some data analysis instruments.

CHAPTER 5: DEVELOPMENT OF DATA ANALYSIS TOOLS

The two main sources of data for the present study were the students' answer scripts, and the video recordings of the students' thinking-aloud protocols and other non-verbal actions, during the pretest and the posttest. The protocols had been transcribed and coded using the coding scheme constructed in the previous chapter. According to Miles and Huberman (1994), there was a need to simplify the data in the coded transcripts to analyse them more effectively. Therefore, this chapter will describe the development of some data analysis tools for the present study.

5.1 PROTOCOL ANALYSIS METHOD

The protocol analysis method employed for the present study used the three processes in data analysis advocated by Miles and Huberman (1994) as a guide. The three processes were (i) data reduction, (ii) data display, and (iii) conclusion drawing and verification. The video recordings of the students' thinking-aloud protocols and other non-verbal actions during the pretest and the posttest had been transcribed and coded using the coding scheme developed in the previous chapter in order to reduce the amount of data to a form suitable for protocol analysis. Then the reduced data for the verbal protocols would be displayed in visual forms. Three types of diagrams, or data analysis tools, would be developed in the next two sections:

(i) Investigation Pathway Diagrams (IPD),

(ii) Investigation Timeline Representations (ITR),

(iii) Summary Tables of Processes and Outcomes (TPO).

Although the transcripts also made reference to the students' working in their answer scripts, the working was not easily discernible in the transcripts as the working was interspersed with a myriad of the students' verbal protocols. Therefore, there was still a need to refer to the students' answer scripts when developing the TPO. The three data analysis tools will be used to draw conclusions about the nature of cognitive and metacognitive processes in mathematical investigation, and to verify the theoretical investigation models, so as to answer the research questions for the present study. However, there was a need to design a fourth data analysis tool to measure the proficiency of the students' performance in investigation in order to compare the development of their processes quantitatively using descriptive statistics:

(iv) Investigation Scoring Rubric (ISR).

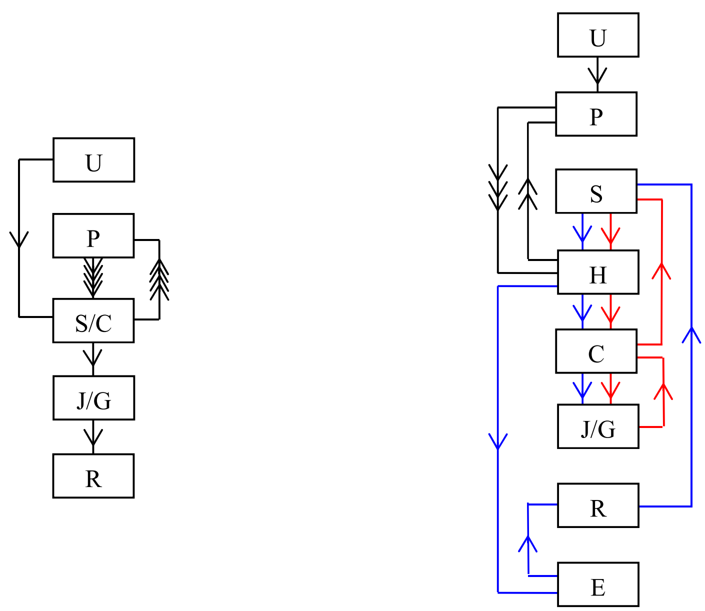
This chapter will only describe the development of the first three data analysis tools. The last instrument ISR will be described in Chapter 8 during the data analysis of the development of investigation processes because there is a need to understand the nature of investigation processes using the first three data analysis tools before developing the last instrument.

5.2 INVESTIGATION PATHWAY DIAGRAM AND INVESTIGATION TIMELINE REPRESENTATION

This section will describe two instruments developed to display the reduced data for the thinking-aloud protocols to aid in the data analysis for the present study, according to the protocol analysis method proposed by Miles and Huberman (1994) as described in the previous section. The two data analysis tools are the Investigation Pathway Diagram (IPD) and the Investigation Timeline Representation (ITR).

The investigation model for cognitive processes developed for the current research (see Section 3.2.1) describes the theoretical interactions or pathways between the various processes. So there is a need to examine what the students actually did during the present study in order to analyse the interactions of their processes. Thus the first data analysis tool, IPD, attempts to trace each student's actual pathways during an investigation. Figure 5.1 shows the IPD for a student's (S5) investigation of Posttest Task 1 (Kaprekar; Type A) and another student's (S9) investigation of Posttest Task 2 (Sausage, Type B). The reasons for grouping the Specialising and Conjecturing stages as the S/C stage for Type A tasks, and grouping the Justifying and Generalising stages as the J/G stage for both types of tasks, have been explained in Section 4.4.4(b). The IPD were constructed by extracting the relevant information from the transcripts that had been parsed into episodes using the stage codes as described in Section 4.4.4.

Legends: The no. of arrows in IPD indicates the no. of times the student went through the pathway. If there was more than one pathway from one stage to another in IPD, the black pathway occurred first, followed by the blue pathway and then the red pathway.



(a) Posttest Task 1 for S5 (b) Posttest Task 2 for S9

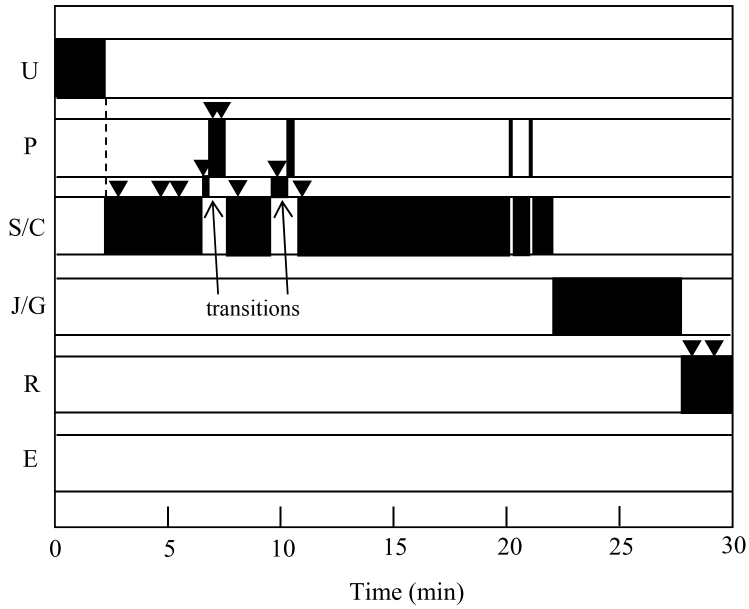
Figure 5.1 Investigation Pathway Diagrams (IPD)

As can be seen from the IPD in Figure 5.1, the students went through some pathways more than once, and so the number of arrows along a pathway will indicate the number of times the students went through the same pathway. For S5, it is clear from the IPD that the student started from the stage of Understanding the Task (U) and then proceeded to the Specialising / Conjecturing (S/C) stage. After that, she went through the loop between the S/C stage and the Problem-Posing (P) stage four times, before going down to the Justifying / Generalising (J/G) stage and then to the Checking (R) stage. It is fairly clear from the IPD that she could not have possibly gone down to the J/G stage before finishing the four cycles between the S/C stage and the P stage since there is no pathway from the J/G stage back to the S/C stage or the P stage.

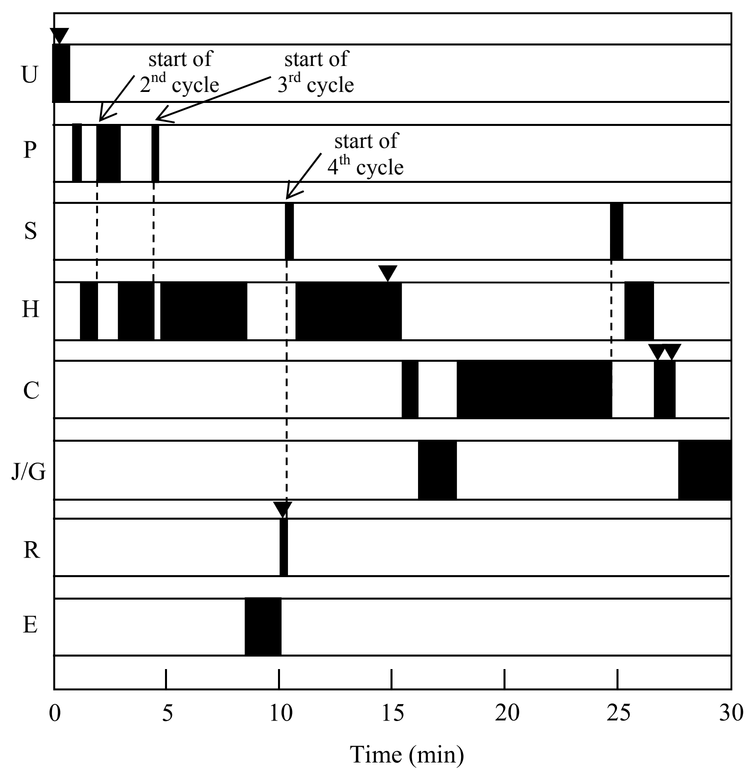
However, if the IPD is more complicated like that of S9 in Figure 5.1, then there is a problem. He started at the U stage and then proceeded to the P stage, followed by the stage of Using Other Heuristics (H). Then it is unclear from his IPD whether he went back from the H stage to the P stage, or he went through the following pathway: H → C → S → H → P. Thus there is a need to use different colours to indicate which pathways occur first. As there are so many pathways, it is not possible to use a different colour for each pathway. Instead, a different colour is used only when there are ambiguities. It was found that three colours were sufficient for the IPD of S9: black first, followed by blue, and then red. With this colour scheme, it is evident from the black pathways that he finished all the cycles between P and H first: U → P → H → P → H → P → H. In fact, after this, he did not go from H → C → S → H as described earlier because this is a red pathway. Instead, he went from H → E → R → S → H as indicated by the blue pathway. It is left to the reader to trace the remaining pathways which will end in the J/G stage.

The limitation of the IPD is that it is not possible to display how much time the students had spent in each stage or episode. Moreover, even with the colour scheme, it would still take quite a while to trace the student's pathways. Therefore, there was a need for another data analysis tool to complement the IPD.

The Investigation Timeline Representation (ITR) was modelled after the timeline representation developed by Schoenfeld (1985) described earlier in Section 2.3.3(a). Figure 5.2 shows the ITR for the investigation of Posttest Task 1 (Kaprekar) by S5, and the investigation of Posttest Task 2 (Sausage) by S9. A dotted line was used to help the reader see the pathway from one stage to another stage when the two stages were further apart. Metacognitive behaviours were indicated by an inverted triangle ▼ at the junctures where they occurred. The ITR also shows the transitions between two stages, which had been explained in Section 4.4.4(b). Where necessary, it would be indicated in the ITR when the next cycle began; if nothing was indicated, it means that the student only went through one cycle of investigation. The ITR were constructed by extracting the relevant information from the transcripts, such as the timings for the start and end of each episode.



(a) Posttest Task 1 for S5



(b) Posttest Task 2 for S9

Figure 5.2 Investigation Timeline Representations (ITR)

The purpose of an ITR is to give an overall picture of the time spent in each stage or episode relative to other stages or episodes (i.e. not the exact amount of duration), and the sequence of the stages or episodes. For S5, it was observed from her ITR that she spent about two minutes to understand the task (U) and then proceeded to the Specialising / Conjecturing (S/C) stage. There were three instances of metacognitive behaviours during this S/C episode. Then there was a short transition between two stages that included an instance of metacognitive behaviour, before her first attempt at problem posing (P). She then proceeded to S/C before another transition to her second attempt at P. It was observed that most of her metacognitive behaviours occurred during the first 10 minutes of her investigation. After that, she spent a long time (about 10 minutes) in the S/C stage, with two short instances at P. It could be inferred from the ITR that she had formulated at least one conjecture because she then spent some time (about five minutes) in the Justifying / Generalising (J/G) stage. Finally, she spent the last part of her investigation checking her solution (R) with two instances of metacognitive behaviours. Since the checking stage only occurred after a problem was solved[8], this suggests that she had justified at least one conjecture. She probably did not have time to extend the task.

For S9, it was observed from his ITR in Figure 5.2 that he spent about one minute to understand the task (U) and then proceeded to the P stage, followed by the stage of Using Other Heuristics (H). After that, he went back to the P stage. There are three possibilities in this second P episode: (i) he started a new cycle by posing the second problem (it does not matter whether he had solved the first problem), (ii) he did not solve the first problem, probably because he was stuck, so he tried to think of a

[8] As explained in Section 2.2.3(g), a student can check the solution in other stages, but the checking stage (R) is only for checking and reviewing the solution after a problem is solved.

second problem but failed, thus he went back to the H stage to try to solve the first problem (i.e. he was still in the first cycle), or (iii) he solved the first problem and he tried to think of a second problem but failed. However, the last possibility is not possible since he went back to the H stage, but what did he go back for? Nevertheless, there is no way to know whether it is possibility (i) or (ii) in the ITR unless it was labelled in the ITR that he started the second cycle during the second P episode (see Fig. 5.2). Similarly for the second and third cycle. But for the fourth cycle, he started to specialise (S) after going through the Extension (E) stage. This means that he had posed a problem to extend in order to generalise, as could also be seen later when he went into the Conjecturing (C) stage and the Justifying / Generalising (J/G) stage. When he entered the S stage the second time, he was trying another example for the same extension, so he was still in the fourth cycle.

The first two data analysis tools, the IPD and the ITR, helped to give a global picture of the students' investigation pathways, which were useful to analyse the data to answer Research Question 1. But they are unable to display the types of sub-processes and outcomes within each stage, e.g. what sub-processes the students had used to understand the task correctly, or what kinds of problems the student had posed, which were needed to analyse the data to answer Research Question 2. Therefore, there was a need for a third data analysis tool.

5.3 SUMMARY TABLES OF PROCESSES AND OUTCOMES

The students' sub-processes (which will be called processes in short, unless there is confusion between the sub-processes and the main investigation processes) were gathered from their thinking-aloud protocols in the transcripts, but their investigation outcomes came from two sources: their answer scripts, and also their thinking-aloud protocols which they did not write down in their answer scripts, e.g. a student might just verbalise that he or she had observed a pattern without writing it down, then found a counter example and rejected the pattern verbally. Since mathematical investigation is not just about the final outcome or result, but also the processes such as going down a false trail and recovering from it (Jaworski, 1994), there is a need to record all the intermediate results. Therefore, the two sources complemented each other to give a more complete picture of the students' outcomes.

Unlike the IPD and ITR which describe a particular student's pathway, the Summary Table of Processes and Outcomes (TPO) displays all the students' processes and outcomes for a particular stage. For example, Table 5.1 shows a TPO used to depict the understanding processes and outcomes for Posttest Task 1 (Kaprekar). To construct the TPO, there was a need to identify the processes and outcomes in the understanding stage. In fact, these processes and outcomes had already been identified in the literature review in Section 2.2.3(a), and codes had been developed in the coding scheme to code these behaviours, e.g. TE represents the process of trying examples (the reader should refer to the Final Coding Scheme in Section 4.6.2 in order to understand the codes in Table 5.1). If a student misinterpreted the task, it was coded as Made Error or Mistake (EM). Since the processes and outcomes are different for each investigation stage, the format of the TPO would be different. Sometimes, it was also necessary to construct more than one TPO for each stage because there were too many processes and outcomes to show in one table (e.g. see the TPO for the pretest tasks in Appendix M).

Table 5.1 Understanding Processes and Outcomes for Kaprekar Task

	Processes						Outcomes	
	TE	RR	RT	HI	MU	Total**	Understood	Misinterpreted
S1	1	3			1	**4**	✓	
S2	1					**0**	✓	
S3	1	2+1+2=5*		1	0+1+1=2	**8**	✓	
S4	1	2	1	3	1	**7**	✓	
S5	1	8	1	4		**13**	✓	
S6	2	7	2	3		**12**		Recovered after 2 min
S7	1	2+0+3=5	4		0+1+1=2	**11**	✓	
S8	2	3	4			**7**		Recovered after 2 min
S9	2	9	6	5	2	**22**		Did not recover
S10	1	2+3=5			0+1=1	**6**	✓	
Total	**12**	**47**	**18**	**16**	**9**	**90**	**7**	**3 misinterpreted; 2 recovered**

* S3 engaged in RR for 2 + 1 + 2 = 5 times means that there were 3 episodes of understanding the task and RR happened 2 times in the first episode, 1 time in the second episode, and 2 times in the third episode; 3 × RR for S1 means that RR happened 3 times in the first episode.
** The 'Total' column shows the total frequency for RR, RT, HI and MU for each student.

5.4 INTER-CODER RELIABILITY TESTS FOR QUALITY OF PROBLEMS POSED AND CONJECTURES FORMULATED

There was a need to decide on the quality of certain outcomes, e.g. whether a problem posed or a conjecture formulated is trivial or non-trivial. The following shows two problems for Posttest Task 1 (Kaprekar; Type A) and two problems for Posttest Task 2 (Sausage, Type B) obtained from their task analysis in Appendix E:

- *First Problem for Kaprekar Task:* Is there any pattern in consecutive terms of the sequence? [Trivial]
- *Second Problem for Kaprekar Task:* Are there numbers that will never appear as the second or subsequent terms of any Kaprekar sequence? [Non-trivial]

- *First Problem for Sausage Task:* Find how to cut the 12 identical sausages to share them equally among the 18 people. [Trivial]
- *Second Problem for Sausage Task:* Find the least number of cuts needed to share the 12 identical sausages equally among the 18 people. [Non-trivial]

The first problem for the Kaprekar Task is considered trivial since this is a common problem for any task that involves sequences, and the pattern to be observed can just be any pattern. The second problem for the Kaprekar Task is considered non-trivial since this problem is more specific, and can only be posed after a student has studied the sequences and tried to observe a relationship among the terms across the sequences. In fact, these terms or numbers are called 'self numbers' in literature (see task analysis in Appendix E p. 498). The first problem for the Sausage Task is considered as trivial since this is a common problem that most people would naturally pose because of how the task is phrased: "I need to cut 12 identical sausages so that I can share them equally among 18 people. Investigate." Moreover, the answer to this problem can just be any cutting method. The second problem for the Sausage Task is considered as non-trivial since this problem requires the students to find a specific cutting method and to justify that the method will give the least number of cuts. In fact, there is even a general formula for the least number of cuts to share n identical sausages equally among m people. However, the reader may disagree with me on the above classification of problems as trivial or non-trivial. Thus there was a need for an inter-coder reliability test for the classification.

Table K1 in Appendix K shows a sample list of problems given to two coders to classify whether each problem is trivial or non-trivial by ticking the appropriate column. The problems were obtained from the task analysis in Appendix E. The two coders for this inter-coder reliability test are the same as those for the inter-coder reliability test for the coding scheme described in Chapter 4. Tables K2 to K5 in Appendix K shows the detailed results of the inter-coder reliability test for the four tasks while Table 5.2 below shows a summary of the number of problems for Coder 1 and Coder 2 that were classified in the same category (either trivial or non-trivial) as my classification. Since the inter-coder agreement of 87% is higher than the minimum standard of 80% for inter-coder reliability for media messages proposed by Riffe et al. (1998), it suggests that the classification of the quality of problems for the four tasks is reliable.

Table 5.2 Inter-Coder Reliability Test for Quality of Problems Posed

Task	Number of Problems	Coder 1	Coder 2	Average
Pretest Task 1	10	8	9	85%
Pretest Task 2	8	6	8	87.5%
Posttest Task 1	10	9	9	90%
Posttest Task 2	7	7	5	85.7%
Total	**35**	**30**	**31**	**87%**

Similarly, there was a need for an inter-coder reliability test for the classification of the quality of conjectures for the four tasks. Table L1 in Appendix L shows a sample list of conjectures given to the same two coders to classify whether each conjecture is trivial or non-trivial. The conjectures were obtained from the task analysis in Appendix E. Tables L2 to L5 in Appendix L shows the detailed results of the inter-coder reliability test for the four tasks while Table 5.3 below shows a summary of the number of conjectures for Coder 1 and Coder 2 that were classified in the same category (either trivial or non-trivial) as my classification. Since the inter-coder agreement of 86% is higher than the minimum standard of 80% for inter-coder reliability for media messages proposed by Riffe et al. (1998), it suggests that the classification of the quality of conjectures for the four tasks is reliable.

Table 5.3 Inter-Coder Reliability Test for Quality of Conjectures Formulated

Task	Number of Conjectures	Coder 1	Coder 2	Average
Pretest Task 1	9	7	8	83.3%
Pretest Task 2	3	2	3	83.3%
Posttest Task 1	9	7	9	88.9%
Posttest Task 2	8	7	7	87.5%
Total	**29**	**23**	**27**	**86%**

5.5 CONCLUDING REMARKS FOR THIS CHAPTER

Chapter 5 has explained the design of three data analysis tools: the Investigation Pathway Diagram (IPD), the Investigation Timeline Representation (ITR) and the Summary Table of Processes and Outcomes (TPO). The construction of the IPD and ITR was done using the students' transcripts which had been parsed into episodes using the stage codes, and the stage codes had passed the inter-coder reliability test for the coding scheme described in the previous chapter. This suggests that the IPD and ITR are reliable instruments used to display the students' actual sequences of investigation episodes so that their pathways could be analysed in order to answer Research Question 1. Since the quality of the problems and conjectures displayed in the TPO had also passed another inter-coder reliability test, it suggests that the TPO is also a reliable instrument which could be used to answer Research Question 2, and Research Question 3 qualitatively. The development of the fourth data analysis tool, the Investigation Scoring Rubric (ISR), to answer Research Question 3 quantitatively, will be discussed in Chapter 8. The next part of the thesis, Part Three, will analyse the data collected in order to answer the three research questions for the present study.

PART THREE: DATA ANALYSIS AND FINDINGS

Part Three of this thesis describes the analysis of data collected to answer the three research questions for the present study. It consists of four chapters. Chapters 6-8 will answer the three research questions while Chapter 9 will provide some implications of key findings for teaching and research.

CHAPTER 6: DATA ANALYSIS OF MATHEMATICAL INVESTIGATION PATHWAYS

In this chapter, the 20 sets of students' thinking-aloud protocols and answer scripts obtained from the posttest in the present study will be analysed to describe the actual mathematical investigation pathways of the 10 Secondary 2 students across the two types of investigative tasks in order to answer Research Question 1. The posttest tasks were chosen as they provided a wider range of processes than the pretest tasks to examine. This had already been discussed in Section 4.6 when the posttest protocols were needed to refine the coding scheme because some behaviours were not found among the pretest protocols.

6.1 THE FIRST RESEARCH QUESTION

Research Question 1 is reproduced below:

RQ1: What is the relationship between the investigation pathways of Secondary 2 students and their outcomes across the two types of investigative tasks?

Scope of Data Analysis

The actual pathways of the students present a macroscopic view of the interactions among the main processes across the two types of tasks. But the pathways only show how the students progressed from one stage to another, not what the students investigated. In mathematical problem solving, if the students cannot find a solution, they will not be able to complete the pathway. But is this true for investigation? In other words, if students are unable to find any pattern for an investigative task, is it possible for them to complete the pathway? A more general question is whether similar investigation pathways imply similar outcomes. Thus there was a need to study what the students investigated for similar pathways in order to find out whether there is any relationship between the pathways and the outcomes. This will help us to understand more fully what a complete or an incomplete pathway indicates.

The chapter will begin by choosing four students to analyse their pathways for Posttest Task 1 (Kaprekar, Type A) using their Investigation Pathway Diagrams (IPD) and Investigation Timeline Representations (ITR) developed in the previous chapter, followed by the use of their thinking-aloud protocols and answer scripts to study what they had investigated in order to compare their outcomes with their pathways. This way of analysing the data will be repeated for Posttest Task 2 (Sausage, Type B). The similarities and differences between the pathways for the two types of tasks will then be summarised.

6.2 INVESTIGATION PATHWAYS FOR TYPE A TASKS

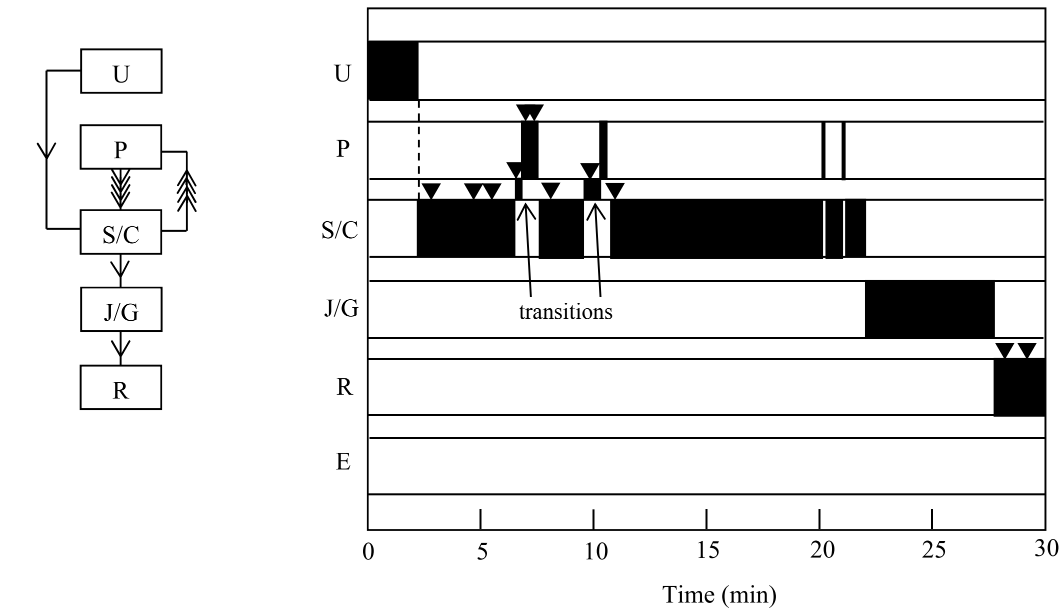
In this section, the actual investigation pathways for the 10 students when they attempted Posttest Task 1 (Kaprekar, Type A) will be examined. The task is reproduced below:

> ***Posttest Investigative Task 1: Add Sum of Digits to Number***
>
> Choose any number. Add the sum of its digits to the number itself to obtain a new number. Repeat this process for the new number and so forth. Investigate.

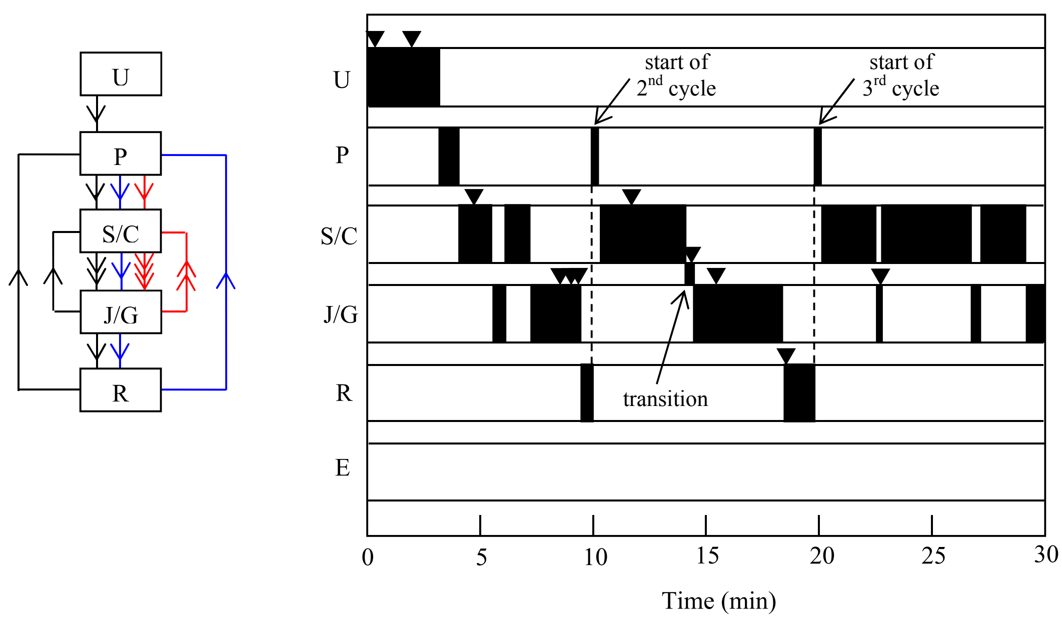
A complete investigation pathway for a Type A task is a pathway that reaches at least the Justifying / Generalising (J/G) stage. From the students' thinking-aloud protocols for the Kaprekar Task, it was found that 8 of them stopped at the Specialising / Conjecturing (S/C) stage, while 2 students (S5,S9) progressed to the J/G stage and the Checking (R) stage. None of them went into the Extension (E) stage since extension was not expected for a 30-minute investigation of a Type A task. Figure 6.1 shows the IPD and ITR for four of the 10 students. The IPD and ITR of the first two students (S5,S9) display a complete pathway while those of the other two students (S1,S10) show an incomplete pathway. The four students were chosen because their investigation outcomes were completely different, which will become clearer at the end of the data analysis in this section. Although the first two students (S5,S9) went through the complete pathway, their IPD and ITR in Figure 6.1 looked different because the first student (S5) went through only one cycle while the second student (S9) went through three complete cycles. For the other two students (S1,S10), their incomplete pathways looked quite similar except for small variations, which should be expected since no two students' investigations were exactly the same.

As explained in Section 6.1, there is a need to examine what the four students investigated in order to find out whether there is any relationship between the pathways and the outcomes, so that we can understand more fully what a complete or an incomplete investigation pathway indicates.

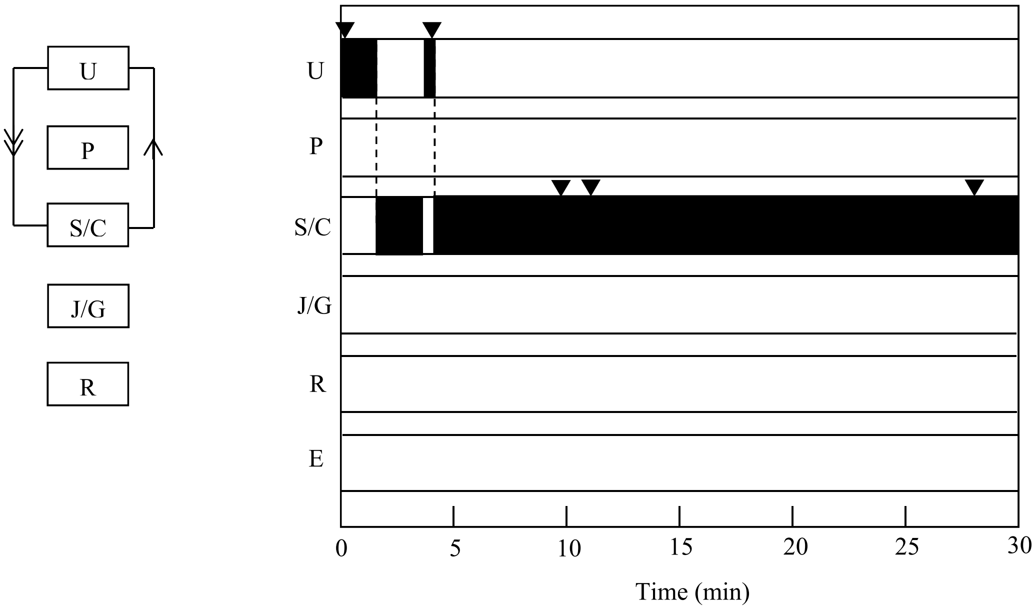
Legends: The no. of arrows in IPD indicates the no. of times the student went through the pathway.
If there was more than one pathway from one stage to another in IPD, the black pathway occurred first, followed by the blue pathway and then the red pathway.
The icon ▼ in ITR indicates metacognitive behaviour.



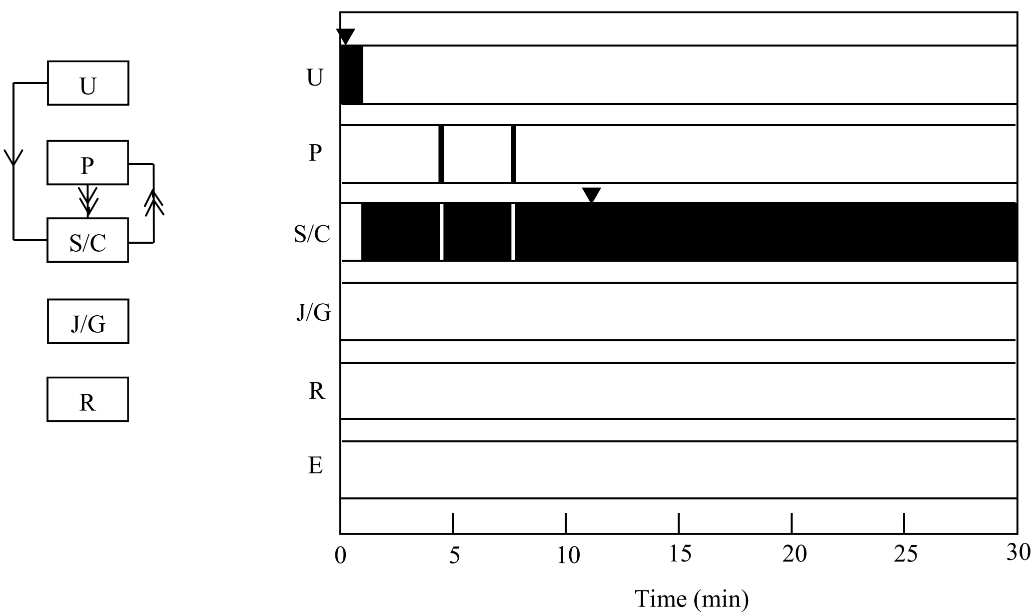
(a) Complete Pathway (S5)



(b) Complete Pathway (S9)



(c) Incomplete Pathway (S10)



(d) Incomplete Pathway (S1)

Figure 6.1 Mathematical Investigation Pathways for Type A Task

(a) Complete Pathway for Student S5

The investigation pathway in Figure 6.1(a) shows that the student (S5) started with the Understanding (U) stage and then proceeded to the Specialising / Conjecturing (S/C) stage without going through the Problem-Posing (P) stage. From her protocols and answer script, it was observed that she tried to understand the task by trying Example 1 with starting number 12 (see her answer script in Appendix J). Since 12 is divisible by 3 but not by 9, then Example 1 is a Type 1a sequence where all the terms and differences between consecutive terms are divisible by 3 but not by 9 (the reader should refer to the task analysis in Appendix E on page 499 to be familiar with the different types of sequences and patterns). She then used the same example to search for patterns without posing the general problem. Unfortunately, she made a calculation mistake that changed the Type 1a sequence into a Type 2 sequence: she wrote 51 + 6 = 56 (not divisible by 3 or 9) when it should be 57 (divisible by 3). So she concluded, "I never find the pattern yet." She proceeded to try Example 2 with starting number 23 (see Appendix J), which is a Type 2 sequence. Unfortunately, she made another mistake that changed this sequence into a Type 1a sequence: she wrote 77 + 7 = 84 (divisible by 3) when it should be 70 + 7 = 77 (not divisible by 3 or 9). Since she did not find any pattern, she read the given checklist of investigation processes (see Appendix H) and monitored her progress. This is the first transition between the S/C stage and P stage as shown in the ITR in Figure 6.1(a).

The student then wrote down the general problem: "Find patterns for these numbers." (see Appendix J) Next, she referred to the checklist again and posed Specific Problem 1 verbally: look for patterns in the sums of the digits of the terms. She went back into the S/C stage to search for patterns in Example 2 but failed. She then searched for patterns in Example 1 and observed that the sums of digits are divisible by 3 (this will be called her Pattern 1), but she rejected it as the sum of the digits of 56 (her calculation mistake described earlier) is not divisible by 3. She went into the second transition where she referred to the checklist and analysed the feasibility of some approaches in the checklist, but she concluded that she could not 'draw a diagram' or use 'guess and check'. She went back into the P stage to pose Specific Problem 2: look for patterns in the differences between consecutive sums of digits. She proceeded to the S/C stage for the third time: she drew an arc between every two consecutive sums of digits in Example 1 and wrote the difference, which was either 3 or 6 (see Appendix J). When she came to the last pair and wrote the difference 5, she sensed something amiss and said, "Eh, this is …" Then she discovered her first calculation mistake. This was a key moment that enabled her to proceed with a long period of S/C (see Fig. 6.1a) where she was able to observe various patterns as follows.

First, the student realised that her Pattern 1 (sums of digits divisible by 3) worked for Example 1 (Type 1a) but not for Example 2 (Type 2). The different types of examples and patterns were discussed in Appendix E on page 499. She then tried Example 3 (Type 1b) where the starting number 36 is divisible by 9, and observed that each term is divisible by 3 (this will be called her Pattern 2a). This pattern is different from Pattern 1 because Pattern 1 is a pattern about the sums of digits, but Pattern 2a is a pattern about the terms of the sequence. She then realised that each term in Example 3 is also divisible by 9 (this will be called her Pattern 3). She went back to Example 2, but because of another calculation mistake (described earlier) that changed Example 2 from Type 2 to Type 1a, she modified her Pattern 2a: the terms eventually become divisible by 3 (this will be called her Pattern 2b since the structure of this pattern is similar to the structure of Pattern 2a). She then tried Example 4 with starting number

47. Unfortunately, she made another calculation mistake that changed Example 4 from Type 2 to Type 1a. But in the next step, she made yet another mistake that changed Example 4 back to Type 2 again. So she rejected her Pattern 2b. Then she went on to say that all the terms in the sequence are divisible by 3 if the starting number is divisible by 3 (this will be called her Pattern 2c). As she was not sure of Pattern 2c, she continued to find more terms for Example 3. She observed that all the terms are divisible by 9 and then formulated a non-trivial conjecture (this will be called her Conjecture 1) as shown in Figure 6.2. The new numbers in Conjecture 1 refer to the terms in the sequence. In fact, Conjecture 1 was a combination of the Type 1a and Type 1b 'multiples' patterns.

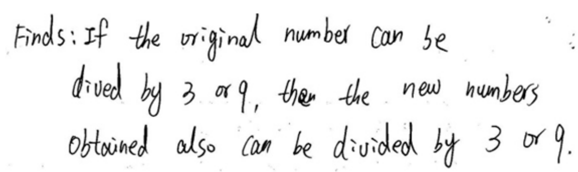


Figure 6.2 Conjecture 1 Formulated by S5 for Kaprekar Task

Next, she went into the P stage for the third time (see Fig. 6.1a) and posed Specific Problem 3: "But what if … it's … 6?" The student was able to use her Conjecture 1 as a springboard to pose a problem with an analogous result: "If the starting number is divisible by 6, does that mean all the terms in the sequence will be divisible by 6?" This is called problem posing by analogy, which was advocated by Kilpatrick (1987) and discussed in detail during the literature review in Section 2.2.3(h). The student proceeded to the S/C stage to try an example and found that the analogous result was false. She went back into the P stage for the last time and posed Specific Problem 4: "Then what if the sum of digits is 2?" It is puzzling how the sum of digits came into the picture. Perhaps because a number is divisible by 3 or 9 if and only if the sum of its digits is divisible by 3 or 9 respectively (divisibility tests for 3 and 9), she might think that this might apply to a number divisible by 2. She proceeded to the S/C stage for the last time by trying an example with starting number 11 whose sum of digits is 2 but it is not divisible by 2. She only found the next two terms in the sequence and then observed that all these number are odd. So she tried another example with starting number 20 whose sum of digits is still 2, but it is divisible by 2 and so is even. After finding the next term, she formulated Conjecture 2: "If the starting number is odd or even, then the new numbers will be odd or even respectively." This is clearly false if she was to continue her two examples, or she could have also referred to her Examples 1 and 2 for counter examples. Instead, she tried to use algebra to prove Conjecture 2. This is the first time she went into the Justifying / Generalising (J/G) stage (see Fig. 6.1a). She wrote $\overline{ab} + a + b = 11a + 2b$, but was unable to continue, so she cancelled her working.

The student tried another example with starting number 21, which is odd, and refuted Conjecture 2 when she obtained an even term 30. This is what Lakatos (1976) called 'naïve testing' and is posited in the Justifying (J) stage in the theoretical investigation model for cognitive processes described in Section 3.2.1. The student then went back to justify Conjecture 1. She re-read the conjecture and immediately provided the proof as shown in Figure 6.3. This suggests that she might have prior knowledge of the divisibility tests for 3 and 9. She then referred to the checklist which told her to check her solution. So she entered into the checking (R) stage (see Fig. 6.1a) by briefly glancing through her examples and checking that the relevant numbers were divisible by 3 or 9. She also reviewed her solution by saying that she had found the pattern for some numbers but she could not find the pattern for the rest. Then the test ended.

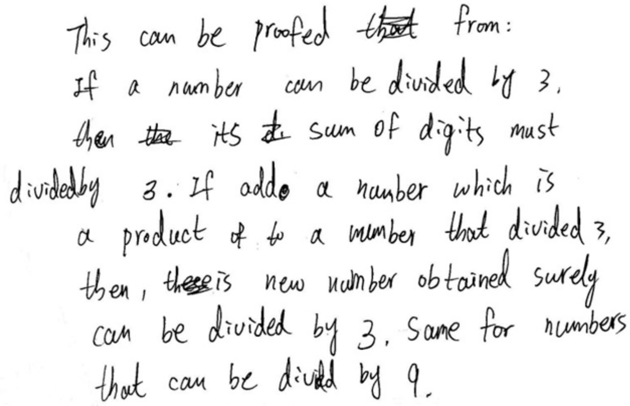

This can be proofed ~~that~~ from:
If a number can be divided by 3,
then ~~the~~ its ~~d~~ sum of digits must
dividedby 3. If adde a number which is
a product of to a number that divided 3,
then, these is new number obtained surely
can be divided by 3. Same for numbers
that can be divild by 9.

Figure 6.3 Proof of Conjecture 1 by S5 for Kaprekar Task

To summarise, S5 was able to observe the Type 1 'multiples' patterns for the Kaprekar Task and to prove that these observed patterns were the actual underlying patterns. She was also able to use her Conjecture 1 as a springboard to pose two specific problems with an analogous result. The main problem with her investigation was the wastage of precious time as a result of her various calculation mistakes that changed the patterns in her examples and confused her. Her digression to her Conjecture 2, which she rejected later, was also a distraction.

(b) Complete Pathway for Student S9

Let us analyse the investigation pathway of another student (S9) who went through three complete cycles. Does that mean he was able to find more underlying patterns for the Kaprekar Task than the previous student (S5) who only went through one complete cycle? The pathway in Figure 6.1(b) shows that the student (S9) went through the first complete cycle starting from the Understanding (U) stage to the Checking (R) stage. From his protocols and answer script, it was observed that the student tried to understand the task by trying Example 1 with starting number 123.

But he made a mistake when he treated the sum of the digits of 123, which is 6, as the new number, so he did not know what to add. He decided to monitor his own understanding by trying another example "to further understand what the task is trying to tell me" (in his own words). Thus he tried Example 2 with starting number 234 and faced the same problem when he ended up with 9 whose sum of digits is still 9. He re-read the task and highlighted the words "its digits to the number itself" in the task statement. Then he discovered his mistake that he must add the sum of the digits of the number to itself to obtain a new number. Finally, he said, "Ok, now I understand the task." However, he still misinterpreted the task as he did not repeat the process for the new number but he chose random starting numbers to repeat the process.

If we just look at the U stage of S9 in his pathway in Figure 6.1(b) and compare with that of the previous student (S5) in Figure 6.1(a), we will not be able to see how S9 struggled to understand the task and how he still misinterpreted the task at the end, despite monitoring his own understanding, which was a metacognitive behaviour. Moreover, S9 progressed to the Problem-Posing (P) stage by posing the general problem of searching for any pattern, which was prescribed as the next logical step in the theoretical investigation model for cognitive processes described in Section 3.2.1, unlike S5 who skipped the P stage. Yet it was S9 who misinterpreted the task and went off in a different direction that his investigation pathway was unable to show.

The student (S9) then went into the Specialising / Conjecturing (S/C) stage by trying Example 3, which consisted of different starting 2-digit numbers without repeating the process for the new number, and formulating his Conjecture 1 as shown in Figure 6.4. After that, he proceeded to the Justifying / Generalising (J/G) stage when he asked, "So now, how do you prove this?" But he tried to test Conjecture 1 using naïve testing first. His Example 4 with starting number 22 was a counter example, so he rejected Conjecture 1 as shown in the same figure.

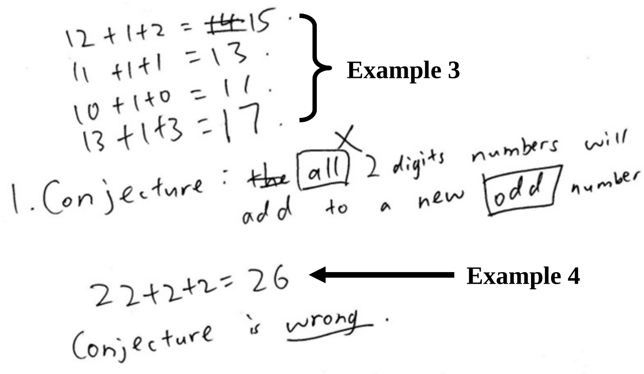


Figure 6.4 Examples and Conjecture 1 by S9 for Kaprekar Task

The student then went back into the S/C stage to modify Conjecture 1 by recognising that his conjecture only applied under some special circumstances. The following shows his protocols:

> "So since my conjecture of all [box up the word 'all' in Conjecture 1 and put a cross beside it] is wrong, so let us see when does this apply to? … Ok, conjecture, I think so it applies to … any two-digit numbers that have its tens, that have its tens place an odd one ... Yah, I think so …" [S9; Kaprekar Task]

The protocols show that the student did not just refute a conjecture when he found a counter example, but he was able to discern the conditions for which the conjecture was false and so he modified the conjecture instead. This is called the reformulation of a conjecture as a result of a local counter example by modifying the conjecture, in accordance to Lakatos' mathematical discovery model described in Section 2.2.2(c), which is different from the reformulation of a conjecture as a result of a global counter example where the conjecture has to be refuted and a new conjecture has to be formulated. The end result was Conjecture 2 as shown in Figure 6.5: “All 2-digit numbers, which [*sic*] tens place is odd, will have a new number that is odd.” He then proceeded to the J/G stage to prove this conjecture by reasoning about the sum of odd and even numbers as shown in the same figure. This is called justifying by using a non-proof argument involving the underlying structure (Mason et al., 1985).

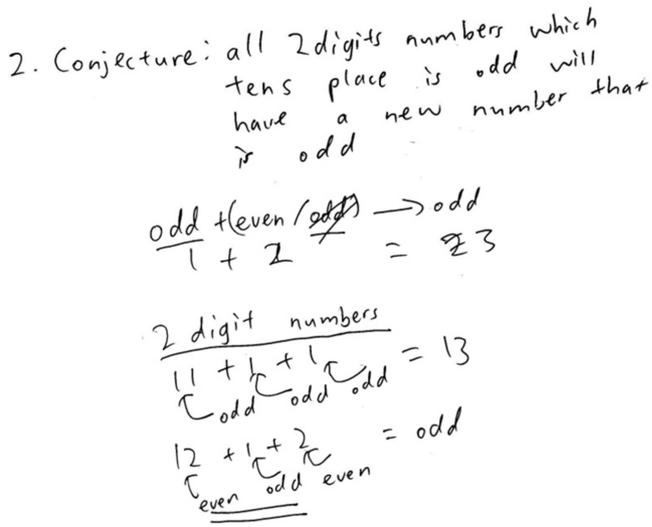


Figure 6.5 Conjecture 2 and Proof by S9 for Kaprekar Task

Next, the student went into the R stage by reviewing his solution and then concluded:

> “Ok, that seems not the … real task is wanting me to find. [Turn back to p. 1] So … the task is wanting me to find … the pattern. Ok … so I think that my second conjecture is a little bit off topic.” [S9; Kaprekar Task]

This was the end of the first cycle for this student (S9) who proved a conjecture about a trivial pattern that was not even relevant to the original task because he had misinterpreted the task by not repeating the process for the new number. It was unlike the complete cycle of the previous student (S5) who proved a conjecture about a non-trivial pattern relevant to the original task. Thus it is possible for a student to go through a complete pathway for investigation by formulating and justifying a trivial conjecture. In other words, a complete investigation pathway for a Type A task does not indicate that the student has discovered a non-trivial pattern for the original task.

The student (S9) began the second cycle by going back to the P stage (see Fig. 6.1b) to pose Specific Problem 1: find a pattern for two-digit numbers. This time, he specialised by making a systematic list of consecutive two-digit starting numbers from 11 to 15 as shown in Figure 6.6, and he observed that the difference between consecutive new numbers is +2. He skipped the starting numbers 16 to 20, and continued from 21 to 22. But he sensed something amiss when he noticed that there was only a small increase from the new number 21 to the new number 24 (coded as metacognitive awareness as described in Section 4.6.1). So he filled in the gap by starting from 20 backwards to 16. Then he observed a big difference of −7 from the new number 29 to the new number 22. This became his Conjecture 3 which he had difficulty phrasing in words (see Fig. 6.6).

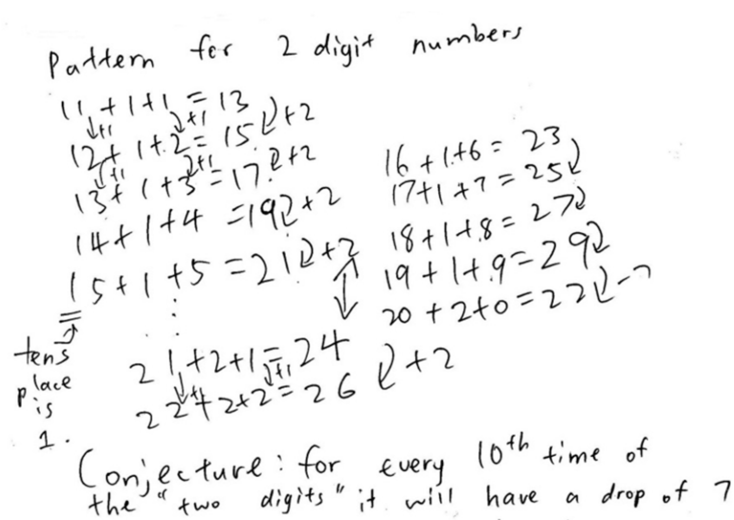


Figure 6.6 Example 5 and Conjecture 3 by S9 for Kaprekar Task

The student continued the same example from the starting number 23 to the starting number 30, and observed the same jump of −7 from the new number 40 to the new number 33 (see Fig. 6.7 which shows only the last part of his working). He proceeded to prove Conjecture 3 by using the underlying structure to explain why there is an increase of 2 for the new number when the starting number goes from 11 to 19 (see Fig. 6.6): there is an increase of 1 in the starting number, and an increase of 1 in the ones digit in the sum of its digits, thus resulting in a total increase of 2 for the new number. Similarly, he used the underlying structure to explain why there is a decrease of 7 for the new number when the starting number goes from 29 to 30 (see Fig. 6.7): there is an increase of 1 in the starting number, and an increase of 1 in the tens digit, but a decrease of 9 in the ones digit in the sum of its digits, thus resulting in a total decrease of 7 for the new number. As the reader should have realised by now, the explanation in words is really too long compared with the visual explanation using arrows in Figures 6.6 and 6.7, so the student had difficulty phrasing his proof in words as shown at the bottom of Figure 6.7.

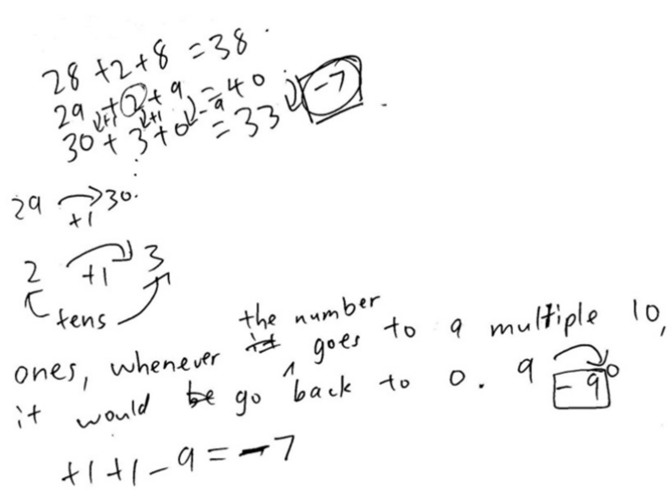


Figure 6.7 Proof of Conjecture 3 by S9 for Kaprekar Task

The student reviewed his solution and realised that he had performed the process only once for a starting number. Then he went into the third cycle (see Fig. 6.1b) by posing Specific Problem 2, “So what about 2 times for 2-digit numbers?” What he meant was that he wanted to perform the process twice for two-digit starting numbers in order to find some patterns. He then specialised by making a systematic list as shown in Figure 6.8. He observed that there is an increase of 4 in consecutive new numbers, except when the starting number goes from 19 to 20, where there is a decrease of 14 from the new number 40 to the new number 26. But he did not notice a serious mistake: when the starting number goes from 14 to 15, which he did not try, there is already a decrease of 5 from the new number 29 to the new number 24. Nevertheless, he formulated Conjecture 4, which was at a higher level of generalisation, because he tried to apply the pattern to the 2-digit numbers where the process was performed *n* times. As usual, he had difficulty phrasing his conjecture as shown at the bottom of the same figure. He then tried to test his conjecture using naïve testing by performing the process for 3 times when the test ended.

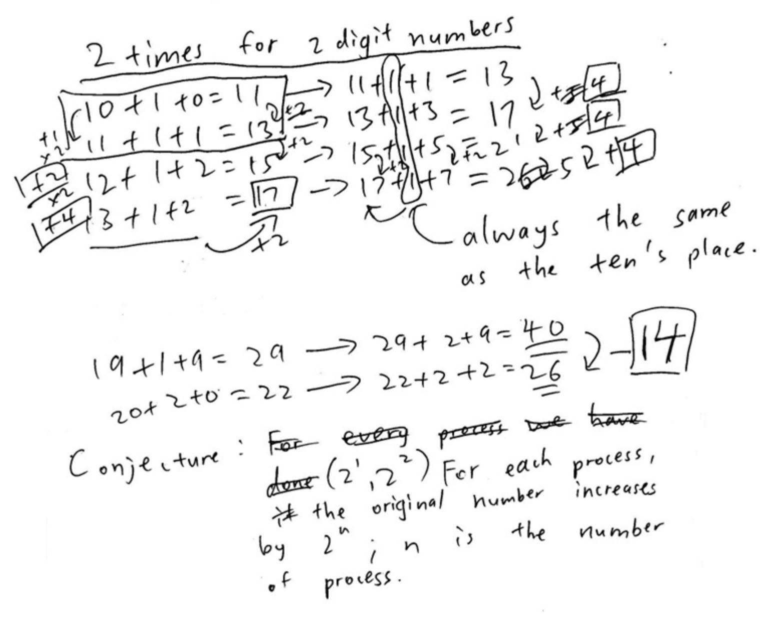


Figure 6.8 Example 6 and Conjecture 4 by S9 for Kaprekar Task

To summarise, S9 was able to observe the underlying patterns and explain why the patterns work. But he misinterpreted the task by not repeating the process for the new number, so the patterns that he had observed were not the patterns for the original task. Ernest (1991) described investigation as the exploration of an unknown land without any fixed destination. This means that the students could investigate anything on the unknown land. But what happens if the students misinterpreted the task? Using Ernest's metaphor, this would mean that the students were lost and somehow got off the unknown land. Now, what happens if the students then went across the sea and discovered a beautiful island? Are we going to say that their discovery is invalid? From another perspective, going off the unknown land means extending the task by changing the given, which will usually create a new task with different patterns for Type A tasks, as explained in Section 2.2.3(h). But the issue was that S9 did not extend the task: he misinterpreted it. I understand that different people might have different opinions about whether his discovery should be considered valid or invalid. But what was evident was that he had demonstrated the ability to find and explain the underlying patterns of the misinterpreted task, even to the extent of formulating a conjecture at a higher level of generalisation.

Although the two students (S5,S9) went through the complete investigation pathway, their investigation outcomes were totally different. In other words, a complete investigation pathway does not indicate that the student has found the underlying patterns of the original investigative task.

(c) Incomplete Pathway for Student S10

We will now analyse an incomplete investigation pathway of a student (S10). The pathway in Figure 6.1(c) shows that the student started with the Understanding (U) stage and then proceeded to the Specialising / Conjecturing (S/C) stage without going through the Problem-Posing (P) stage. From her protocols and answer script, it was observed that the student tried only one example with starting number 23 (Type 2 sequence) to understand the task. Figure 6.9 shows the first part of her working.

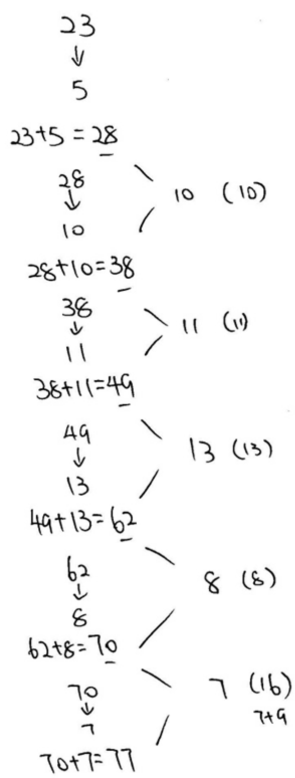


Figure 6.9 First Part of Only Example of S10 for Kaprekar Task

The student then proceeded to the S/C stage by using the same example to search for patterns. When she could not find any pattern after four minutes into the investigation, she went back to the U stage (see Fig. 6.1c) by re-reading the task and checking that she had performed the operations as stated in the task statement correctly. This is the metacognitive process of monitoring her own understanding. The student realised the importance of this process right at the start of the investigation to ensure that she did not go down a false trail due to any misinterpretation or incorrect carrying out of the operations. She then spent the remaining 25 minutes of her investigation in the S/C stage. When she was searching for patterns, she missed out the difference 5 between the starting number 23 and the second term 28. Instead, she started with the difference 10 between the second term 28 and the third term 38 (see Fig. 6.9). It does not matter as the pattern will not be affected. At first, she observed a 10-11-13-8-7-14 repeating pattern in the differences between consecutive terms (this will be called her Pattern 1), but she soon found exceptions such as 2, 4, 5, 16, 17 and 19. About 14 minutes into the test, she made a crucial observation about the relationship between the 'exceptions' and the numbers in the 10-11-13-8-7-14 pattern:

$$
\begin{aligned}
19 &= 10 + 9, \\
11 &= 2 + 9, \\
13 &= 4 + 9, \\
17 &= 8 + 9, \\
16 &= 7 + 9, \\
14 &= 5 + 9.
\end{aligned}
$$

In other words, the basic pattern is actually the 10-2-4-8-7-5 repeating pattern[9], and the exceptions are actually 19, 11, 13, 17, 16 and 14, which are different from the corresponding numbers in the basic pattern by 9 (this will be called her Pattern 2).

[9] Actually, the basic pattern is the 1-2-4-8-7-5 repeating pattern where 10 = 1 + 9, but the student had not obtained the number 1 before, so she was unable to make such an observation.

This is another way of looking at the Type 2 'digital roots' pattern[10] (the reader should refer to the task analysis in Appendix E on page 500 to be familiar with the different patterns for this task). However, the student did not see this as the underlying pattern. So she continued finding more terms to search for patterns. About 27 minutes into the investigation, she decided to organise her working in another way as shown in Figure 6.10, but she started the list from 7 instead of 10, which will not affect the pattern anyway. Although the consecutive differences were shown listed vertically, i.e. 7, 5 + 9, 10, 2, 4, 8, etc., she actually wrote the list row by row, i.e. she wrote 7, 7 + 9, 7 in the first row, followed by the second row, etc.

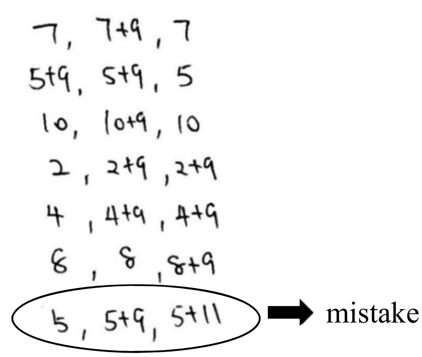


Figure 6.10 Discovery of Most Complicated Pattern for Kaprekar Task by S10

The following shows the student's protocols as she filled in the numbers row by row in Figure 6.10. Notice that she said "oh" after writing the three numbers in the first row, which suggests that she might have observed something from organising her working in the manner shown in the same figure.

[10] Provided the differences between consecutive terms are less than 20. If the differences are more than 20, e.g. 22 = 4 + 9 + 9, the pattern will be different from, e.g. 13 = 4 + 9, but the digital roots of both 13 and 22 are still 4.

> "So … is there any pattern? Ok … 7 … then is … 7 + 9 … 7 … 1, 2, 3 … [Start writing pattern row by row] 7, 7 + 9, 7 … oh … is … 5 + 9, 14 … 5 + 9, 5 … 10, 10 + 9, 10 … 2, 2 + 9, 2 + 9 … 4, 4 + 9, 4 + 9 … 8, 8 —" [S10; Kaprekar Task]

Unfortunately, the invigilator interrupted her at this juncture to inform her that she had one minute left for the test. She continued to write 8 + 9 in the second last row, paused for 4 seconds, and wrote the last line wrongly, when the pattern should have gone back to the first row. In particular, the last number 5 + 11 = 16 messed up the entire pattern, which caused her to say, "16 … oh, s***!" Then she checked her working briefly and asked, "What is wrong?" before the test ended. In other words, the student had actually observed the Type 2 'digital roots' pattern, but she made a serious mistake while writing down her conjecture when she was interrupted.

To summarise, S10 was able to observe the much more complicated Type 2 'digital roots' pattern. But her investigation pathway was incomplete, unlike the complete pathway of S5 who only discovered the less complicated Type 1 'multiples' pattern. Thus a complete investigation pathway just indicates that the student has progressed to the Justifying / Generalising (J/G) stage. In other words, it is possible for a student with an incomplete investigation pathway to discover a much more complicated pattern than another student with a complete investigation pathway.

(d) Incomplete Pathway for Student S1

We will now analyse an incomplete pathway from another student (S1). The pathway in Figure 6.1(d) shows that the student started with the Understanding (U) stage and then proceeded to the Specialising / Conjecturing (S/C) stage without going through the Problem-Posing (P) stage. From his protocols and answer script, it was observed that he tried Example 1 with starting number 21 (Type 1a sequence) to understand the task. Figure 6.11 shows the first part of his working.

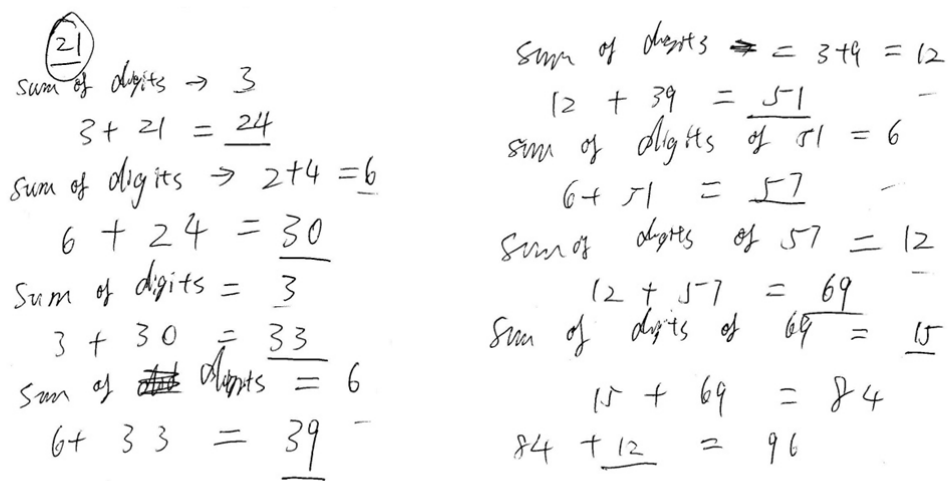


Figure 6.11 First Part of Example 1 of S1 for Kaprekar Task

When the student obtained the new number 30, he immediately proceeded to the S/C stage to search for patterns, but was unable to find any. He then specialised some more using the same example. After reaching the new number 96, he went back to the P stage to pose the general problem: "What is the pattern?" He proceeded to the S/C stage again to search for patterns, but was unable to find any. He entered the P stage the second time by saying that he was trying to find a formula for the general term of the sequence by starting from the first term in this manner:

21, 21 + 3, 21 + 3 + 6, 21 + 3 + 6 + 3, … (see his working in Fig. 6.12 later).

This means that he needed to find a pattern for the sums of digits: 3, 6, 3, … He then went into the S/C stage for the rest of the investigation. He soon observed that the consecutive sums of digits repeat in this manner: 3, 6, 3, 6, 12, 6, 12, 15, 12, 15. He specialised some more using the same example and found that the pattern repeated for the second time. He continued finding more terms, but then he discovered some counter examples for the sums of digits as shown in bold below:

3, 6, 3, 6, 12, 6, 12, 15, 12, 15,
3, 6, 3, 6, 12, 6, 12, 15, 12, 15,
3, 6, 3, 6, 12, 6, 12, 15, 12, 15,
12, 6, **12**, 6, 12, **15**.

He then checked his working and realised that "the sequence got some problem" (in his own words), so he rejected the pattern. He was not able to observe the Type 1a 'multiples' pattern: the sums of digits are divisible by 3 but not by 9 if the starting number is divisible by 3 but not by 9. In fact, there is also a Type 1a 'digital roots' pattern: the digital roots of the differences between consecutive terms of a Type 1a sequence will alternate between 3 and 6 (see the task analysis in Appendix E on page 500), although the students were not expected to observe the 'digital roots' pattern. Moreover, there is a simple argument that the differences between consecutive terms will never repeat. Notice first that this is an increasing sequence because the next term is the sum of the previous term and its digits. This means that the term will get bigger and bigger until it reaches a number with so many digits that the sum of its digits will be more than 15. In fact, when the pattern started to fail in the above example, the numbers had become bigger until the 3's were replaced by 12's. Of course, it is still possible for 3 to appear again later, e.g. if the bigger number has many zeros. Although the student realised that the sums of digits did not repeat, he still attempted to describe the terms of the sequence in terms of the first term as shown in Figure 6.12. However, he concluded verbally that the sequence did not repeat the third time.

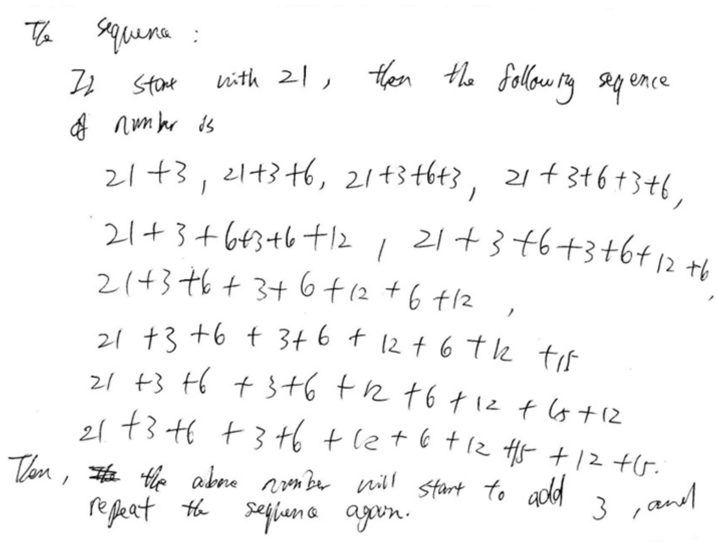

The sequence :
It start with 21, then the following sequence
of number is
21+3, 21+3+6, 21+3+6+3, 21+3+6+3+6,
21+3+6+3+6+12, 21+3+6+3+6+12+6,
21+3+6+3+6+12+6+12,
21+3+6+3+6+12+6+12+15
21+3+6+3+6+12+6+12+6+12
21+3+6+3+6+12+6+12+15+12+15.
Then, the above number will start to add 3, and
repeat the sequence again.

Figure 6.12 Conclusion of S1 for Kaprekar Task

The student then tried Example 2 with starting number 32 (Type 2), but he did not go far because he only had 4 minutes left. Again he wrongly believed that the sums of digits would repeat, but he soon found a counter example. He was puzzled and he asked, “How come it’s like that?” Then the test ended. To summarise, S1 was able to see that there were two types of sequences, but he was not able to describe their underlying patterns. He kept thinking that the sums of digits would repeat, but he was not able to use the simple argument described above to reason that the sums of digits would never repeat in that manner.

(e) Incomplete Pathways for the Remaining Six Students

The remaining 6 students (S2-S4,S6-S8) did not complete their investigation pathway, just like S1 and S10 described earlier in this section. None of them progressed to the Justifying / Generalising (J/G) stage as they were unable to observe the underlying patterns. Table 6.1 shows a summary of their investigation pathways and outcomes

.

Table 6.1 Summary of Pathways and Outcomes for Type A Tasks

	Non-trivial Outcomes	Trivial Outcomes / Misinterpreted Task
Complete Pathway	**S5**	**S9**
Incomplete Pathway	**S10**	**S1**, S2-S4, S6-S8

* Students in bold are those whose transcripts have been described in detail earlier

One would expect some variations in their investigation because no two students' outcomes would be exactly the same. For example, S3 and S8 had actually observed the Type 1a 'multiples' pattern (the reader should refer to the task analysis in Appendix E to be familiar with the different types of patterns and sequences), unlike all the other students except for S5 who completed her pathway; but unlike S5, the two students (S3,S8) did not realise that this was the underlying pattern and so they kept trying to find how the sums of digits (or the differences between consecutive terms) would repeat but failed.

Just like S10, S4 and S6 were the other two students who tried to find a pattern for a Type 2 sequence, but unlike S10, they were unable to observe the 'digital roots' pattern: S4 made a mistake towards the end that changed the Type 2 sequence into a Type 1b sequence, while S6 made an unusual observation: she thought that the pattern was that the difference between two consecutive terms is 10, followed by 4 terms, and then the pattern repeats, but it was incorrect (see Figure 7.4 on page 298 in Section 7.2.4 to understand what her pattern means).

The last two students, S2 and S7, tried to find a pattern in all the three types of sequences (Types 1a, 1b and 2), but they were also not successful: S2 tried to fit the same pattern in all the three types of sequences, but in the end, he realised that each type of sequences has its own pattern although he could not find it; S7 exhibited the most instances of metacognitive behaviours at 14, but these behaviours were not effective in helping her observe any pattern. In fact, S7 engaged in metacognitive behaviours more often than S5, who completed the investigation pathway but engaged in only 11 instances of metacognitive behaviours.

(f) Summary of Mathematical Investigation Pathways for Type A Tasks

Eight out of the 10 students did not complete the investigation pathway for the Kaprekar Task (Type A): they stopped at the Specialising / Conjecturing (S/C) stage because they were unable to observe the underlying patterns. Only 2 students (S5,S9) completed the investigation pathway: they reached the Justifying / Generalising (J/G) stage and correctly proved a non-trivial conjecture each. However, it was discovered that the pathways of the students did not tell much about their outcomes. Two students could have similar pathways but they could have investigated totally different things. For example, both S5 and S9 went through the complete investigation pathway, but S9 misinterpreted the task and observed patterns that are different from those for the original task, while S5 understood the task correctly and discovered the Type 1 'multiples' pattern. Similarly, the inability to complete the investigation pathway does not mean that the students were poor in their outcomes. For example, S10 was stuck in the S/C stage for most of her investigation but she discovered the Type 2 'digital roots' pattern, which is a lot more complicated than the Type 1 'multiples' pattern found by S5 who had completed her pathway.

6.3 INVESTIGATION PATHWAYS FOR TYPE B TASKS

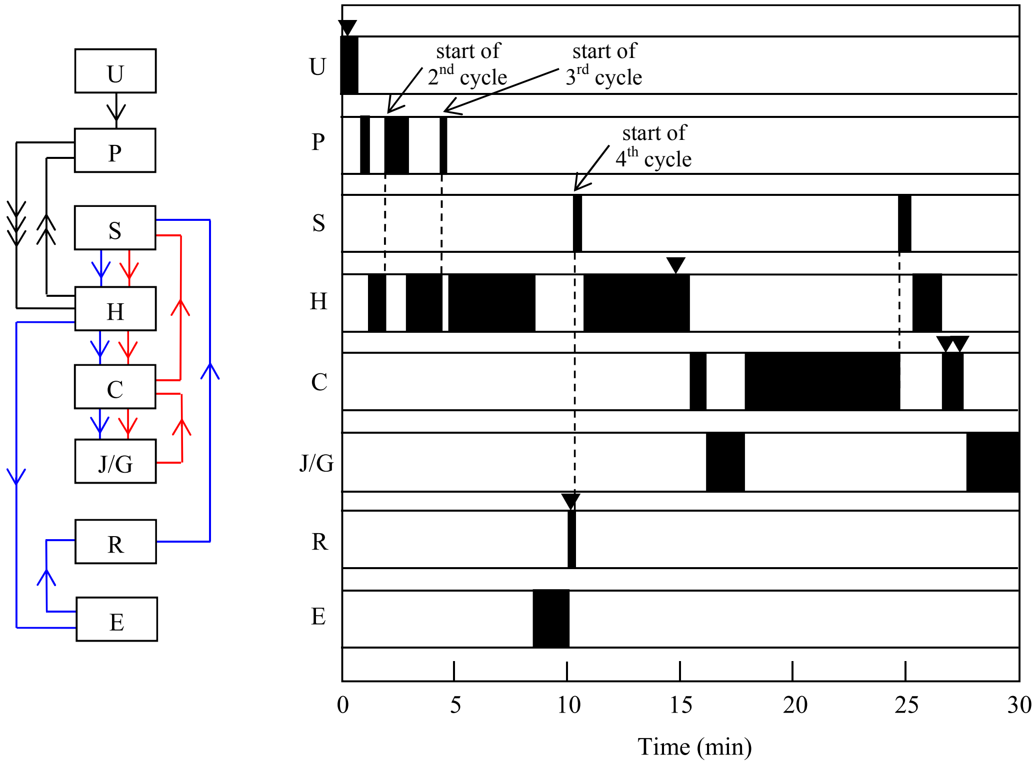
In this section, the actual investigation pathways for the 10 students when they attempted Posttest Task 2 (Sausage, Type B) will be examined. The task is reproduced below:

> ***Posttest Investigative Task 2: Sausages***
>
> I need to cut 12 identical sausages so that I can share them equally among 18 people. Investigate.

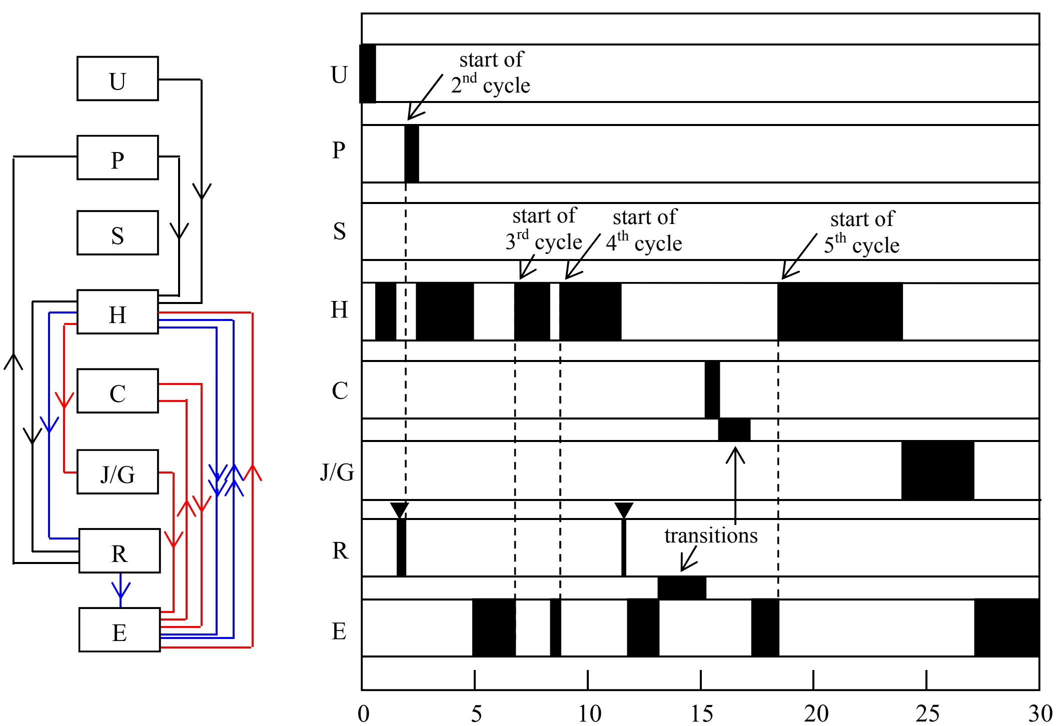
Since students are expected to extend a Type B task in order to generalise, a complete investigation pathway for a Type B task is a pathway that reaches at least the Justifying / Generalising (J/G) stage. This is similar to the complete pathway for a Type A task, but the difference is that a student can enter the Justifying (J) stage for a Type B task without generalising (G) because he or she can formulate a conjecture that is not a general result by using other heuristics and then justify it, as posited in the theoretical investigation model of cognitive processes described in Section 3.2.1.

From the students' thinking-aloud protocols for the Sausage Task, it was found that all of them extended the task, but only 5 students (S2-S4,S8,S9) entered the J/G stage. Figure 6.13 shows the Investigation Pathway Diagrams (IPD) and Investigation Timeline Representations (ITR) for four of the 10 students. The IPD and ITR of the first two students (S2,S9) display a complete pathway while those of the other two students (S1,S7) show an incomplete pathway. The four students were chosen because their investigation outcomes were completely different, which will become clearer at the end of the data analysis in this section.

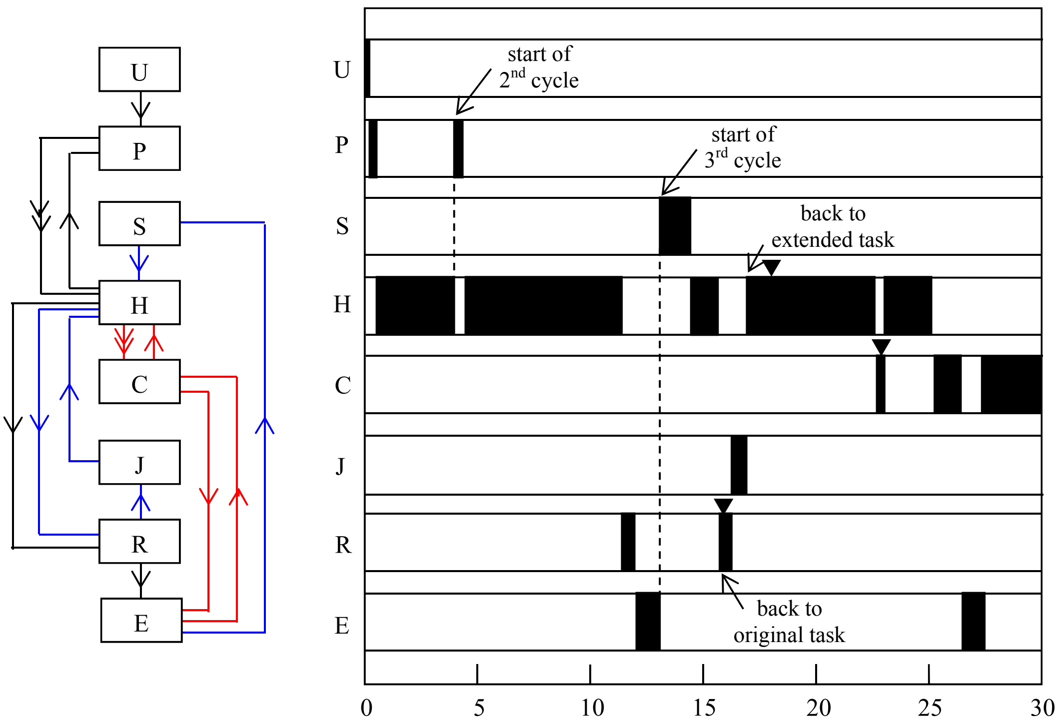
Legends: The no. of arrows in IPD indicates the no. of times the student went through the pathway.
If there was more than one pathway from one stage to another in IPD, the black pathway occurred first, followed by the blue pathway and then the red pathway.
The icon ▼ in ITR indicates metacognitive behaviour.



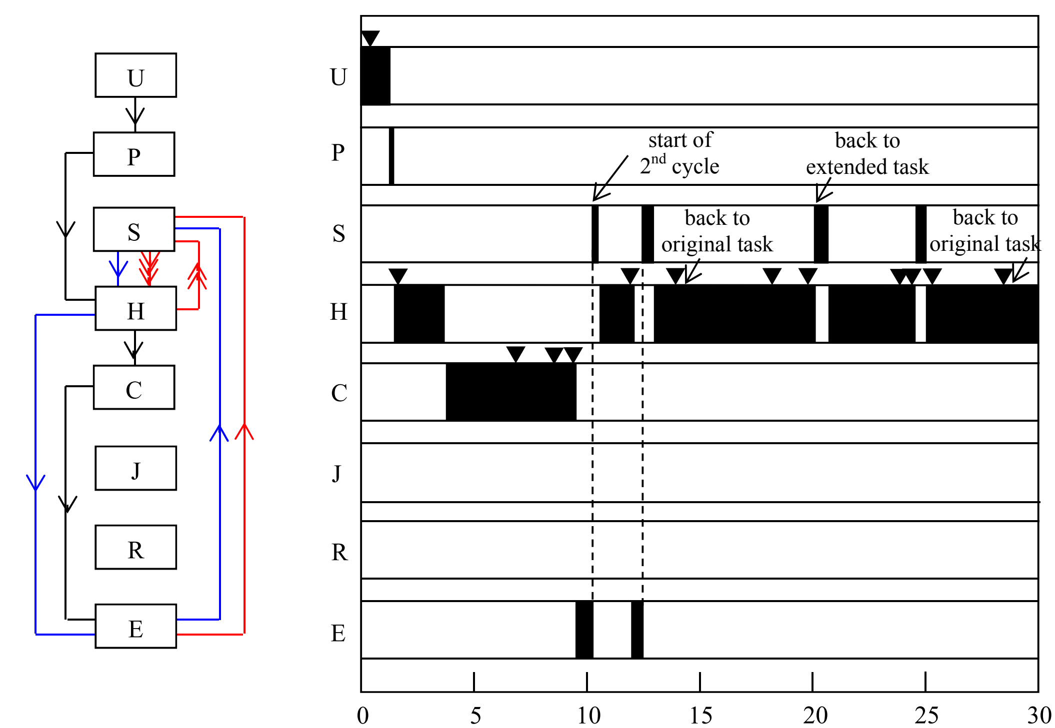
(a) Complete Pathway (S9)



(b) Complete Pathway (S2)



(c) Incomplete Pathway (S1)



(d) Incomplete Pathway (S7)

Figure 6.13 Mathematical Investigation Pathways for Type B Task

(a) Complete Pathway for Student S9

The investigation pathway in Figure 6.13(a) shows that the student (S9) went through the first cycle from the Understanding (U) stage to the stage of Using Other Heuristics (H). From his protocols and answer script, it was observed that he posed Problem 1 after understanding the task: "How do I share 12 sausages with 18 people?" But his idea of how to share was *not* how to cut since he solved the problem by finding the fraction of sausage each person will receive, which is 12/18 = 2/3. This was the end of the first cycle as he had finished solving the problem that he had posed. He went into the second cycle starting from the Problem-Posing (P) stage to the H stage (see Fig. 6.13a). He wanted to find out why the method for solving Problem 1 worked, so he posed[11] Problem 2: "How do I calculate the amount of sausages per person?" Then he used the formula 'amount of sausages divided by amount of people' to obtain 12/18 = 2/3. This was the end of the second cycle.

The student went into the third cycle starting from the P stage, but this time he ended the cycle at the Extension (E) stage. He posed Problem 3 verbally: "How many times did I need to cut the sausages?" He then discussed two methods: cut each sausage into 3 equal parts (called Cutting Method A or the Usual Method in the task analysis of the Sausage Task in Appendix E), and cut each sausage at the 2/3-mark to divide the sausage into a 2/3 part and a 1/3 part (called Cutting Method B or the Shortest Method). In the end, he found that the least number of cuts required to share the 12 sausages equally among the 18 people is 12, but he did not realise that this is only a conjecture that needs to be proven. Figure 6.14 shows his solution for Problem 3.

---

[11] This was called Problem 2 because he said so himself, and the start of the second cycle was based on the start of his Problem 2. At a later stage, he realised that Problem 2 was actually not a problem but an explanation of why the method of solution for Problem 1 worked.

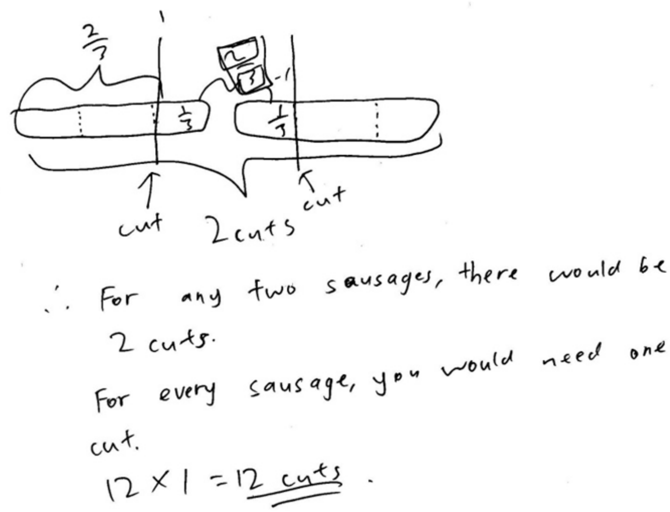


Figure 6.14 Cutting Method A for Example 1 of S9 for Sausage Task

Although the student did not explicitly pose the intended problem of finding the least number of cuts needed to share 12 sausages equally among 18 people, he ended up doing so when he solved Problem 3 above. He then decided to extend the task to find the least number of cuts for different numbers of sausages. After that, he reviewed his previous solutions and realised that his Problem 2 was actually not a problem, but an explanation of why the method of solution for Problem 1 worked. So he actually reversed the order of the Checking (R) stage and the E stage in the theoretical investigation model for cognitive processes by extending the task before checking his previous solutions. This was the end of the third cycle.

The student then went into a fourth cycle (see Fig. 6.13a). Unlike the previous cycles where he only used other heuristics (H), in this cycle, the student also engaged in specialising (S), conjecturing (C) and justifying (J) because he extended the task in order to generalise (G). He began with Example 2 (specialising): share 10 sausages equally among 18 people (the original task, where 12 sausages were shared equally among 18 people, was counted as Example 1). But in order to find the least number of cuts for Example 2, he needed to use other heuristics (H) again. This would lead to the formulation of a conjecture (C) later. Figure 6.15 shows his working for Example 2.

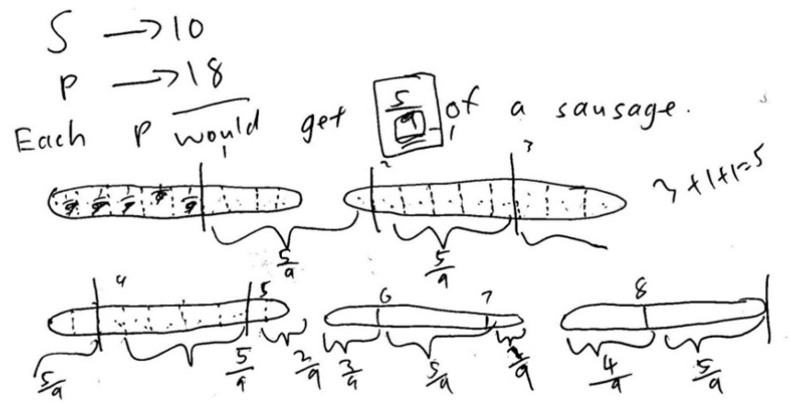


Figure 6.15 Example 2 of S9 for Sausage Task

In Example 2, the student first found that the fraction of sausage each person will receive is 5/9. He then drew 5 sausages and cut them accordingly. After cutting all the 5 sausages, he was able to divide them equally with no remainder. So he forgot that there were actually 10 sausages and 18 people, not 5 sausages and 9 people. Thus he wrongly believed that there were only 8 cuts when it should be 16. He tried to find a pattern by looking at Example 1 shown in Figure 6.14 and then he observed a pattern based on the 2 sausages and 3 people in the figure: the numerator for the fraction of sausage each person will receive is the number of sausages, while the denominator minus 1 is the least number of cuts. His protocols from the moment he started to search for patterns in Example 1 in Figure 6.14 until he made this discovery are given below, but there was no indication of how he ever thought of linking the denominator of the fraction to the least number of cuts just by subtracting 1.

> "Why is it 8 cuts? Um … from what you can see … if you actually notice … the number of sausages … for the number of cuts, it is actually [point pen at 2/3 in first sausage in Example 1 as shown in Fig. 6.14], see that there is, this is the fraction, right [box up 5/9 in Example 2 as shown in Fig. 6.15]? Fraction is actually … this is also a fraction [box up 2/3 in Example 1] so [clear his throat] in fact, the number of sausages is the top one, ah, which is quite true. The number of cuts [draw box around 3 for 2/3 in Example 1] is the denominator minus 1 [write beside denominator 3: – 1]. This is also [draw box around 9 for 5/9 in Example 2 and write beside denominator 9: – 1]. It is the same, right?" [S9; Sausage Task]

As shown in the above protocols, he made the same link to Example 2 by subtracting 1 from the denominator of 5/9 to give 8 cuts. After that, he tried to prove this conjecture by using some reasoning for about two minutes but failed. Then he decided to use algebra: x sausages and y people. He said that he wanted to derive a formula for the least number of cuts in terms of x and y by using the fraction of sausage each person will receive, which is x / y. In other words, he had changed his extension from finding the least number of cuts for different numbers of sausages to finding the least number of cuts for different numbers of people as well. Then he discovered his mistake in Example 2: the least number of cuts is 16 since there are 10 sausages and 18 people, not 5 sausages and 9 people. He struggled to express the least number of cuts in terms of x and y for about two minutes and then he realised the main problem:

> "That is the problem. You cannot use x … number of sausages … when it is in reduced form. How do I write it? How do I write it while it is in reduced form?" [S9; Sausage Task]

What he meant is when the fraction of sausage each person will receive is reduced to the lowest terms, the numerator might no longer be x, e.g. $x / y = 10/18 = 5/9$ in Example 2, where the numerator is now 5 and not equal to x, which is 10. This means that he had difficulty finding the least number of cuts in terms of x and y since he needed the fraction in the lowest terms. After struggling for another 1.5 minutes, the following shows his protocols leading to the discovery of the formula for the least number of cuts needed to share x sausages equally among y people: y – HCF (x,y).

> "How do I derive a formula for this? … The formula, ah … hard leh because it is in a fraction, and fractions sometimes it can be improper. Some fractions can be bigger and they need to be reduced. Okay. When I reduce the number of cuts …
> Eh, wait! I think I saw something, ah. [Flip to p. 2] Yah! [Underline: 18] 18 people and [turn back to p. 3] there are 16 cuts. So how should I say about this? There are 18 people and 16 cuts … Um … how should I say about this? Eh … hey, hey, hey, hey, hey, hey … hey, I found out something! … The HCF [write: HCF] which is the highest common factor [continue writing] for x & y [stop writing]. Then we … is the HCF for … this one [draw brackets around: HCF for x & y]. Wait, I found out something! There are a total of 18 people, right? But there are 16 cuts [write: 18 ppl, 16 cuts] is minus 2 [draw arrow to link 18 and 16 and write: – 2]. It is exactly the [start writing] HCF for x & y [stop writing].
> Okay, so we can find out something, right? So the formula is just a conjecture ah. The [start writing] Formula is [stop writing] x, eh, no. If I have not remembered is y [continue writing] y – (HCF of x & y) [stop writing]." [S9; Sausage Task]

The student first tried to link 16 cuts and 18 people by finding the HCF of these two numbers, which is 2. Then he realised that $18 - 2 = 16$, just like what he did for $9 - 1 = 8$ earlier. In this way, he did not use the fraction in the lowest terms but the original fraction x / y. This means that he could relate the least number of cuts to x and y. But it was still not clear from his protocols how he ended up with $18 - 2 = 16$. This is a strange way of linking HCF(10,18) = 2 to 16 and 18 by subtraction, as the arithmetic operations for this kind of sharing problems are often multiplication and division, not addition or subtraction. Although this idea might follow from what he did for $9 - 1 = 8$, the question remains as to why he could think of doing $9 - 1 = 8$ in the first place.

Unlike his solution to Problem 3 where he did not realise that it was only a conjecture, this time round he recognised that "the formula is just a conjecture" as shown in the protocols above. He decided that he should "do more examples so that I can be more sure [*sic*] that my conjecture is true". So he tried Example 3 (specialising): share 12 sausages equally among 16 people. This was *not* considered the start of another cycle since the fourth cycle had not ended as he had not solved his extension. He used other heuristics (H) to solve Example 3 and then entered the C stage by verifying that his conjecture was still correct. Next, he entered the J/G stage to prove his conjecture, which was a general result, by using reasoning, but he failed. Then the test ended.

To summarise, the student (S9) had done very well to obtain the general formula for the least number of cuts. As the formula was not easy to obtain, the students were not expected to find it: they were only expected to specialise systematically for different numbers of sausages and / or people, and to *try* to find the formula by searching for patterns. But the student (S9) only used two random examples to search for patterns and he found the general formula. Although it was not clear from his protocols how he managed to do that, a parallel could be drawn about his solution and the formal proof for the formula (the reader should refer to Appendix E on page 509 to be familiar with the main idea behind the proof). We will just consider Case 2 where the number of sausages n is less than the number of people m. If n and m are co-primes (Case 2a), the least number of cuts is $m - 1$. The main idea is to arrange all the sausages in a row and treat it as one long sausage as shown in Figure 6.16. Since there are m people, the least number of cuts must be $m - 1$, provided that the cuts do not coincide with the gaps between the sausages, which is true if n and m are co-primes. Flash back to Example 2 in Figure 6.15 when S9 made a mistake and cut 5 sausages for 9 people, instead of 10 sausages for 18 people in his extension. Notice that he cut the sausages from the first one to the last one row by row, as if they were arranged in a row. Since 5 and 9 are coprimes, his mistake for Example 2 is actually Case 2a.

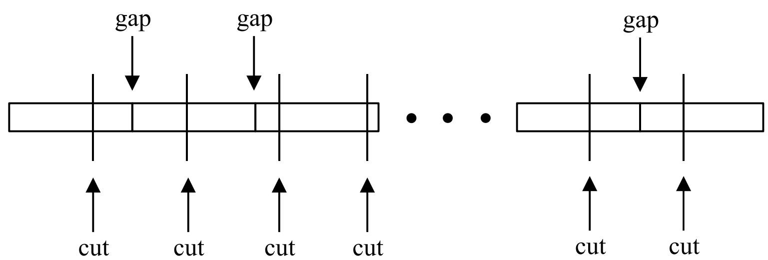


Figure 6.16 Proof of General Formula for Sausage Task

After the student discovered that it should be 10 sausages and 18 people, where 10 and 18 are not coprimes (Case 2b in formal proof), this scenario can be treated as 2 sets of the 5 sausages in Figure 6.15. Since he had found that the formula for the least number of cuts for the 5 sausages and 9 people is $m - 1 = 9 - 1 = 8$, the formula for the 10 sausages and 18 people can be viewed as $2(m - 1) = 2m - 2 = m' - \text{HCF}(n',m') = 18 - \text{HCF}(10,18) = 18 - 2 = 16$, which is essentially the idea behind the proof for Case 2b, although this might not be how the student discovered that 2 = HCF (10,18). Thus what the student had done was actually how a person can discover the formula by using the underlying structure of the 5 sausages and 9 people, where 5 and 9 are coprimes (Case 2a), to generalise to Case 2b, where n' and m' are not co-primes. This is what Mason et al. (1985) called the use of the 'underlying structure' to explain why a proof works, or what Pólya's (1957) and Lakatos (1976) called the use of 'heuristic reasoning' as a scaffold to construct a formal proof (see Section 2.2.3e).

This will become clearer if one considers a 'counter example' of how I initially did this investigation, which will not lead to the general formula. Figure 6.17 shows how I cut the 5 sausages for the 9 people: since each person will receive 5/9 of a sausage, I just cut each sausage at the 5/9-mark to give to 5 people first; then I cut the 4/9 that remains of the last sausage into 4 equal parts, so that the remaining 4 people will each receive 4/9 + 1/9 = 5/9 part. Thus the least number of cuts is 5 + 3 = 8. Notice that the underlying structure of this cutting method is addition, unlike the subtraction structure of S9's method: 9 – 1 = 8. Since the structure of the general formula is also subtraction, my method will not lead me to discover the formula.

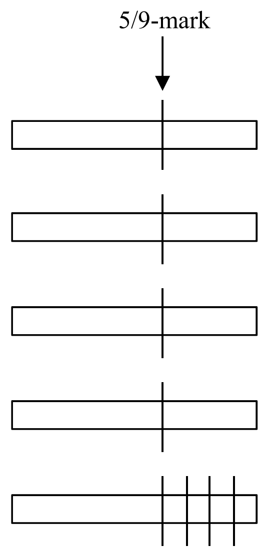


Figure 6.17 Another Cutting Method for Sausage Task

It is interesting to note that all the other 9 students did not cut the sausages in the manner that S9 did it. For example, all the other 4 students (S1, S4,S8,S10), who used Cutting Method B for the original task, cut the sausages at the 2/3-mark in a way similar to how I did it in Figure 6.17. Only S9 arranged the sausages in a row and cut from left to right, as shown in Figure 6.14 above, where the second sausage was cut at the 1/3-mark instead of at the 2/3-mark. This probably explains why all the other students did not discover the formula.

(b) Complete Pathway for Student S2

Let us now analyse the investigation pathway of another student (S2) who reached the Justifying / Generalising (J/G) stage during the extension of the Sausage Task. Does that mean that his investigation outcomes will be quite similar to those of the previous student (S9)? The pathway in Figure 6.13(b) shows that the student (S2) went through the first cycle from the Understanding (U) stage to the Checking (R) stage without entering the Problem-Posing stage (P). From his protocols and answer script, it was observed that he did not pose any specific problem, but he just went ahead to find how to cut the 12 sausages. He said that "the simplest way" (in his own words) would be to cut each sausage into 18 equal parts (called Cutting Method C in the task analysis of this task in Appendix E, which the students were not expected to use, as it is a long method). Figure 6.18 shows how he drew and cut a sausage into 18 equal parts.

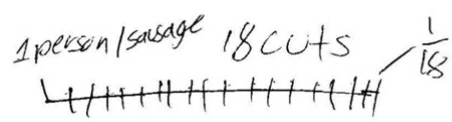


Figure 6.18 Cutting Method C of S2 for Sausage Task

In the process of cutting, he found the total number of cuts without posing this as a problem. He went into the R stage to review his solution and concluded that the cutting method was too troublesome. He then entered the P stage in the second cycle (see Fig. 6.13b) by posing the problem of finding the least number of cuts explicitly, which was the intended problem for this task. He proceeded into the stage of Using Other Heuristics (H) by finding the fraction of sausage each person will receive. But it still took him quite a while to figure out that he could cut each sausage into 3 equal parts, which is Cutting Method A, and he also found that the total number of cuts was 24. He did not discover Cutting Method B that gives the least number of cuts at 12. Instead, he wrongly accepted that 24 cuts was the least number of cuts because he did not try to prove it but went on to extend the task by changing the given to 3 sausages and 20 people respectively (Extension 1).

He then went into the H stage in the third cycle to solve his extension by finding how to cut the sausages, and the amount of sausages each person will receive. Because the numbers 3 and 20 are co-primes, it was very natural for him just to cut each sausage into 20 equal parts. Then he just swapped the two numbers around to 20 sausages and 3 people (Extension 2). This time, he went into the H stage in the fourth cycle to find the amount of sausages each person will receive, without even finding the cutting method. He then reviewed (R) his solutions and realised that it was “just simple division” (in his own words). What he meant was that the amount of sausages each person will receive could be found by using a simple division. Next, he tried to think of other extensions but he was stuck. There was a period of hesitation (coded as X for the stage code and shown as a *transition* in between stages in the ITR in Fig. 6.13b) for about 2 minutes. Suddenly, he entered the Conjecturing (C) stage by writing down the general formula for sharing things equally as shown in Figure 6.19.

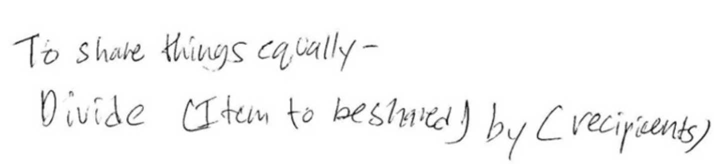


Figure 6.19 General Formula for Sharing by S2 for Sausage Task

He treated this formula as a conjecture as his subsequent protocols show that he was unsure of the formula. Then he entered another period of hesitation (X) as he did not know what to do next. In the end, he extended the task for the third time by randomly changing the given to 10 sausages and 17 people respectively (Extension 3). So he entered the H stage in the fifth cycle by using the above general formula to find the fraction of sausage each person will receive. He then decided to test the formula by changing the context of the original task from sharing sausages to sharing the usage of a computer among 9 brothers over one week, thus entering into the Justifying / Generalising (J/G) stage. He found that each brother will get to use 'up to $18\frac{2}{3}$ hours of computer' by dividing 7 days × 24 hours by 9. In the end, he concluded:

> "So it's proved that, I think it does not only work on food stuff … [start writing] works on other things with numerical value [stop writing]." [S2; Sausage Task]

However, there was no need to prove the formula in the first place since the formula was derived based on simple reasoning that the student had learnt in primary school mathematics on fractions. In the end, he wrongly accepted the formula as true based naïve testing. He then entered the R stage to check all his solutions.

To summarise, the student (S2) did not do well for the investigation for two main reasons: (i) he was unable to discover Cutting Method B that gives the least number of cuts, and (ii) he did not extend the task to generalise the least number of cuts, but he went on to find a trivial formula for sharing *n* items equally among *m* people, unlike the previous student (S9) who found a non-trivial general formula for the least number of cuts. Thus the two students (S2,S9) actually went in totally different directions in their investigation even though both of them completed their pathway. In other words, a complete investigation pathway does not indicate that a student has found any non-trivial result for an investigative task.

(c) Incomplete Pathway for Student S1

We will now analyse an incomplete investigation pathway of a student (S1). The pathway in Figure 6.13(c) shows that the student went through the first cycle from the Understanding (U) stage to the Problem-Posing (P) stage, followed by the stage of Using Other Heuristics (H). From his protocols and answer script, it was observed that he read the task statement only once and immediately posed the problem of finding how to cut the 12 sausages. Then he thought of cutting each sausage into 3 equal parts (Cutting Method A), which solved his first problem. In the process, he found that each person will receive 2/3 of a sausage. He then entered the P stage in the second cycle by posing his second problem, which was the intended problem for this task: "What is the least number of cuts?" He thought about the fraction of sausage each person will receive for a while, before discovering Cutting Method B (cut each sausage at the 2/3-mark), which gives the least number of cuts at 12. However, he accepted this conjecture as true without testing, since he then entered the Checking (R) stage to check his solution and he was satisfied that his working was correct.

Next, the student explicitly said that he wanted to extend the task (E). He then swapped the two given numbers around to give 18 sausages and 12 people (Extension 1). While he was writing his Extension 1, he suddenly realised that he needed to find a general formula for the least number of cuts (Extension 2), which was the intended extension. So he entered the Specialising (S) stage of the third cycle to select another example to specialise: he chose 4 sausages and 10 people because he wanted it to be similar to the original task, where the number of sausages is less than the number of people. Although he had discovered Cutting Methods A and B when solving the original task, he was still clueless about how to cut the 4 sausages in his extension. He began by cutting each sausage into 2 equal parts, and then he realised that there were not enough pieces to share. In the end, he decided to enter the R stage to review how he cut the sausages for the original task. In the process, he queried whether the least number of cuts for the original task could be less than the 12 cuts that he had obtained. So he entered into the Justifying (J) stage to prove that the least number of cuts was indeed 12 by using a non-proof argument (see his proof in Section 7.3.5). But there was no generalisation because this was not a general result. That was why the J stage was not combined with the Generalising (G) stage in this case.

He then went back to the H stage to try to solve his Extension 2 by applying the cutting methods used in the original task. But he was stuck because he did not find the LCM when using Method A to cut the sausages for the original task, so he was unable to apply the main idea behind Method A, which is LCM, to the extension. At first, he thought that the 'formula' had something to do with multiples of 10. In the end, after struggling for another 2 minutes, he realised that the total number of sausages after cutting had to be a multiple of 10, so he discovered that he could use the LCM of 4 and 10, which is 20, to help him find how to cut the sausages: since there will be 20 pieces of sausages after cutting to share equally among the 10 people, then each sausage should be cut into 20/4 = 5 equal parts, which is essentially Method A. He then found that the fraction of sausage each person will receive is 2/5, which led him to cut each sausage twice: at the 2/5-mark and the 4/5-mark. This is essentially Method B that gives the least number of cuts at 8 in this case. Again, he accepted this conjecture as true without testing. He then entered into the Conjecturing (C) stage to search for a formula linking the fraction of sausages each person will receive for the original task (i.e. 2/3) and for the extension (i.e. 2/5) as shown in his protocols:

> "Um … 2/5. [Flip to p. 1] This is 2/3. [Turn back to p. 3] Oh yah, um … So if I have … there seems to be some relationship between the numbers … Uh, I can find a formula maybe." [S1; Sausage Task]

Then he went back to the H stage to write his solution for his extension properly as shown in Figure 6.20.

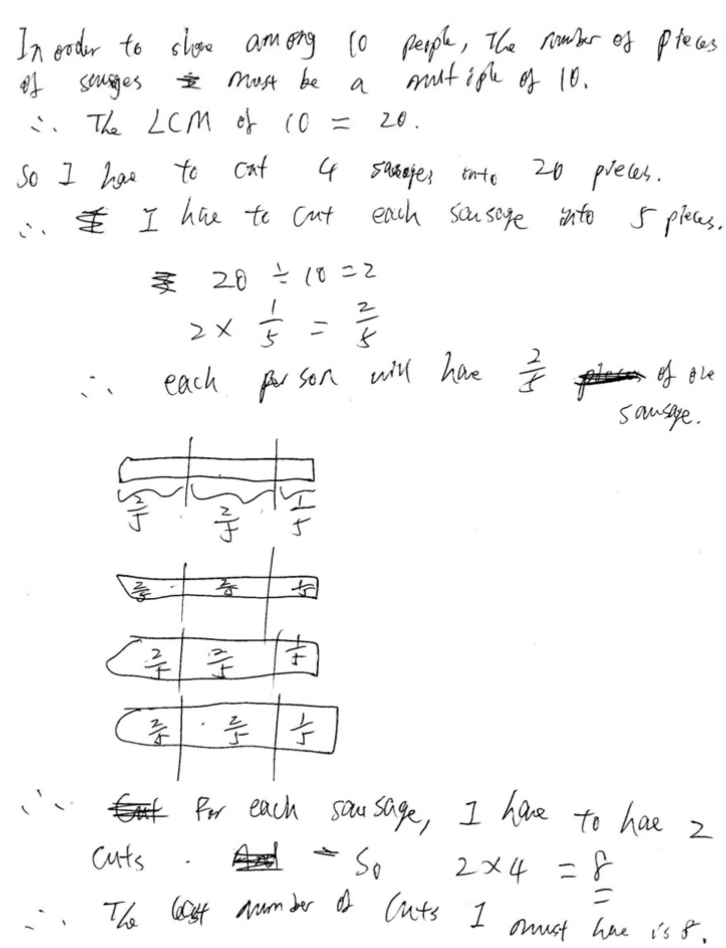


Figure 6.20 Cutting Method B of S1 for Extension of Sausage Task

He then entered the C stage again to try to find a formula for the least number of cuts by comparing the original task and his extension, but he failed to find one. So he decided to go back to the E stage to write down his extension because he had previously posed his extension verbally. This time, he used algebraic notations to represent the numbers of sausages and people as shown in his working in Figure 6.21:

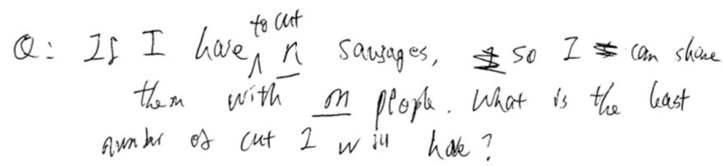


Figure 6.21 Extension of S1 for Sausage Task

He then entered the C stage again to try to generalise the cutting method for his extension by using algebraic notations, but he did not know how to write LCM(n,m). Instead, he used $2m$ to represent LCM(4,10) since LCM(4,10) = 20 = 2 × 10 for the extension, where m = 10 is the number of people. He reasoned that each sausage should be cut into $2m$ / n equal parts, which is Cutting Method A, but he then made a mistake in thinking that the 10 people will share a total of 2 × $2m$ pieces of sausages, when in fact it should just be a total of $2m$ pieces. Soon, he was stuck as he did not know how to represent how to cut each sausage in terms of algebraic notations. The test ended without him discovering any formula for the least number of cuts.

To summarise, the student (S1) was able to pose the intended problem of finding the least number of cuts for the original task, and the intended extension of finding a general formula for the least number of cuts. He even proved that 12 is the least number of cuts for the original task, but he was unable to discover the general formula. Thus he was not able to complete the investigation pathway for the Sausage Task. However, compared with the previous student (S2) who completed the pathway for the same task, this student (S1) actually performed better in the investigation because the previous student (S2) only obtained a trivial formula for sharing n items equally among *m* people. Therefore, the ability to complete an investigation pathway does not mean that the student is able to produce significant outcomes.

(d) Incomplete Pathway for Student S7

We will now analyse an incomplete pathway from another student (S7). The pathway in Figure 6.13(d) shows that the student started with the Understanding (U) stage and then she went into the Problem-Posing (P) stage. But she did not pose any problem explicitly as she just said, "So … I think that … need to find the …" She then went into the stage of Using Other Heuristics (H) by finding multiples of 18 and observing that 36 is divisible by 12, i.e. she had found the LCM of 12 and 18, which is 36. This led her to conclude that each of the 12 sausages should be cut into 3 equal parts (Cutting Method A) in order to share them equally among 18 people, and each person will get 2 parts. Her working thus far is shown in Figure 6.22.

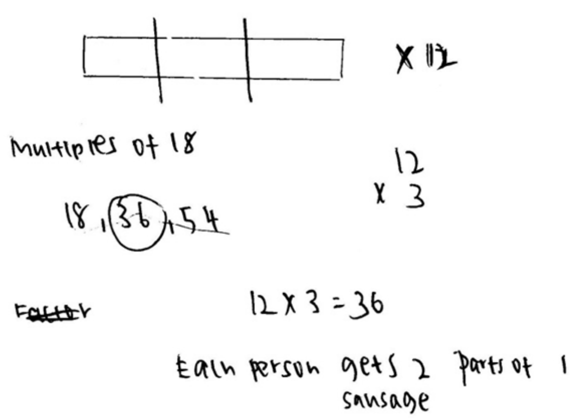


Figure 6.22 Cutting Method A of S7 for Sausage Task

She then found the next common multiple of 12 and 18, which is 72, and she obtained 6 when she divided 72 by 12. This led her to cut each sausage into 6 equal parts (her second cutting method). Immediately, she went into the Conjecturing (C) stage by observing that the number of parts each sausage should be cut into is a factor of 18 since 3 and 6 are factors of 18. But this is wrong because the number of parts each sausage can be cut can be any multiple of 18 as well. She then cut each sausage into 9 parts (her third cutting method), which is a factor of 18, in order to be more certain of her observation. She calculated $18 \times 9 = 162$, which is not divisible by 12, so she thought that her observation was wrong. But she soon discovered her mistake: it should be $12 \times 9 = 108$, which is divisible by 18 people, so that each person will get 6 parts. The student then paused to monitor her progress:

> "Am I going on the right track? [Flip to p. 1] … I don't think I know. [Turn back to p. 2] … Maybe I should try for another 5 minutes and see how it goes ..."
> [S7; Sausage Task]

She then decided to list the factors of 18, but this time she confused herself: she thought that the number of parts each person will get is a factor of 18. She checked her working and soon discovered this mistake. Then she cut each sausage into 18 parts (her fourth cutting method, which happened to be Cutting Method C), which is a factor of 18, and realised that it could still work. But she monitored her progress again and concluded that she was going nowhere:

> "I seem to be going nowhere … I'm just revolving around that … factors of 18."
> [S7; Sausage Task]

This suggests that she was still in the C stage and she had not yet formulated her observation as a conjecture to be proven or refuted. She then decided to think of other problems to pose. In the end, she decided to extend the task (E) in order to generalise.

She entered the Specialising (S) stage of the second cycle by choosing 12 sausages and 20 people. This was considered her Example 2 (the original task of 12 sausages and 18 people was considered as Example 1). She then went back into the H stage to solve her Example 2 by using the same approach: find factors of 20. She cut each sausage into 2 parts, which is a factor of 20, and realised that it did not work. So she monitored her progress for the third time and believed that she was on the wrong track. Interestingly, she then decided to incubate for a short while:

> "Let me rest my mind for a while ... I think I'm very confused. [Pause 3 s]"
> [S7; Sausage Task]

Incubation is a stage of creativity in Wallas' Creativity Model (see literature review in Section 2.2.2e) where a person takes a break from getting stuck in problem solving and relaxes the mind: think about the problem in a more relaxed state and environment and let the images from the subconscious surface. It was taught to the students during Lesson 6 of the teaching experiment (see Appendix C), but incubation and thinking aloud are diametrically opposite: you cannot incubate when you are thinking aloud. So it was surprising that the student tried to incubate for 3 seconds during the test. In fact, she did it three times during the investigation for the Sausage Task (she did not incubate for the other posttest task and all the other students did not incubate at all). The following shows her protocols for the second incubation period where she seems to understand the meaning of incubation:

> "I think, perhaps, I should let my mind rest for a while, then maybe an idea will just pop up. [Pause 6 s]" [S7; Sausage Task]

After her first incubation period of 3 seconds, she went into E stage the second time by re-reading the task and thinking of what problems to pose. In the end, she decided to extend the task in order to generalise again. She entered the S stage by choosing 5 sausages and 18 people (Example 3). Then she went back into the H stage to solve her Example 3 by using the same approach again: she listed the factors of 18 and then cut each sausage into 6 parts, which is a factor of 18, but it did not work. Next, she tried cutting each sausage into 3 parts and then 9 parts, which are also factors of 18, but they also did not work. So she monitored her progress for the fourth time and then decided to try for another five more minutes. In fact, she monitored her progress a total of 10 times, but all her metacognitive behaviours were not effective: she did not know what else to do except to continue trying in the same direction.

The student decided to go back to try the original task. Since she was still in the same H stage as before, there was a need to indicate in her ITR when she was using other heuristics (H) to solve her extension and when she was using other heuristics (H) to solve the original task (see Fig. 6.13d). This time, she drew all the 12 sausages to visualise how to cut, but she ended up listing the multiples of 12, multiples of 18 and factors of 18 to help her to think of a cutting method. Then she was stuck again. So she decided to enter the S stage by choosing 24 sausages and 9 people (Example 4) for her extension. To be clear that she was now extending the task, instead of solving the original task, there was a need to indicate this in her ITR (see Fig. 6.13d). She then went back into the H stage to solve her Example 4 by listing multiples of 9 and factors of 24, but she was still stuck. So she went back into the S stage by choosing 22 sausages and 8 people (Example 5), followed by the H stage to solve her Example 5. This time, she found the HCF of 22 and 8, which is 2, and then she cut each sausage into 2 parts, which did not work. So she went back to the original task and used the HCF approach, but it still did not work. Then the test ended.

Overall, the student (S7) went on a wild goose chase. Although she monitored her progress numerous times, her metacognitive processes were not effective since she did not know what else to do except to continue in the same approach using multiples and factors. She was the only student in the present study who tried to search for a pattern in the multiples and factors of the numbers of sausages and people in order to determine how to cut the sausages. Perhaps she was confused between the two types of tasks where she was supposed to search for any pattern in a Type A task, but to pose a specific problem to solve for this Type B task. Just like the previous student (S1), this student (S7) did not reach the Generalising (G) stage and so did not complete the investigation pathway, but she did not do well for her investigation, unlike the previous student (S1). This shows once again that whether an investigation pathway is complete or incomplete does not tell much about the actual investigation.

(e) Pathways for the Remaining Six Students

For the remaining 6 students, just like S2 and S9 described earlier in this section, 3 of them (S3,S4,S8) had completed their investigation pathway by reaching the Justifying / Generalising (J/G) stage since they had attempted to justify a general result during the extension of the Sausage Task. The remaining 3 students (S5,S6,S10) did not complete their pathway as they did not formulate any conjecture for the extended task. Table 6.2 shows a summary of their investigation pathways and outcomes.

Table 6.2 Summary of Pathways and Outcomes for Type B Tasks

	Non-trivial Outcomes	Trivial Outcomes / Misinterpreted Task
Complete Pathway	**S9**, S8	**S2**, S3, S4
Incomplete Pathway	**S1**, S10	**S7**, S5, S6

* Students in bold are those whose transcripts have been described in detail earlier

One would expect some variations in their investigation because no two students' outcomes would be exactly the same although they might have completed the pathway. For example, the non-trivial conjectures formulated by S8 and S9 were very different in nature: S9 found a general formula for the least number of cuts, while S8 discovered a generalised cutting method to share $6n$ sausages equally among $6n + 6$ people. But the trivial conjecture formulated by S4 was quite similar to the one formulated by S2: S4 found a general formula for the amount of sausages each person will receive, but S2 generalised further to a formula for sharing n items equally among m people. All these conjectures will be described in detail in the data analysis in Sections 7.3.8 and 7.3.9 later.

For those who did not complete their pathway, one of them (S10) discovered Cutting Method B that gives the least number of cuts for the original task, just like S1 described earlier in this section, while the other two students (S5,S6) did not discover Cutting Method B, just like S7 described earlier. However, there were still some differences in their investigation. For example, S10 did not extend the task to find a formula for the least number of cuts, unlike S1; while S5 did not use factors and multiples, unlike S7. On the other hand, S6 also used factors and multiples like S7, but unlike S7, S6 was concerned that the middle part of a sausage is different from its rounded ends in terms of shape when a sausage is divided into 3 equal parts. The details of their investigation will be described in the data analysis in Section 7.3 later.

(f) Summary of Mathematical Investigation Pathways for Type B Tasks

Five out of the 10 students completed the investigation pathway for the Sausage Task (Type B) as they were able to reach the Justifying / Generalising (J/G) stage by attempting to justify a general result during the extension of the task. The other 5 students did not complete the investigation pathway as they were unable to formulate any conjecture for the extended task. However, it was discovered that the pathways of the students do not reveal much about their outcomes. Two students could have similar pathways but they could have investigated totally different things. For example, both S2 and S9 completed the pathway, but S9 found a non-trivial general formula for the least number of cuts while S2 discovered a trivial formula for sharing n items equally among m people. Similarly, the inability to complete an investigation pathway does not mean that the student was poor in his or her outcomes. For example, S1 did not complete the pathway, but he was able to discover and prove that Cutting Method B will give the least number of cuts for the original task, and he tried to find a general formula for the least number of cuts, which was the intended extension for the task, although he did not succeed. But S2, who completed the pathway, was unable to find Cutting Method B, and he did not know how to extend the task to generalise the number of cuts that he had found. Instead, he went in a totally different direction and found a trivial general result that is clearly true.

6.4 INVESTIGATION PATHWAYS ACROSS BOTH TYPES OF TASKS

The theoretical investigation model for cognitive processes developed for the present study in Section 3.2.1 had posited that the pathways for the two types of investigative tasks would be different because of their diverse natures. After analysing the students' investigation pathways for the Kaprekar Task (Type A) and the Sausage Task (Type B) in Sections 6.2 and 6.3 respectively, this section will compare their pathways across both types of investigative tasks.

With reference to the students' IPD and ITR for the Type A task shown in Figure 6.1 on pages 223-224, it was observed that they had engaged in Specialising / Conjecturing (S/C) very often, and they did not use other heuristics (H) or extend (E) the task. These were in accordance with what were posited in the theoretical investigation model for Type A tasks, where the students were expected to search for any pattern (conjecturing) by trying examples (specialising), without the need to change the given to extend the task because extension will usually change the task to a new one with completely different patterns. With reference to the students' IPD and ITR for the Type B task shown in Figure 6.13 on pages 247-248, it was observed that they had spent a lot of time in the H stage, and they only entered the Specialising (S) stage or the Conjecturing (C) stage after extending (E) the task. Again, these were consistent with the prescription of the theoretical investigation model for Type B tasks, where the students were expected to solve specific problems for the original task by using other heuristics, before they extend the task to generalise by trying examples (specialising) to search for patterns (conjecturing).

Another difference between the two types of tasks is the Justifying / Generalising (J/G) stage. For Type A tasks, conjectures formulated based on specialising are usually general results, so the students could enter the J/G stage to try to justify their conjectures, which will lead to generalisation if proven (e.g. see the J/G stage in the pathways of S5 and S9 in Figure 6.1). But for Type B tasks, conjectures formulated for the original task based on using other heuristics are usually not general results, so the students could enter the Justifying (J) stage without generalising (G) even after proving their conjectures (e.g. see the J stage in the pathways of S1 and S7 in Figure 6.13). However, conjectures formulated for the extended task based on specialising are usually general results, which will lead to generalisation if proven (e.g. see the J/G stage in the pathways of S2 and S9 in Figure 6.13).

On the other hand, the actual pathways show that the students did not follow the pathways of the theoretical investigation model exactly. For example, some of them skipped the Problem-Posing (P) stage and went straight from the Understanding (U) stage to the S/C stage (e.g. see the pathways of S1, S5 and S10 for the Type A task in Figure 6.1), while others skipped the Checking (R) stage after solving a problem (e.g. see the pathways of the four students for the Type B task in Figure 6.13 where they entered some of the new cycles without checking). One of them (S9) even extended the task before checking his previous solution for the Type B task. These are to be expected as the theoretical investigation model only prescribes the logical pathways for an investigation.

6.5 SUMMARY OF ANSWER TO RESEARCH QUESTION 1

This section will summarise the main findings from examining the data collected for the present study in order to answer Research Question 1. Analysis of the students' investigation pathways in Section 6.2 and 6.3 has suggested that their pathways generally do not have a direct relationship with the outcomes across both types of investigative tasks. An investigation pathway only presents a global picture of the interactions of a student's processes when he or she attempts an investigative task, but it does not tell much about his or her investigation outcomes. Two students could have similar pathways but they could have investigated totally different things. For example, both S5 and S9 went through the complete pathway for the Type A task, but S9 misinterpreted the task and observed patterns that are different from those for the original task, while S5 understood the task correctly and discovered a non-trivial pattern for the original task. A student could also complete an investigation pathway by formulating and attempting to prove a trivial conjecture that is a general result (e.g. S2 for the Type B Task). Therefore, a complete investigation pathway does not indicate that the student has discovered a non-trivial pattern or solved a non-trivial problem that results in a generalisation.

Similarly, the inability to complete the investigation pathway does not mean that the students were poor in their outcomes. For example, S10 was stuck in the Specialising / Conjecturing (S/C) stage for most of her investigation but she discovered a non-trivial pattern for the Type A task, which is a lot more complicated than the pattern found by S5 who had completed her pathway. Another example is S1 who had proven that Cutting Method B will give the least number of cuts for the Type B task, and he even tried to find a general formula for the least number of cuts but failed to complete the pathway, compared with S2 who completed the pathway by attempting to prove a trivial general result, but was unable to discover Cutting Method B or extend the task to generalise the number of cuts that he had found. Therefore, an incomplete investigation pathway does not indicate that the student has not discovered a non-trivial pattern or formulated a non-trivial conjecture.

6.6 CONCLUDING REMARKS FOR THIS CHAPTER

Chapter 6 has answered Research Question 1 on the relationship between the investigation pathways of Secondary 2 students and their outcomes across the two types of investigative tasks. Chapter 7 will then answer Research Question 2 on the effect of the students' processes on the outcomes of their investigation.

CHAPTER 7: DATA ANALYSIS OF MATHEMATICAL INVESTIGATION PROCESSES AND OUTCOMES

In this chapter, the 20 sets of students' thinking-aloud protocols and answer scripts obtained from the posttest in the present research will be analysed to study the investigation processes and outcomes of the 10 Secondary 2 students across the two types of investigative tasks in order to answer Research Question 2. The posttest tasks were chosen as they provided a wider range of processes and outcomes to examine, just like for Research Question 1 as explained at the start of Chapter 6.

7.1 THE SECOND RESEARCH QUESTION

Research Question 2 is reproduced below:

RQ2: What is the effect of the cognitive and metacognitive processes of Secondary 2 students on the outcomes of their investigation?

Scope of Data Analysis

This chapter will begin by using the Summary Tables of Processes and Outcomes (TPO) developed in Chapter 5 to analyse whether the 10 students' cognitive and metacognitive processes had helped them to produce significant outcomes in their investigation of the Kaprekar Task (Type A), e.g. posing non-trivial problems and formulating non-trivial conjectures. This method of data analysis will be repeated for the Sausage Task (Type B). Finally, the data analysis will be used to validate the two theoretical investigation models and to refine them if necessary.

7.2 MATHEMATICAL INVESTIGATION PROCESSES AND OUTCOMES FOR TYPE A TASKS

As explained in Section 5.3 in the construction of the TPO, the processes and outcomes are different for each investigation stage and so the format of the TPO would be different. Sometimes, it was also necessary to construct more than one TPO for each stage because there were too many processes and outcomes to show in one table. Table 7.1 shows the processes and outcomes for each investigation stage for the Kaprekar Task (Type A). The reasons for grouping the Justifying and Generalising stages as the J/G stage have been explained in Section 4.4.4(b). Stage 8 (Extension) is not included because the students were not expected to extend the Type A task within 30 minutes of investigation, and they also did not extend. The classification of an outcome (problem posed or conjecture formulated) as trivial or non-trivial had passed the inter-coder reliability test (see Section 5.4).

In this section, the 10 sets of thinking-aloud protocols and answer scripts for the Kaprekar Task will be analysed to study the effect of the 10 students' processes on their investigation outcomes. The Kaprekar Task is reproduced below.

Posttest Investigative Task 1: Add Sum of Digits to Number

Choose any number. Add the sum of its digits to the number itself to obtain a new number. Repeat this process for the new number and so forth. Investigate.

Table 7.1 Investigation Processes and Outcomes for Type A Tasks

Stage	Cognitive Processes	Metacognitive Processes	Outcomes
Stage 1: Understanding the Task (U)	• Re-reading task (RR) • Rephrasing task (RT) • Highlighting key information (HI) • Trying example (TE)	• Monitoring understanding (MU)	• Understood task correctly • Misinterpreted task • Recovered from misinterpretation
Stages 2: Problem Posing (P)	Referring to following to think of problem to pose: • task statement • current working • previous result • given checklist	• Analysing feasibility of goal or problem posed (MG)	• Posed general problem • Posed trivial specific problems • Posed non-trivial specific problems
Stage 3: Specialising (S)	• Random specialising • Purposeful specialising • Systematic specialising	• Analysing feasibility of plan to specialise (MF) • Metacognitive awareness (MA)	• Generated representative examples • Generated non-representative examples
Stage 4: Conjecturing (C)	Searching for patterns in the following: • Terms of sequence • Differences between consecutive terms • Across sequences	• Analysing feasibility of plan to search for patterns (MF) • Metacognitive awareness (MA)	• Observed trivial or non-trivial patterns • Formulated trivial or non-trivial conjectures
Stages 5 / 6: Justifying / Generalising (J/G)	• Naïve testing to refute conjecture • Using a non-proof argument • Using a formal proof	• Analysing feasibility of plan to justify conjectures (MF) • Metacognitive awareness (MA)	• Proved conjecture leading to generalisation • Proved conjecture but did not lead to generalisation
Stage 7: Checking (R)	Checking correctness of working (CW): • step by step for most parts • step by step for some parts only • glancing through it briefly	• Monitoring progress[12] (MP) • Reviewing solution to see if it had achieved the goal (MR) • Metacognitive awareness (MA)	• Discovered major errors on time • Discovered major errors late • Did not discover major errors at all

[12] Since monitoring progress can occur in any stage, and it does not matter at which stage the students monitor their progress (unlike metacognitive awareness), it will be more appropriate to analyse this process for the entire investigation together with the processes in the checking stage.

7.2.1 Understanding the Task (Stage 1)

Table 7.2 shows the TPO for the understanding processes and outcomes for the 10 students' investigation of the Kaprekar Task. It was observed that 7 of the 10 students tried one example (TE) each while 3 students tried 2 examples each to understand the task. Only one student (S2) did not appear to re-read (RR) or rephrase the task (RT), highlight key information (HI) or monitor his understanding (MU). The other 9 students engaged in RR, RT, HI and MU from 4 to 22 instances each. RR occurred a lot more frequently than RT, HI and MU, with some students re-reading the task 7 to 9 times each. Only 5 students engaged in HI. Although 6 students engaged in MU, its frequency is the lowest at 9 occurrences. A total of 3 students misinterpreted the task, out of whom, 2 of them recovered after about two minutes into the investigation.

Table 7.2 Understanding Processes and Outcomes for Kaprekar Task

	Processes						Outcomes	
	TE	RR	RT	HI	MU	Total**	Understood	Misinterpreted
S1	1	3			1	**4**	✓	
S2	1					**0**	✓	
S3	1	2+1+2=5*		1	0+1+1=2	**8**	✓	
S4	1	2	1	3	1	**7**	✓	
S5	1	8	1	4		**13**	✓	
S6	2	7	2	3		**12**		Recovered after 2 min
S7	1	2+0+3=5	4		0+1+1=2	**11**	✓	
S8	2	3	4			**7**		Recovered after 2 min
S9	2	9	6	5	2	**22**		Did not recover
S10	1	2+3=5			0+1=1	**6**	✓	
Total	**12**	**47**	**18**	**16**	**9**	**90**	**7**	**3 misinterpreted; 2 recovered**

* S3 engaged in RR for 2 + 1 + 2 = 5 times means that there were 3 episodes of understanding the task and RR happened 2 times in the first episode, 1 time in the second episode, and 2 times in the third episode; 3 × RR for S1 means that RR happened 3 times in the first episode.
** The 'Total' column shows the total frequency for RR, RT, HI and MU (excluding TE) each student.

The issue was whether the understanding processes used by the students had helped them to interpret the task correctly or to recover from any misinterpretation, and how did these processes help. From the detailed protocol analysis of the 3 students (S6,S8, S9) who misinterpreted the task, it was observed that all of them misinterpreted the task because they did not re-read, rephrase or highlight the *relevant* parts of the task while trying their first example. The first student (S6) engaged in the understanding processes 6 times (not yet 7 times) while trying Example 1 but she still misinterpreted the task by not repeating the process for the new number because she did not re-read, rephrase or highlight the part of the task that said, "Repeat this process for the new number." It was only after she finished trying Example 2 that she re-read this part of the task and immediately she realised her mistake. Similarly, the second student (S8) did not re-read, rephrase or highlight the part of the task that said, "Add the sum of its digits to the number itself" and so she misinterpreted the task by treating the sum of the digits of a number as the new number. But because she ended up with a one-digit sum as the new number, she was stuck as she did not know how to find the sum of the digits of a one-digit number. Then she re-read "Add the sum of its digits to the number itself" three times and finally interpreted the task correctly.

The investigation pathway and outcomes of the third student (S9) for the Kaprekar Task had been discussed in detail in Section 6.2(b) on page 229. Just like S8 above, S9 misinterpreted the task by not adding the sum of the digits of the starting number to itself because he did not re-read, rephrase or highlight the part of the task that said, "Add the sum of its digits to the number itself." Because he was stuck with a one-digit sum as the new number, he recovered from his first misinterpretation after about two minutes into the investigation when he re-read the relevant part of the task. But he still did not repeat the process for the new number, just like the misinterpretation of S6 above. It was only after 20 minutes into the investigation when he looked at the task statement during the review of a solution that he suddenly realised that he must repeat the process. However, he only repeated the process once for each starting number, so he did not recover from his second misinterpretation at all. One possible explanation why he did not see the need to repeat the process more than once for each starting number is because he might have misunderstood that he could extend the task by changing the given at this stage.

The above data analysis suggests that the students should re-read one part of the task statement, perform the corresponding operation on their first example, and then repeat this procedure until all the operations are properly carried out, in order to understand the task correctly. In fact, this was exactly what 5 other students (S1,S3-S5,S7) had done: they re-read or rephrased all the *relevant* parts of the task during the trying of the first example and they interpreted the task correctly. The remaining 2 students (S2,S10) did not re-read, rephrase or highlight the relevant parts of the task while trying the first example, but they still managed to interpret the task correctly, although it could not be excluded that they might have referred to the task statement when trying the first example without reading it aloud.

Of the 6 students who monitored their understanding (MU), 3 of them (S1,S4,S9) did so at the start of the test but it did not prevent one of them (S9) from misinterpreting the task. The other 3 students (S3,S7,S10) engaged in MU at a later stage when they did not find any pattern, in order to check whether they had misinterpreted the task by performing the operations wrongly, but there was nothing wrong. Thus it seems that quite a number of the students went back to monitor their understanding because they were unable to discover any pattern, but students who were able to progress in their investigation might not engage in MU. For example, S5 was able to discover some patterns and so she did not appear to monitor her understanding.

Summary

To summarise, the main understanding processes that helped the students in the present study to interpret the task correctly or to recover from any misinterpretation were re-reading or rephrasing the *relevant* parts of the task statement while trying the first example to understand the task. The metacognitive process of monitoring the understanding did not seem to be helpful.

7.2.2 Problem Posing (Stage 2)

Table 7.3 shows the TPO for the problem-posing processes and outcomes for the 10 students' investigation of the Kaprekar Task. The classification of specific problems as trivial or non-trivial had passed the inter-coder reliability test (see Section 5.4). It was observed that only 3 students (S3,S7,S9) posed the general problem of searching for any pattern at the start of the investigation while the other 7 students just went ahead to search for patterns without verbalising the general problem. However, 2 students (S1,S5) went back to pose the general problem at a later stage when they were stuck. In other words, most of the students did not see the need to pose the general problem for Type A tasks explicitly.

Table 7.3 Problem-Posing Processes and Outcomes for Kaprekar Task

	Processes					Outcomes	
	Refer to following to think of problem to pose				MG	General Problem	Specific Problem
	Task Statement	Current Working	Previous Result	Given Checklist			
S1		1				Later	1 non-trivial
S2							
S3						Start	
S4							
S5	1		2	3		Later	2 trivial; 2 non-trivial
S6							
S7						Start	
S8							
S9			2			Start	2 trivial
S10							
Total	**1**	**1**	**4**	**3**	**0**	**5**	**4 trivial; 3 non-trivial**

It was further observed from Table 7.3 that 3 of the students also posed specific problems for the Type A task. The first student (S1) posed the problem of finding a formula for the general term of the sequence at about 8 minutes into the investigation while looking at his current working to search for patterns. However, he did not analyse the feasibility of the goal or the problem (MG). Although this is a non-trivial problem that is worth pursuing, it is not easy to find the formula since the differences between consecutive terms do not follow a fixed increasing or decreasing pattern. In fact, there is no known formula for the general term of a Kaprekar sequence. Even if the student could not determine at this stage that it was not going to be easy to find the formula, he did not monitor his progress when he was unable to find any pattern. Instead, he just persisted in trying to find the formula. As a result, he not only did not find the formula, but he also did not discover any other pattern. The rest of his investigation had already been discussed in detail in Section 6.2(d).

The second student (S5) posed four specific problems. When she failed to find any pattern, she referred to the given checklist of investigation processes (see Appendix H) before posing Specific Problem 1: look for patterns in the sums of digits of the terms. Similarly, she referred to the checklist when she failed to find any pattern again, but this time, she re-read the task before posing Specific Problem 2: look for patterns in the differences between consecutive sums of digits. Thus she posed the two trivial problems by referring to the checklist or the task statement to think of a problem to pose. Her two non-trivial problems had been described in detail in her investigation pathway in Section 6.2(a) on page 227. She was able to use her previous result, which was her Conjecture 1, as a springboard to pose Specific Problems 3 and 4 with analogous results. This is called problem posing by analogy, which was advocated by Kilpatrick (1987) and discussed in detail during the literature review in Section 2.2.3(h). Just like the previous student, this student did not analyse the feasibility of the goal or the problem (MG), but in her case, the four problems were worth pursuing.

The third student (S9) misinterpreted the task by not repeating the process for the new number, and his investigation had been described in detail in Section 6.2(b). He posed two trivial problems: find a pattern for two-digit numbers (if the process was performed once), and find a pattern for two-digit numbers if the process was performed twice (at this stage, he had discovered that he did not repeat the process for the new number, but he decided to perform the process only twice). In both cases, the student referred to the previous result when posing the problem. Just like the previous two students, this student did not analyse the feasibility of the goal (MG), but in his case, the two problems were worth pursuing.

Summary

To summarise, most of the students in the present study did not see the need to pose the general problem of searching for any pattern for Type A tasks explicitly, but they just went ahead to search for patterns. Unlike understanding the task, where re-reading or rephrasing the task statement had helped the students to interpret the task correctly, most of the specific problems posed for the Type A task were not the result of referring to the task statement to think of a problem to pose. Instead, the students referred to the given checklist of investigation processes or a previous result to think of a problem to pose. What is noteworthy is the ability of one student to use a conjecture as a springboard to pose two non-trivial problems with analogous results. However, none of the students analysed the feasibility of their goal, which might have helped prevent one of them from pursuing a goal that was too difficult to achieve.

7.2.3 Specialising (Stage 3)

Table 7.4 shows the TPO for the specialising processes and outcomes for the 10 students' investigation of the Kaprekar Task. Mason et al. (1985) advocated choosing examples randomly to understand the task and systematically to search for patterns. For example, if a student starts investigating from the number 10, followed by 11 and then 12, this is systematic specialising. However, what if a student intends to choose a two-digit number, but he or she chooses a random two-digit number instead? This will be called 'purposeful specialising' since the student *purposefully* chooses a three-digit number although the choice of the number is still random. Thus it is decided that there are three types of specialising processes that are mutually exclusive: (i) random specialising, (ii) purposeful specialising, and (iii) systematic specialising. Since all the students used the random example(s) generated for understanding the task to search for patterns as well, the table has also included these examples under random specialising. But examples used to test conjectures (naïve testing) in the justifying stage are *not* included in this table. The examples generated were considered representative if the students were able to generate all the three types of sequences: Type 1a, Type 1b and Type 2 (see task analysis in Appendix E on page 499).

Table 7.4 Specialising Processes and Outcomes for Kaprekar Task

	Processes					**Outcomes**		
	Types of Specialising			**MF**	**MA**	**Total no. of e.g.**	**Rep. Examples**	**Not Rep. Examples**
	Random	**Purposeful**	**Systematic**					
S1	2 (E.g. 1,2)					**2**		Type 1a,2
S2	3 (E.g. 1,2,4)	1 (E.g. 3)		1	1	**4**	✓	
S3	1 (E.g. 1)	2 (E.g. 2,3)				**3**		Type 1a,2
S4	1 (E.g. 1)	3 (E.g. 2-4)			1	**4**		Type 2
S5	2 (E.g. 1,2)	5 (E.g. 3-7)		2		**7**	✓	
S6	4 (E.g. 1-4)	2 (E.g. 5,6)				**6**		Type 1b,2
S7	5 (E.g. 1-3, 5,6)	2 (E.g. 4,7)				**7**	✓	
S8	6 (E.g. 1-6)					**6**	✓	
S9*	2 (E.g. 1-2)	1 (E.g. 3)	2 (E.g. 5-6)			**5**	NA	
S10	1 (E.g. 1)					**1**		Type 2
Total	**27 (60%)**	**16 (36%)**	**2 (4%)**	**3**	**2**	**45**	**4 (44%)**	**5 (56%)**

* S9 misinterpreted the task, so his patterns were no longer the same as the original task. As a result, his examples could not be classified as representative because there was only one type of examples. Moreover, E.g. 4 was not included because it was used to test conjecture in the justifying stage.

From Table 7.4, it was observed that the 10 students generated a total of 45 examples to specialise, or an average of 4.5 examples per student. The actual number of examples generated by a student ranged from 1 to 7 examples. Since the Type A task involves a sequence, it is possible to examine just one sequence because there will be many numbers from just one example. In fact, the student (S10) who tried only one example generated 37 terms, which was the highest number of terms for a sequence generated among all the students for this task. It was further observed that most of the examples (60%) chosen for specialising were random, probably because of how the task was phrased: "Choose any number." So 3 of the students (S1,S8,S10) might think that they were supposed to choose random numbers only. Only one student (S9) chose systematic examples, accounting for only 4% of all the examples generated for specialising, but this was because he misinterpreted the task by not repeating the process for the new number and so he chose consecutive starting numbers to search for patterns at a later stage of his investigation.

The examples chosen purposefully can be classified into two categories. The first category is to choose a particular type of numbers to search for *any* pattern. For example, 4 students (S3,S5,S7,S9) chose numbers with different digits, from one-digit to three-digit numbers, to try to find any pattern. The second category is to choose a particular type of numbers to be more certain of a pattern before formulating it as a conjecture. This is still in the specialising stage, unlike the naïve testing of a conjecture in the justifying stage. For example, S2 purposefully chose a starting number ending with 0 for his Example 3 because he thought that there was a pattern in the last digits of the terms in the sequence in his Example 2 but he was not sure of the pattern yet. Similarly, S4 purposefully chose numbers whose sum of digits is 5 for his Examples 2-4 because he observed some kind of pattern for his Example 1, where the sum of the digits of the starting number is also 5.

It was further observed from Table 7.4 that only 4 students generated representative examples that included all the three types of sequences (Types 1a, 1b and 2). Most of them tried either 6 or 7 examples, except for one student who tried only 4 examples.

Of the other 5 students who did not generate all the representative examples, most of them tried between 1 to 4 examples, except for one student who tried 6 examples. So it appears that the more examples the students tried, the higher the chance they might generate all the types of sequences. This is because they specialised randomly most of the time, and so it all depended on luck. However, the students should specialise systematically to ensure that they would generate all the representative examples.

Table 7.4 shows that only 2 students analysed the feasibility of their plan to specialise (MF), which is a metacognitive process. The first student (S5) could not find any pattern, so she referred to the given checklist of investigation processes (see Appendix H) on two occasions and analysed the feasibility of the methods given in the checklist, but concluded that methods, such as drawing a diagram and guess and check, were not suitable for the task. The second student (S2) started with a three-digit number and soon obtained a four-digit number. He then analysed the feasibility of the plan to specialise further, but for some strange reason, he decided not to continue the example because he believed that "it will be difficult" (in his own words). The following shows his protocols:

> "I already reach 1000, so I don't think I should continue further as I am unable to find … that will be difficult. Narrow my range down to below 1000 …" [S2; Kaprekar Task]

For the stage of specialising, students should analyse what numbers to choose to specialise purposefully or systematically. Some questions they could ask themselves include: "Should I choose to specialise from 1 to 9 systematically? Is it feasible or worth pursuing, or should I try 10 to 19 instead?" "Should I try a two-digit or a three-digit number? Does the number of digits matter for this task?" "I want to choose another starting number where the sum of its digits is 5 to see if this is the pattern. Is it feasible or worth pursuing, or should I choose another starting number instead?" However, none of the students analysed the feasibility of their plan in this manner. If they had, they might have realised that the number of digits in the starting number is not crucial for this task because the sequence is increasing, unlike the happy and sad numbers which they had encountered in their pretest. Similarly, for the student (S4) who purposefully chose a starting number whose sum of digits is 5 for his Examples 2-4 as described earlier, if he had analysed his plan, he might have realised he was restricting himself to only one type of sequences (Type 2) by choosing the starting numbers for all his examples[13] in this manner. He only accidentally generated another type of sequence when he made a calculation mistake in his last example that changed the sequence from Type 2 to Type 1b.

Another metacognitive process in this investigation stage is metacognitive awareness. This is a new process discovered in the course of the present research and so a new code MA was created in the fine-tuning of the coding scheme for the thinking-aloud protocols (the reader should refer to Section 4.6.1 on page 189 to be familiar with the description of this process). Students who possess MA are constantly aware or conscious of what they are doing. This awareness can help them save time as explained and illustrated by the following examples. Table 7.4 shows that only 2 students exhibited MA in the stage of specialising. The first student (S4) was trying Example 2 when he obtained 119 as shown in Figure 7.1.

---

[13] Although his Example 1 was chosen randomly, its sum of digits happened to be 5, which caused him to purposefully choose Examples 2-4 in this manner because he suspected that there was a pattern.

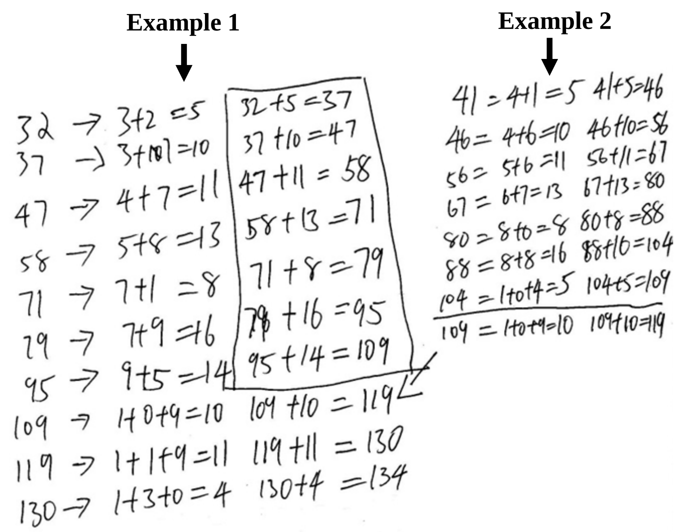


Figure 7.1 Examples Generated by S4 for Kaprekar Task

The student realised that 119 had occurred in his Example 1 and he drew an arrow to indicate the link. Actually, 109 in his Example 2 had already occurred in his Example 1 but he did not observe it. Nevertheless, the awareness that 119 had appeared before had saved him a lot of precious time because he did not continue his Example 2 since the subsequent terms for both examples would be the same. The second student (S2) obtained the number 379 in his Example 4 and he thought that he had obtained this number before. The following shows his protocols:

> "[Start writing] 365 + 14 = 379 [stop writing]. [Flip through the pages] When have I come across this before?" [S2; Kaprekar Task]

However, 379 had not appeared before. The closest number that he had obtained was 369 which appeared in his Example 2. Thus his metacognitive awareness was not effective in this case. The importance of MA will become more evident when we look at some counter examples. For example, a student (S8) tried an example with starting number 4 on page 2 of her answer script, but she did not realise that 4 was also the starting number for her Example 1 on page 1 of her answer script (see Fig. 7.2 which shows both examples). So she wasted quite a lot of precious time repeating her Example 1. Another student (S5) also made the mistake of trying the same starting number 12 for her Examples 1 and 5 without realising it.

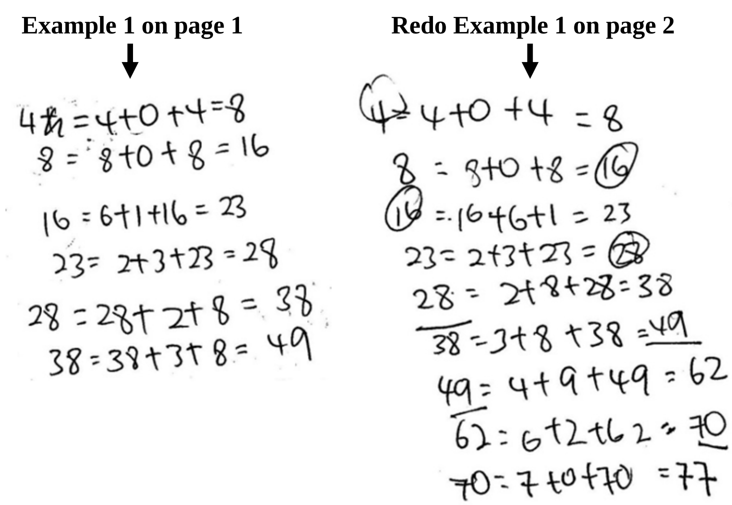


Figure 7.2 Same Examples Generated by S8 for Kaprekar Task

Summary

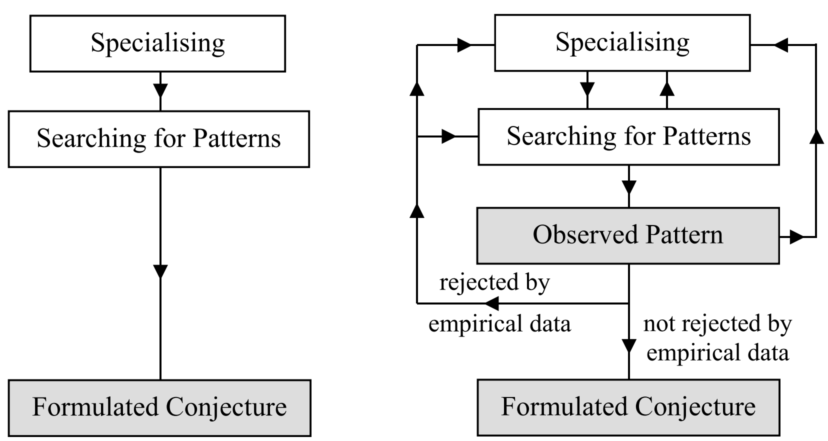
To summarise, most of the students in the present study specialised randomly instead of systematically to search for patterns. As a result, most of them did not generate all the types of sequences. Some students were able to choose examples purposefully to search for any pattern or to be more certain of a pattern, although they did not realise that the number of digits is not a crucial element for the Kaprekar Task. There were very few instances of metacognitive behaviours and most of them were not effective. However, metacognitive awareness had helped one student save a lot of time, but the lack of it had resulted in two students repeating examples that they had tried earlier.

7.2.4 Conjecturing (Stage 4)

The treatment for the data analysis in this section will be different from the previous sections because there is a need to discuss the new outcomes and interactions found in the course of the present research, and to analyse the types of patterns and conjectures discovered by the students, before examining the effect of the conjecturing processes on the investigation outcomes.

(a) New Outcomes and Interactions

Figure 7.3(a) shows parts of the theoretical model for cognitive processes developed for the present study (see Section 3.2.1 for the full model). It was posited in the model that students should try examples (specialise) and then search for patterns for Type A tasks. If they have found a pattern, the observed pattern is only a conjecture.



(a) Theoretical Model (b) Refined Model

Figure 7.3 Interaction between Specialising and Conjecturing for Type A Tasks

However, from the students' thinking-aloud protocols for the Kaprekar Task, it was found that all the students in the present study went from 'Specialising' to 'Searching for Patterns' and then back to 'Specialising' very often, either because they could not find a pattern and so they had to specialise some more, or they tried to search for patterns every time they generated a term for the sequence. Table 7.5 shows the protocols of a student (S7) who went through a few cycles alternating between trying examples (TE) and searching for patterns (SP) for the Kaprekar Task. Thus the theoretical model in Figure 7.3(a) needs to be refined to include the loop between 'Specialising' and 'Searching for Patterns' as shown in Figure 7.3(b).

Table 7.5 Alternating between Trying Example and Searching for Patterns

Line	Time	Protocols	Stage Code	Bhvr Code	Remarks
83	13:20	So [start writing] 681 … so + 6 + 8 + 1 = [stop writing and use calculator] so 6 + 8 + 1 = … eh? + 681 = 696 [write: 696].	S	TE4	Trying Example 4
84	13:40	So 696 [write: 696] … [use calculator] 696 … – 681 = 15 [draw ↕ between 681 and 696 and write: 15].	C	SP	Searching for patterns in differences between consecutive terms
85	13:54	[Continue writing] + 6 + 9 + 6 = [stop writing and use calculator] 696 + 6 + 9 + 6 = 717 [write: 717] …	S	TE4	Continue trying Example 4
86	14:09	So 717 [write: 717] minus [use calculator] 717 – 696 = 21 [draw ↕ between 696 and 717 and write: 21]. Never mind, I think I should continue trying, but 21 and 15, I don't see any link.	C	SP	Searching for patterns
87	14:27	717 [start writing] + 7 + 1 + 7 = [stop writing and use calculator] 7 + 1 + 7 = 15 … oh, 717 + 7 … + 1 + 7 = 732 [write: 732].	S	TE4	Continue trying Example 4
88	14:51	732 [draw ↕ between 717 and 732] – 717 = 15 again [write: 15].	C	SP	Searching for patterns

Moreover, it was observed from the students' protocols that there was usually a time gap between observing a pattern and considering the pattern as a conjecture, unlike what was posited in the theoretical model in Figure 7.3(a) where the observed pattern is immediately treated as a conjecture. This was because when the students first spotted a pattern, they were usually unsure that this was the pattern. So the students were observed trying more examples to be more certain of the pattern first, before they accepted it as a conjecture and then tried to prove it. Sometimes, they even found counter examples to reject the observed pattern before they could formulate it as a conjecture. Thus there was a need to distinguish between observing a pattern and formulating a conjecture, and so two new codes ‘Observed Pattern’ (OP) and ‘Rejected Observed Pattern’ (RP) were created in the refinement of the coding scheme as described in Section 4.4.2(a). Table 7.6 shows the protocols of the same student as above (S7) who observed Pattern 2a (OP2a) in her Example 4 (TE4), but she was not sure of the pattern, so she continued trying the example and found a counter example to reject the pattern (RP2a). If the behaviour in Line 89 was coded as SP (searching for patterns) instead of OP2a, then there would not be any pattern to reject in Line 92. Therefore, the theoretical model needs to be refined to reflect these two outcomes as shown in Figure 7.3(b): the loop between observing a pattern and trying more examples, and what happens if a pattern is rejected. The full refined investigation model will be shown later in Section 7.4.2.

Table 7.6 Observed Pattern and Rejected Observed Pattern

Line	Time	Protocols	Stage Code	Bhvr Code	Remarks
89	14:57	Eh, I seem to see a pattern in it.	C	OP2a	Observed Pattern 2a: Difference between consecutive terms seem to alternate between 15 and 21
90	14:59	732 [start writing] + 7 + 3 … + 2 = [stop writing] … [use calculator] 7 + 3 + 2 = … 12 … + 732 = 744 [write: 744].	S	TE4	Continue trying Example 4
91	15:21	744 [write: 744] [draw ↕ between 732 and 744 and write: 12]	C	SP	Searching for patterns
92	15:24	No, what, it doesn’t seem to be a pattern again.	C	RP2a	Rejected Pattern 2a because of counter example 12 in Line 91

However, it was possible for students to treat the observed pattern as a conjecture immediately after observing the pattern, in the manner posited in the theoretical model in Figure 7.3(a). Table 7.7 shows the protocols of a student (S9) who observed a pattern, which he immediately said was his conjecture (FC1) in Line 51.

Table 7.7 Observed Pattern Treated as Conjecture

Line	Time	Transcript	Stage Code	Bhvr Code	Remarks
43	04:09	Ok … for a … let's start easy … let's start with 2-digit numbers [write: (2 digit)]. Ok, for two-digit numbers like 12 [write 12], we add digit 1 and 2 [write: + 1 + 2] —	S	TE3	Trying Example 3
44	04:24	— we will get 14 [write: = 14].	S	EM2	Minor Error 2: should be 15
45	04:26	And for … another digit [*sic*: number] like 11 [write: 11].	S	EM3	Major Error 3: did not repeat process
46	04:29	We add 1 [write: + 1] we add 1 [write: + 1] we will get 13 [write: = 13] … So, what about 10 [write: 10]? We will have 1 + 0 [write: + 1 + 0] 11 [write: 11] …	S	TE3	Continue trying Example 3
47	04:40	Now, now, this is strange, right? …	C	MA	Found numbers strange, which led to discovery of Error 2 in Line 49
48	04:43	Let's try one more example. [Start writing] 13 + 1 + 3 [stop writing] we will get … um, if I am not wrong, it's 17 [write: 17], right? …	S	TE3	Continue trying Example 3
49	04:55	Now, I think I calculate this wrongly [cancel 14]. It's supposed to be 15 [write: 15].	S	ED2	Discovered Minor Error 2
50	05:00	Ok, as you can see for two-digit numbers —	C	SP	Searching for patterns
51	05:03	Ok, this is my conjecture. Ok, the conjecture [write: Conjecture:] for me now is that … um, the 2 [write: the 2] [cancel: the] [replace it with: all] all 2 [continue writing] digits numbers will add up to a new odd number [stop writing] …	C	FC1	Formulated Conjecture 1

In other words, an observed pattern (OP) is different from a formulated conjecture (FC) if the student tries more examples to be more certain of the observed pattern first, but if the student treats an observed pattern as a conjecture immediately after observing it, then his or her protocol will be coded directly as FC instead of OP.

(b) Conjecturing Outcomes

There is a need to analyse the types of patterns and conjectures discovered by the students before examining the effect of the conjecturing processes on the investigation outcomes. Table 7.8 shows the TPO for the conjecturing outcomes for the 10 students' investigation of the Kaprekar Task. The term 'Type 1 Patterns' refers to both Type 1 'multiples' and 'digital roots' patterns, and the term 'Type 2 Patterns' refers to both Type 2 'multiples' and 'digital roots' patterns (the reader should refer to the task analysis in Appendix E on page 499 to be familiar with these patterns). The classification of patterns and conjectures as trivial or non-trivial had passed the inter-coder reliability test (see Section 5.4).

Table 7.8 Patterns and Conjectures for Kaprekar Task

	Related to Types 1 and 2 Patterns				Other Types of Patterns				Total No. of Patt.	Total No. of Conj.
	Patterns		Conjectures		Patterns		Conjectures			
	non-trivial	trivial	non-trivial	trivial	non-trivial	trivial	non-trivial	trivial		
S1		2		1		1			**3**	**1**
S2		2			1		1		**3**	**1**
S3	1(1)	2		1		1			**4(1)**	**1**
S4		2		1					**2**	**1**
S5	3(3)		1(1)			3		1	**6(3)**	**2(1)**
S6		2		1		1			**3**	**1**
S7		2				1			**3**	**0**
S8	1(1)	1				1			**3(1)**	**0**
S9					2(1)*	2(1)	2(1)	2(1)	**4(2)**	**4(2)**
S10	1(1)	1	1(1)						**2(1)**	**1(1)**
Total	**6(6)**	**14**	**2(2)**	**4**	**3(1)**	**10(1)**	**3(1)**	**3(1)**	**33(8)**	**12(4)**
	20 (61%)		**6 (50%)**		**13 (39%)**		**6 (50%)**			

* The number in brackets indicates the number of correct patterns or conjectures, e.g. 2(1) trivial patterns for S9 indicates that S9 had observed 2 trivial patterns, but only one of them was correct.

Table 7.8 shows that the 10 students observed a total of 33 patterns, or an average of 3.3 patterns per student. The actual number of patterns observed by a student ranged from 2 to 6 patterns. Out of the 33 patterns, 6 + 3 = 9 are non-trivial (27%) and 14 + 10 = 24 are trivial (73%). A total of 20 patterns (or 61%) are related to the Types 1 and 2 patterns. The table also shows that the 10 students formulated 12 conjectures, or an average of 1.2 conjectures per student. The actual number of conjectures formed by a student ranged from 0 to 4 conjectures. Out of the 12 conjectures, 2 + 3 = 5 are non-trivial (42%) and 4 + 3 = 7 are trivial (58%). A total of 6 conjectures (or 50%) are related to the Types 1 and 2 patterns. S5 observed the highest number of patterns (which is 6) while S9 formulated the most number of conjectures (which is 4). S4 and S10 observed the least number of patterns (which is 2) while two students (S7,S8) did not formulate any conjecture at all.

Although Table 7.8 shows that there were 20 patterns and 6 conjectures *related* to the Types 1 and 2 patterns, most of these patterns and conjectures are not the *actual* Types 1 and 2 patterns for the Kaprekar task. For example, S1 initially thought that the consecutive sum of digits will repeat in this manner: 3, 6, 3, 6, 12, 6, 12, 15, 12, 15 (his investigation had been described in detail in Section 6.2d). Although this is related to the Type 1a pattern, the actual pattern is that all these sums of digits are divisible by 3 but not by 9. Only patterns and conjectures that are the actual Types 1 and 2 patterns are considered as non-trivial. From Table 7.8, it was observed that only 4 students (S3,S5,S8,S10) observed the non-trivial actual Type 1 or 2 patterns, and only 2 of them (S5,S10) went on to formulate it as a conjecture. Although S3 and S8 had observed that all the sums of digits or all the terms in some of the sequences that they had generated were divisible by 3, they did not see this as the actual pattern.

Instead, they kept trying to find how the sums of digits or the differences between consecutive terms would repeat, but they failed because this was not the actual pattern. S5 was the only student who observed the Type 1 ‘multiples’ patterns and she also proved the corresponding conjecture correctly (her investigation had been described in detail in Section 6.2a), while S10 was the only student who discovered the complicated Type 2 ‘digital roots’ pattern but she wrote down the corresponding conjecture wrongly when she was interrupted by the invigilator towards the end of the test (her investigation had been described in detail in Section 6.2c).

Let us now examine other types of patterns and conjectures that are not related to the Types 1 and 2 patterns. Table 7.8 shows that there were 13 patterns and 6 conjectures in this category, and most of them were false, except for 2 correct patterns and 2 correct conjectures from S9 who had misinterpreted the task and discovered patterns and formulated conjectures that were different from those of the original task (his investigation had been described in detail in Section 6.2b). For example, S2 observed that the last digits of consecutive terms in his Example 2 (Type 1b) were 9, 7, 5, 4, 3, 2, 1, 0, and then it would repeat (this will be called his Pattern 3). This pattern has nothing to do with the Type 1b patterns. In the end, he formulated it as Conjecture 1:

> “So there is a chance that this is what I am looking for. But then again it is only a theory. It needs to be proved [*sic*] throughout, ok. [Start writing] Common point: 0, 9, 7, 5, 4, 3, 2, 1, 0; ending digits of each number [stop writing].” [S2; Kaprekar Task]

Pattern 3 and Conjecture 1 of S2 were classified as non-trivial, but the pattern and conjecture were actually false. An example of a trivial pattern was from S6 who observed that the difference between two consecutive terms in her Example 1 is 10, followed by 4 terms, and then the pattern repeats (this will be called her Pattern 1) as shown in Figure 7.4. Although the student found a counter example in Example 1 (not shown in the figure) later to reject her Pattern 1, it is puzzling why she came back to her Pattern 1 again for her Example 3, which she also rejected after finding a counter example in Example 3 (not shown in the figure) later.

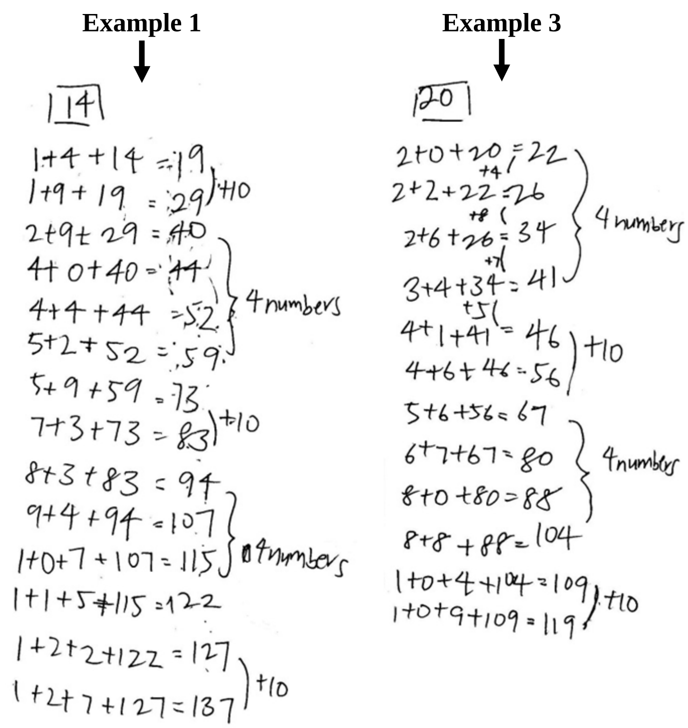


Figure 7.4 Trivial Pattern Observed by S6 for Kaprekar Task

There are other types of patterns not related to the Types 1 and 2 patterns, but they are also underlying patterns for the Kaprekar Task, e.g. all the sequences are increasing and the self numbers described in the task analysis of this task in Appendix E. However, none of the students discovered any of these other patterns.

(c) Conjecturing Processes vs. Outcomes

Table 7.9 shows the TPO for the conjecturing processes and outcomes for the 10 students' investigation of the Kaprekar Task. The issue is what had helped the students to observe the Types 1 and 2 patterns, so only patterns and conjectures related to these types are shown in the table. Since the conjecturing outcomes may also depend on the specialising outcomes, such as the number of examples generated and whether these examples were representative of the Types 1 and 2 sequences, there is also a need to include these specialising outcomes in the table.

Table 7.9 Conjecturing Processes and Outcomes for Kaprekar Task

	Specialising Outcomes		Conjecturing Processes****			Conjecturing Outcomes related to Types 1 and 2	
	No. of E.g.	*Rep. Examples	Searched for patterns in: Terms	Diff. b/w Terms	Others	Observed Patterns	Formulated Conjectures
S1	2	Type 1a,2		✓		2	1
S2	4	✓	✓	✓		2	
S3	3	Type 1a,2		✓		3(1)***	1
S4	4	Type 2		✓		2	1
S5	7	✓	✓	✓	✓	3(3)	1(1)
S6	6	Type 1b,2		✓		2	1
S7	7	✓		✓	✓	2	
S8	6	✓	✓	✓	✓	2(1)	
S9	5	NA**		✓		NA**	
S10	1	Type 2		✓	✓	2(1)	1(1)
Total	**45**	**4 rep.**	**3**	**10**	**4**	**20(6)**	**6(2)**

* The examples generated were considered representative if and only if all the three types of sequences were generated: Type 1a, Type 1b and Type 2. If otherwise, the type(s) generated will be specified.

** S9 misinterpreted the task, so his patterns were no longer the same as the original task. As a result, his examples could not be classified as representative because there was only one type of examples.

*** The number in brackets indicates the number of correct patterns or conjectures, e.g. 3(1) patterns for S3 indicates that S3 had observed 3 patterns related to the Types 1 and 2 patterns, but only one of them was correct. Since only actual Types 1 and 2 patterns or conjectures were correct, this also indicates that S3 had observed one actual Type 1 or 2 pattern.

**** The students did not analyse the feasibility of the plan (MF). Although some students exhibited metacognitive awareness (MA), it was not included in this table because it would be analysed in the checking stage in Section 7.2.6 since it had something to do with the students sensing something amiss and checking their working.

From Table 7.9, it was observed that the number of examples generated in the stage of specialising did not correspond to whether the students were able to observe the actual Types 1 and 2 patterns in the conjecturing stage. For example, S10 tried only one example but she found the complicated Type 2 ‘digital roots’ pattern as explained in part (b) of this section, while S7 generated the largest number of examples but she was unable to observe any actual pattern. Therefore, whether a student was able to observe the actual patterns did not depend on the number of examples generated. Similarly, whether the examples generated were representative of the Types 1 and 2 patterns did not correspond to whether the students were able to observe the actual patterns. For example, S2 tried 4 examples that were representative of all the Types 1 and 2 sequences, but he was unable to observe the actual patterns of these sequences. Of course, if the students did not generate any example that belonged to a particular type of sequences, they would not be able to observe the actual patterns of that type of sequences, e.g. S4 generated only 4 examples and all of them were Type 2 sequences, so he would not be able to observe any Type 1 pattern.

Since each example is a sequence, it is possible to generate very few examples but many terms for each example, e.g. S10 tried only one example but she generated 37 terms for the sequence, which is the highest number of terms for a sequence generated among the students for this task. Although S10 found the complicated Type 2 ‘digital roots’ pattern from this one example, S1 generated 36 terms for a Type 1 sequence but he was unable to find the actual patterns. In fact, 7 students (S1-S6,S10) generated more than 20 terms for at least one of the sequences that they investigated, but only 4 students (S3,S5,S8,S10) observed the actual patterns (as indicated by the number in brackets in the table), although 2 of them (S3,S8) did not recognise that their observed pattern is the actual pattern. This shows that the number of terms generated for a sequence did not correspond to whether the students were able to observe the actual patterns. On the other hand, if too few terms were generated, it might be difficult to observe any pattern, especially the complicated Type 2 ‘digital roots’ pattern.

It was further observed from Table 7.9 that all the students searched for patterns in the differences between consecutive terms, which are equal to the sums of the digits of the preceding terms, but only 4 students (S3,S5,S8,S10) were able to observe the actual patterns. Another place to search for patterns is the terms of a sequence because all the terms are divisible by 3 or 9 for Type 1 sequences, but otherwise for Type 2 sequences (Types 1 and 2 ‘multiples’ patterns). Table 7.9 shows that 3 students (S2,S5,S8) tried to find a pattern in the terms of a sequence, but only S5 was able to discover the actual Type 1 ‘multiples’ patterns. Although S8 also observed that all the terms in one of her sequences are multiples of 3, she did not recognise that this is the actual pattern. This shows that searching for patterns in the correct places might not correspond to whether the students were able to observe the actual patterns.

The issue now is to examine what had helped S5 to discover the Type 1 ‘multiples’ patterns and S10 to discover the Type 2 ‘digital roots’ patterns. The Secondary 2 students in the present research had studied some number patterns in Secondary 1, but the usual patterns that they had learnt were very different from the Types 1 and 2 patterns of the Kaprekar Task. The following lists some examples of sequences that the students had learnt in Secondary 1.

Linear: 1, 4, 7, 10, 13, … (next term obtained from previous term by adding 3)

Perfect Squares: 1, 4, 9, 16, 25, …

Perfect Cubes: 1, 8, 27, 64, 125, …

Powers: 3, 9, 27, 81, 243, …

Two special cases of a linear sequence are multiples of 3 and multiples of 9:

Multiples of 3: 3, 6, 9, 12, 15, … (next term obtained from previous term by adding 3)

Multiples of 9: 9, 18, 27, 36, 45, … (next term obtained from previous term by adding 9)

But the multiples of 3 and multiples of 9 sequences are in a fixed increasing order, i.e. the next term is obtained from the previous term by adding a constant. This means that there is a relationship between the term T_n and its position n, i.e. it is possible to find a formula for the n-th term or general term of the sequence, e.g. $T_n = 3n$ for the multiples of 3 sequence. This may explain why the students in this study searched for differences in consecutive terms of the sequences. However, the Types 1 and 2 patterns for the Kaprekar Task are different from what the students had learnt. For example, consider the Type 1a sequence generated by S1 for his Example 1 below:

21, 24, 30, 33, 39, 51, 57, 69, 84, 96, …

Although these are multiples of 3, they are not in a fixed increasing order, i.e. the next term is not obtained from the previous term by adding a constant, and so there is no simple relationship between the term T_n and its position n. This means that it is not possible to find a simple formula for the n-th term in terms of its position n. In fact, the numbers to add to the previous terms to obtain the next terms (or the differences between consecutive terms, which are also equal to the sums of the digits of the preceding terms for the Kaprekar task) are also multiples of 3 as shown below:

3, 6, 3, 6, 12, 6, 12, 15, 12, …

However, there is also no relationship between these numbers and their positions. Therefore, some of the students (S3,S8) did not treat the Type 1a ‘multiples’ pattern as the actual pattern since this kind of patterns is not the usual types of patterns that they had learnt before. Instead, just like most of the other students, they tried to find a fixed increasing pattern or a repeating pattern for consecutive sums of digits or the differences between consecutive terms. Although the students might not have learnt the idea of a repeating pattern in their normal school lessons before, they had encountered repeating patterns during the familiarisation lesson where the last digits of powers of 3 repeat with a period of 4, and in Pretest Task 1 where the sad numbers enter into a loop (see outlines of Lessons 1 and 2 in Appendix C).

From the protocols of the only student (S5) who discovered the Type 1 ‘multiples’ patterns, it was observed that she was able to apply the divisibility tests for 3 and 9 to prove that the Type 1 ‘multiples’ patterns are always true for Type 1 sequences. Therefore, knowing the divisibility tests for 3 and 9 might have helped S5 to discover the Type 1 ‘multiples’ patterns for the Kaprekar Task. Although the divisibility tests are not in the secondary school mathematics syllabus, it is possible that some students in the present study might have learnt them as an enrichment, especially when the divisibility test for 3 is an efficient way to check whether a number is divisible by 3, and if a number greater than 3 is divisible by 3, then it cannot be a prime number (prime numbers are in the Secondary 1 syllabus). However, knowing the divisibility tests for 3 and 9, and knowing when to apply them in situation that calls for them, are two different things.

Similarly, the Type 2 'digital roots' pattern discovered by only one student (S10) is also not the usual types of patterns that the students had learnt before. S10 took quite a long while to observe that the differences between consecutive terms actually follow a basic 10-2-4-8-7-5 repeating pattern[14], where the exceptions, 19, 11, 13, 17, 16 and 14, are different from the corresponding numbers in the basic pattern by 9 as shown below (the reader should refer to her full investigation described in Section 6.2(c) on page 237 to understand the following).

$$19 = 10 + 9,$$
$$11 = 2 + 9,$$
$$13 = 4 + 9,$$
$$17 = 8 + 9,$$
$$16 = 7 + 9,$$
$$14 = 5 + 9.$$

This is actually another way of looking at the Type 2 'digital roots' pattern (the reader should refer to the task analysis in Appendix E on page 500 to be familiar with the different patterns for this task). But the student did not see this as the actual pattern yet. About 20 minutes into the test, she decided to organise her working by listing all the first differences between consecutive terms in one line so that it was easier to search for patterns. She then looked for patterns in the differences between consecutive first differences as shown in Figure 6.9 on page 237, and observed that the absolute second differences are 1, 2, 5, 1 and 7 respectively, as illustrated below (since she did not write down her observation but she just verbalised it):

[14] Actually, the basic pattern is the 1-2-4-8-7-5 repeating pattern where 10 = 1 + 9, but the student had not obtained the number 1 before, so she was unable to make such an observation.

```
28,    38,    49,    62,    70,    77,    91  ← original sequence
  \   /  \   /  \   /  \   /  \   /  \   /
   10      11     13     8      7     14     ← first differences
     \   /   \   /  \   /  \   /  \   /
       1       2      5      1      7        ← absolute second differences
```

This shows that the student knew how to search for patterns in the second differences, although this is not part of the secondary school mathematics syllabus. It is possible that the student might have learnt it before during some enrichment, or she might have just extended what she knew about searching for patterns in the first differences to searching for patterns in the second differences. Whatever the case, this suggests how she might have discovered her basic 10-2-4-8-7-5 repeating pattern in the first place: she might have tried to find the difference between the first differences 10 and 19, which are *not* consecutive first differences, and observed that 19 – 10 = 9. This is akin to finding the absolute second differences above, although the second differences are the differences between *consecutive* first differences.

In the literature review of Schoenfeld's problem-solving model in Section 2.2.2(d), Schoenfeld (1985) had discovered that some of his students were able to actively make use of their mathematical knowledge, which he called 'resources', to solve mathematical problems. Similarly, it was found in the above data analysis that the students in the present study only had the knowledge of the usual types of patterns that they had learnt in Secondary 1, and they lacked the knowledge of other types of patterns that were critical to the discovery of the Types 1 and 2 patterns for the Kaprekar Task. Only one student (S10) was able to apply what she had learnt to search for patterns in the second differences which resulted in her discovering the Type 2 'digital roots' pattern, while the other student (S5) was able to associate the divisibility tests for 3 and 9 with the sums of digits in the Kaprekar Task which resulted in her discovering the Type 1 'multiples' pattern. In other words, Schoenfeld's idea of an active application of resources had played an important role in helping the two students to observe the actual patterns.

Table 7.9 shows that 4 students (S5,S7,S8,S10) also searched for patterns in other places. As discussed above, S10 looked for patterns in the absolute second differences of her Type 2 sequence. Interestingly, S5 also searched for patterns in the absolute second differences of her Type 1a sequence[15], but she did not find any. The third student (S7) tried to search for patterns in the differences between the tens digit of consecutive terms, and the differences between the ones digit of consecutive terms, as shown in Figure 7.5. She observed that the differences between the tens digit of consecutive terms revolved around 0 and 1, but she was unable to proceed further.

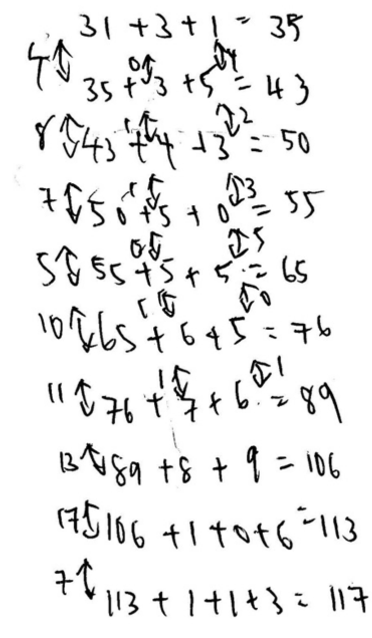


Figure 7.5 Searching for Pattern by S7 for Kaprekar Task

[15] Actually, S5 searched for patterns in the absolute differences between consecutive sums of digits. But consecutive sums of digits are actually equal to the differences between consecutive terms of a Kaprekar sequence. Therefore, the absolute differences between consecutive sums of digits are equal to the absolute second differences of the Kaprekar sequence.

The fourth student (S8) tried to find a pattern in the differences between consecutive odd terms of a sequence as shown in Figure 7.6. She observed that the first two differences are 12, but she soon found a counter example in the third difference 21.

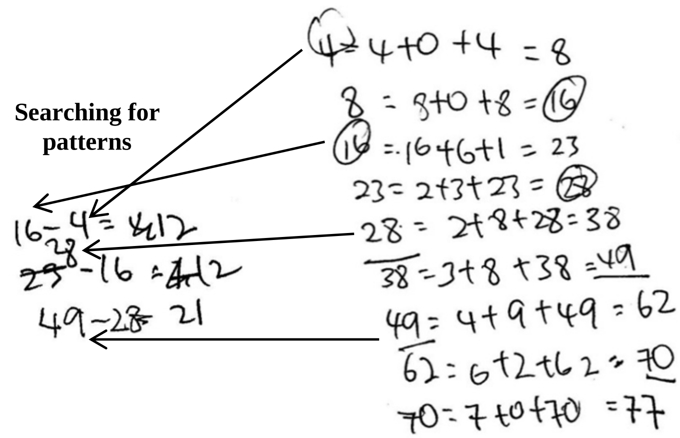


Figure 7.6 Searching for Pattern by S8 for Kaprekar Task

For the conjecturing stage, the students were supposed to analyse the feasibility of their plan to search for patterns (MF). An example of such a plan is the plan to search for patterns within a sequence or the plan to search for patterns across sequences. But the students in the present study just went ahead to search for patterns within a sequence without analysing the feasibility of such a plan. As a result, none of them searched for patterns across sequences. Although the students did search for patterns by referring to different sequences, they were still searching for patterns *within* each of these sequences, such as the differences between consecutive terms. If the students had searched for patterns *across* sequences, they might have discovered the self numbers described in the task analysis of this task in Appendix E. Self numbers, such as 1, 3, 5, 7, 9, 20 and 31, will never appear as the second or subsequent terms of any Kaprekar sequence. These self numbers could only be discovered if the students compared the terms across sequences.

Summary

To summarise, new investigation outcomes and new interactions among the processes were discovered from analysing the empirical data obtained in the present study, so these new outcomes and interactions were incorporated in the refined investigation model for cognitive processes. It was also found that generating all the representative examples and searching for patterns in the correct places did not help most of the students to observe the Types 1 and 2 patterns of the Kaprekar Task. This was because these kinds of patterns were not the usual types of patterns that they had learnt during their normal school lessons. However, two students were able to actively apply their knowledge, or what Schoenfeld (1985) called ‘resources’, to discover the Type 1a ‘multiples’ pattern or the Type 2 ‘digital roots’ pattern. Moreover, all the students did not analyse the feasibility of their plan to search for patterns, so none of them searched for patterns across sequences and thus they were unable to observe patterns that exist across sequences, e.g. the self numbers.

7.2.5 Justifying and Generalising (Stages 5 and 6)

Based on the literature review in Section 2.2.3(e) on page 75, some interesting issues for the present study to examine are: (i) Do students justify their conjectures, or do they wrongly accept their observed patterns as true without testing or based on naïve testing? (ii) Do students justify their conjectures by using a non-proof argument or a formal proof? Table 7.10 shows the TPO for the justifying and generalising processes and outcomes for the 10 students’ investigation of the Kaprekar Task.

Table 7.10 Justifying / Generalising Processes and Outcomes for Kaprekar Task

	Conjecture	Justifying Processes			Justifying and Generalising Outcomes
		Naïve Testing	Non-proof Argument	Formal Proof	
S1	Wrong Trivial Conjecture 1				Wrongly accepted conjecture as true without testing
S2	Wrong Non-trivial Conjecture 1	✓			Refuted conjecture based on naïve testing
S3	Wrong Trivial Conjecture 1				Test ended after formulating conjecture
S4	Wrong Trivial Conjecture 1				Test ended after formulating conjecture
S5	Correct Non-trivial Conjecture 1		✓		**Proven conjecture correctly (generalisation)**
	Wrong Trivial Conjecture 2	✓		✓	Refuted conjecture based on naïve testing
S6	Wrong Trivial Conjecture 1				Wrongly accepted conjecture as true without testing
S9**	Wrong Trivial Conjecture 1	✓			Refuted conjecture based on naïve testing
	Correct Trivial Conjecture 2		✓		**Proven conjecture correctly (generalisation)**
	Correct Non-trivial Conjecture 3	✓	✓		**Proven conjecture correctly (generalisation)**
	Wrong Non-trivial Conjecture 4		✓		Test ended during justifying
S10	Correct Non-trivial Conjecture 1				Test ended after formulating conjecture
Total	**12**	**4**	**4**	**1**	**3 proven; 3 generalisation***

* There was no generalisation for all the conjectures except for the three proven conjectures.
** S7 and S8 were omitted from the table because they did not formulate any conjecture to justify; S9 misinterpreted the task, so his conjectures were no longer the same as the original task.
*** The students did not exhibit the metacognitive behaviours, MF and MA, in this stage.

It was observed from Table 7.10 that only 8 students (except S7 and S8) had formulated at least one conjecture, but 2 of them (S1, S6) wrongly accepted their conjecture as true without testing. For 3 of the students (S3,S4,S10), the test ended just after they had formulated their conjecture and it could not be inferred from their protocols whether they would accept their conjecture as true without testing. The remaining 3 students (S2,S5,S9) understood that they needed to justify their conjecture(s). All 3 of them tried naïve testing, which was advocated by Lakatos (1976) to refute a conjecture based on counter examples. For S2, he refuted his only conjecture based on naïve testing. For S5 and S9, they both refuted one of their conjectures based on naïve testing. None of the students wrongly accepted their conjectures as true based on naïve testing.

Table 7.10 shows that 2 of the students (S5,S9) also tried to justify their conjecture(s) using a non-proof argument. For S5, she correctly proved her Conjecture 1 on the Type 1a 'multiples' pattern, which was described in detail in Section 6.2(a), by using the divisibility tests for 3 and 9 (see her proof in Figure 6.3 on page 229). For S9, he misinterpreted the task and so his conjectures were different from those of the original task. However, he was able to use the underlying structure to correctly prove two of his conjectures, one of which was a non-trivial one (see his proofs in Figure 6.5 and Figure 6.7 in Section 6.2b). But for his last conjecture, he made a mistake and so his non-trivial conjecture was wrong. Again he tried to use the underlying structure to prove the conjecture, but the test ended while he was still in the process of justifying. Only one student (S5) attempted to use a formal proof involving algebra. The following shows her protocols when she formulated Conjecture 2 verbally and then tried to use algebra to prove the conjecture but failed:

> "So if the start number is … odd number, then the final number also be [*sic*] odd number ... If the … start number is even number [write: even number] even number, the rest will be even number ... But can I prove this finding? ... Can I use algebra to try … this example? ... Hmm, the two digits are a and b [write: $\overline{ab}$] ... a and b, so now I use $\overline{ab} + a + b$ ... so $\overline{ab}$ [continue writing] $+ a + b = ...\ 11a + 2b$ [stop writing] ... Here what I want is that ... $a + ...\ a + ...\ b = ...$ no, $a + ...$ Oh, cannot use algebra [cancel her algebra working]." [S5; Kaprekar Task]

None of the students in the present study analysed the feasibility of their plan to justify (MF). If S5 had done so, she would have realised that she could have easily referred to her previous five examples, which also involved an odd or an even starting number, to see if she could find any counter example to refute her Conjecture 2 (naïve testing). In fact, there were many counter examples in her previous five examples which she could have used, without even resorting to the use of algebra. In the end, because she did not know how to prove her conjecture, she tried another example (naïve testing) and soon found a counter example to refute the conjecture.

Table 7.10 shows that only 3 of the 12 conjectures were correctly proven by 2 students (S5,S9). Since these conjectures were general results, the proven conjectures were generalisations. As posited in the theoretical investigation model for cognitive processes described in Section 3.2.1, the issue in the generalising stage is whether a proven conjecture will lead to a generalisation.

Summary

To summarise, 2 of the 10 students still wrongly accepted their conjecture as true without testing, even after the teaching experiment, but none of the students wrongly accepted their conjectures as true based on naïve testing. Only 3 students attempted to justify their conjecture(s). All 3 of them attempted naïve testing and refuted their conjecture(s) based on counter examples, while 2 of them managed to prove their conjecture(s) using a non-proof argument, leading to generalisation. Only one student tried a formal proof using algebra but failed. It seems that formal proofs are beyond the level of Secondary 2 students. None of the students analysed the feasibility of their plan to justify (MF) nor exhibited any metacognitive awareness (MA) in this stage.

7.2.6 Checking (Stage 7)

Table 7.11 shows the TPO for the checking processes and outcomes for the 10 students' investigation of the Kaprekar Task. As explained in the theoretical model for cognitive processes in Section 3.2.1, students should check their working (coded as CW in the protocols) occasionally in all stages and not wait until the Checking (R) stage. However, it is more appropriate to analyse all the checking processes and outcomes together in the same section. Similarly, monitoring progress (MP), and metacognitive awareness (MA) where students sensed something amiss[16], will also be analysed in this section although they could occur in previous stages. Reviewing the solution to see if it has met the goal of the task (MR) is a metacognitive process which includes examining whether the answer is reasonable or logical, evaluating the efficacy of a method of solution, and looking for an alternative method. The errors made (EM) are classified as major or minor. Major errors are defined as those that will have a serious impact on the investigation, e.g. mistakes that change the underlying patterns so that the students are not able to discover the correct patterns. But if the students discover these errors almost immediately afterwards, then these errors will be considered as minor because the impact on the investigation will not be so great.

[16] Metacognitive awareness (MA) that helped students to be aware of what they were doing, which might have helped them save time by not repeating what they had done before, were dealt with in the stages that it had occurred, e.g. in Section 7.2.3 for the Kaprekar Task.

Table 7.11 Monitoring / Checking Processes and Outcomes for Kaprekar Task

	Processes							Outcomes	
	Check Working				MP	MR	MA	Errors Made*	Errors Discovered
	Most Parts	Some Parts	Glance Briefly	Total					
S1	2	2		**4**			1	1 minor	1 minor
S2				**0**	4		1	3 major + 1 minor = 4	1 minor
S3	1		2	**3**	4			2 major + 2 minor = 4	1 major (20 min later**) + 1 minor = 2
S4	3			**3**				4 major + 1 minor = 5	1 major (7 min later) + 1 minor = 2
S5			2	**2**	5	2	2	3 major + 3 minor = 6	1 major (8 min later) + 1 minor = 2
S6		3		**3**			1	1 major + 4 minor = 5	2 minor
S7		1		**1**	11		1	3 minor	1 minor
S8			1	**1**	10			1 major	0
S9			1	**1**	5	2	2	1 major + 7 minor = 8	1 major (1 min later) + 5 minor = 6
S10	1	1	1	**3**			1	1 major + 1 minor = 2	1 minor
Total	**7**	**7**	**7**	**21**	**39**	**4**	**9**	**16 major + 23 minor = 39**	**6 major + 14 minor = 20**

* Errors due to misinterpretation had been dealt with in Table 7.2, and errors due to accepting conjectures as true without testing or based on naïve testing had been dealt with in Table 7.10, so these errors were omitted from this table. Observing incorrect patterns or formulating incorrect conjectures, as discussed in Table 7.8, were *not* errors.

** The time indicated in brackets for 'Errors Discovered' refers to the time interval between making the major error and discovering the error. No time is indicated for the discovery of a minor mistake because the discovery time is not an important factor when the mistake is minor.

Table 7.11 shows that 9 students checked their working on 21 occasions, or between 1 to 4 times each. Only S2 did not appear to check his working at all. From their protocols, it was observed that only S5 checked all her working in the checking stage of the review phase, while S9 checked his working once during the transition between the conjecturing and justifying stages. Since all the other students did not reach the checking stage, all their checking was done in the specialising and conjecturing stages when they were trying examples and searching for patterns. The frequencies for the three processes of checking happened to be the same: 7 times for checking working step by step for most parts of an example (or the entire example), 7 times for checking working step by step for some parts of an example, and 7 times for checking working by just glancing through it briefly. None of the students checked their working by other means, such as working backwards.

The issue is which checking processes were effective in helping the students discover their mistakes. A detailed analysis of their protocols suggests that metacognitive awareness (MA), which includes the ability to sense something amiss, had played a more important role than any of the checking processes in helping the students to discover their errors. For example, S1 obtained a 9 for the sum of the digits of the number 234 in his Example 1, and he immediately sensed something amiss because all the previous sums of digits were either 3, 6, 12 or 15, but never 9. He asked, "How come is 9?" Then he recalculated only some parts of his working (i.e. the previous step) and discovered the error very soon: it should not be 234 but 237, whose sum of digits is 12. This calculation error was considered minor because he discovered it almost immediately afterwards. Another example is when S2 obtained 16 for the sum of the digits of the number 439 in his Example 2, he immediately realised that something was amiss because all the previous sums of digits were either 9 or 18. The following shows his protocols:

> "[Continue writing] 432 + 9 = 441; 439 + 16 [stop writing]. Then why do you get 16? Oh, 441. Error … [cancel previous 2 lines] [start writing] 432 + 9 = 441; 441 + 9 = 450." [S2; Kaprekar Task]

The protocols show that he discovered his mistake because he sensed that something was amiss when he obtained a questionable result. These two vignettes (S1,S2) also suggest that it is not necessary to check all the working step by step all the time, which is very time consuming, if the students are able to sense something amiss when they obtain a questionable result.

Let us now look at a counter example. S4 made two calculation mistakes in his Example 1 that changed the patterns, which prevented him from observing the actual patterns. When he was stuck, he paused for 12 seconds while flipping through the two pages of his answer script. Since he did not say anything during the pause, he may have been checking his working by glancing through it, because he soon discovered the first major mistake, though this is unclear. Then he redid Example 1 from the beginning, so the second mistake in his original example was no longer relevant. Probably because of his earlier mistake, he decided to pause on three separate occasions to check most parts of his working, but his working was correct. However, towards the end of the test, he had stopped checking his working altogether, probably due to the time constraint. This was when he made a calculation error in his Example 4 that changed the pattern as shown in Figure 7.7. But he did not sense anything amiss when the sums of digits had changed drastically from the 5-10-11-13-17-7 pattern to the 9-18 pattern. He even concluded that the sums of digits from 153 onwards were mostly 9 with an exception of 18. Thus the lack of metacognitive awareness had resulted in the student checking his working at the wrong places.

Most of the minor mistakes were calculation errors which the students discovered soon enough (within one minute). However, most of the students did not explicitly check their working before spotting these errors, although it is possible that the students might have glanced through the previous step of their working.

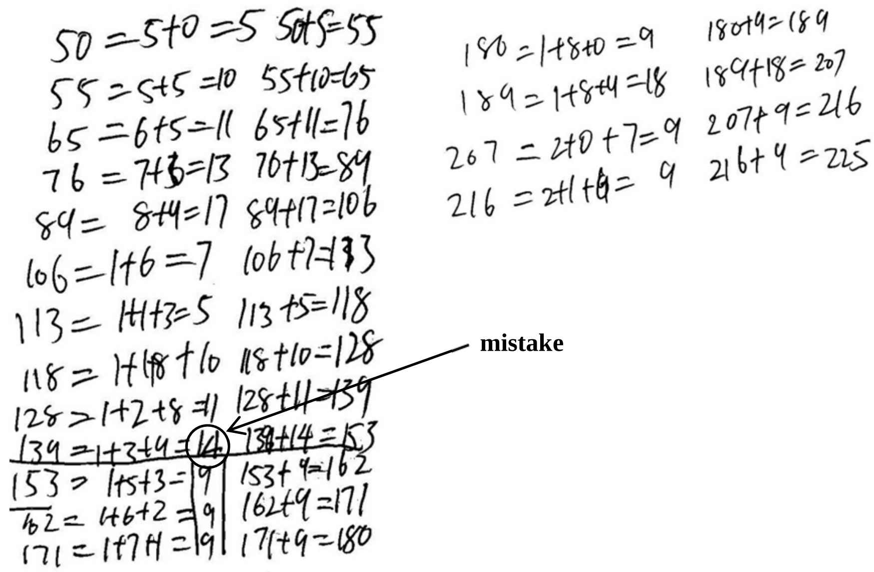


Figure 7.7 Failure to Spot Mistake by S4 for Kaprekar Task

Table 7.11 shows that only 2 students (S5,S9) reviewed their solution (MR) since both of them were the only ones who had justified a conjecture and so they should review if the solution had met the goal of the task. The first student (S5) reviewed her solution twice by glancing through her working briefly to check that her proven Conjecture 1 was only true for some of her examples, but she acknowledged that she was unable to find a pattern for the other examples. Then she concluded that this was the result of her investigation. The second student (S9) reviewed his solution after proving his Conjecture 2 and realised that the conjecture was not relevant (actually, the conjecture was just very trivial). The following shows his protocols:

> "[Flip back to p. 1] Ok, I am referring back … All 2-digit numbers … which [*sic*] tens place is odd will have a new number that is odd [flip back to p. 2] … Ok, that seems not the … real task is wanting me to find. [Flip back to p. 1] So … the task is wanting me to find … the pattern. Ok … so I think that my second conjecture is a little bit off topic." [S9; Kaprekar Task]

S9 reviewed his solution the second time after proving his Conjecture 3 and realised that he had "answered the question to a certain extent" (in his own words). Then he discovered that he did not repeat the process for the new number, a misinterpretation error discussed earlier in Section 7.2.1 under the understanding stage. This suggests that both students were able to review their solution effectively when they realised that they had not fully met the goal of the task. However, both students did not engage in other aspects of reviewing their solution: they did not examine whether their solution was reasonable or logical, nor did they evaluate the efficacy of their method of solution or look for an alternative method.

Table 7.11 shows that only 6 students exhibited overt signs of monitoring progress (MP) during their investigation for a total of 39 times. Two students (S7,S8) monitored their progress the most frequently on 10 to 11 occasions each, but that does not mean that they were able to produce significant outcomes, such as non-trivial patterns or conjectures. On the contrary, it was precisely because they were stuck that they monitored their progress so often, but it was not effective. The following shows some of the protocols of S7 when she was monitoring her progress:

> "I think I'm going on the wrong track … I think I shall search for a new pattern instead."
>
> "Never mind, I think I, I shall see if … I think I try to re-read the question …"
>
> "I think I'm going on the wrong track ... My answer don't [*sic*] seem logical ... Never mind, I continue."
>
> "I think I might … Am I going on the right track? ... *I don't think I know* [emphasis mine] … Maybe I'll try for another, for a while more … and see how it goes …"
>
> "I think I'm going nowhere … Should I think of … a *new approach* [emphasis mine] instead? … Never mind, I think I try another number …"
>
> [S7; Kaprekar Task]

Her second last protocol sums up the main problem: she realised that there was no way for her to know whether she was going on the right track. Thus the only thing that she could do was to keep going in the same direction for a while or to change approach. But she did not know of any new approach, so she continued trying another example, as shown in her last protocol above. On the other hand, one student (S3) decided to check his working while monitoring his progress when he was stuck. The following shows his protocols:

> "I'm stuck. I've no idea what I'm looking for. Choose any number. Add the sum of digits to the number itself. Investigate. Ok, 6, 12, ok, maybe I've some calculation error, so maybe I should just check." [S3; Kaprekar Task]

However, he only glanced through his working on two occasions without doing any recalculation, so he did not discover the two calculation errors that changed the patterns in his Examples 1 and 2, which prevented him from discovering the actual patterns. It was only 20 minutes after making the second major error that he decided to check his working step by step for most parts of his Example 1, which helped him to discover that mistake.

Another student (S5) monitored her progress when she was stuck by referring to the given checklist of investigation processes (see Appendix E) to find out what she could do to observe a pattern. When she read 'Pose a related problem' in the checklist, she changed her approach from looking for patterns in consecutive sums of digits (her Specific Problem 1) to looking for patterns in the differences between consecutive sums of digits (her Specific Problem 2). When she listed these differences, which were either 3 or 6, she realised something was amiss when she obtained 5, which enabled her to discover one of her major errors (her investigation had been described in detail in Section 6.2a). Since she was able to progress after discovering this mistake at about 11 minutes into the test, she did not appear to monitor her progress for the rest of her investigation.

Summary

To summarise, metacognitive awareness (MA), which includes the ability to sense something amiss when a questionable result is obtained, had played a more important role than the checking processes in helping the students to discover their mistakes. Only two students reviewed their solution, and they were able to review effectively when they realised that their solution had not fully met the goal of the task. But none of the students evaluated the efficacy of their method of solution or looked for an alternative method during the review of their solution. It was also found that most of the students monitored their progress only when they were stuck. Some of them did not know what else to do when monitoring their progress, except to continue in the same direction, or to 'change approach' by trying a new example, which is not very different from continuing in the same direction; while others were able to monitor their progress more effectively by changing the approach to search for patterns elsewhere or by checking their working to see if they had made any mistake.

7.3 MATHEMATICAL INVESTIGATION PROCESSES AND OUTCOMES FOR TYPE B TASKS

In this section, the 10 sets of thinking-aloud protocols and answer scripts for the Sausage Task (Type B) will be analysed to study the effect of the 10 students' processes on their investigation outcomes. The Sausage Task is reproduced below.

> ***Posttest Investigative Task 2: Sausages***
>
> I need to cut 12 identical sausages so that I can share them equally among 18 people. Investigate.

Table 7.12 shows the processes and outcomes for each investigation stage for the Sausage Task. As posited in the theoretical investigation model for cognitive processes in Section 3.2.1, some of the processes and outcomes for Type B tasks are different from those for Type A tasks. For example, the students were expected to extend Type B tasks, and so Stage 8 (Extension) is included in the table together with Stage 2 (Problem Posing) since both stages involve posing problems. Similarly, the students were expected to use other heuristics in Stage 3 to solve the problems that they had posed, and they would also have to specialise during extension in order to generalise, so Stage 3 in the table includes both Specialising and Using Other Heuristics. The classification of an outcome (problem posed or conjecture formulated) as trivial or non-trivial had passed the inter-coder reliability test (see Section 5.4).

Table 7.12 Investigation Processes and Outcomes for Type B Tasks

Stage	Cognitive Processes	Metacognitive Processes	Outcomes
Stage 1: Understanding the Task (U)	• Re-reading task (RR) • Rephrasing task (RT) • Highlighting key information (HI) • Visualising Information (VI)	• Monitoring understanding (MU)	• Understood task correctly • Misinterpreted task • Recovered from misinterpretation
Stages 2: Problem Posing (P) and Extension (E)	Referring to following to think of problem to pose: • task statement • current working • previous result • given checklist	• Analysing feasibility of goal or problem posed (MG)	• Posed trivial specific problems • Posed non-trivial specific problems
Stage 3: Specialising (S) and Using Other Heuristics (H)	• Random specialising • Purposeful specialising • Systematic specialising • Using reasoning • Using algebra	• Analysing feasibility of plan to specialise or use other heuristics (MF) • Metacognitive awareness (MA)	• Formulated correct conjecture to generalise • Used other heuristics effectively, quite effectively or ineffectively
Stage 4: Conjecturing (C)	• Using reasoning • Using algebra	• Analysing feasibility of plan to formulate conjecture (MF) • Metacognitive awareness (MA)	• Formulated trivial or non-trivial conjectures
Stages 5 / 6: Justifying / Generalising (J/G)	• Naïve testing to refute conjecture • Using a non-proof argument • Using a formal proof	• Analysing feasibility of plan to justify conjectures (MF) • Metacognitive awareness (MA)	• Proved conjecture leading to generalisation • Proved conjecture but did not lead to generalisation
Stage 7: Checking (R)	Checking correctness of working (CW): • step by step for most parts • step by step for some parts only • glancing through it briefly • working backwards	• Monitoring progress[17] (MP) • Reviewing solution to see if it had achieved the goal (MR) • Metacognitive awareness (MA)	• Discovered major errors on time • Discovered major errors late • Did not discover major errors at all

[17] Since monitoring progress can occur in any stage, and it does not matter at which stage the students monitor their progress (unlike metacognitive awareness), it will be more appropriate to analyse this process for the entire investigation together with the processes in the checking stage.

7.3.1 Understanding the Task (Stage 1)

Table 7.13 shows the TPO for the understanding processes and outcomes for the 10 students' investigation of the Sausage Task. It was observed that 2 of the 10 students (S1,S8) did not appear to re-read (RR) or rephrase the task (RT), highlight key information (HI), visualise information (VI) or monitor their understanding (MU). The other 8 students engaged in the understanding processes from 1 to 11 instances each. RR, RT and VI occurred a lot more frequently than HI and MU. Only 4 students engaged in MU, which has the lowest total frequency at 6. A total of 3 students misinterpreted the task (same as the Kaprekar Task described in Section 7.2.1), out of which 2 of them recovered after about 4 minutes and 8 minutes into the investigation. Unlike the Kaprekar Task (Type A), there was no need for the students to try examples (TE) to understand the Sausage Task (Type B) but they could visualise the given information (VI) by drawing a diagram.

Table 7.13 Understanding Processes and Outcomes for Sausage Task

	Processes						Outcomes	
	RR	**RT**	**HI**	**VI**	**MU**	**Total**	**Understood**	**Misinterpreted**
S1						**0**	✓	
S2				1		**1**	✓	
S3	3+1=4*	1+1=2	1	0+2=2	1+1=2	**11**		Recovered after 4 min
S4		1	1			**2**	✓	
S5	2	2	1	3		**8**		Recovered after 8 min
S6	1		1	1		**3**		Did not recover
S7	2	2		2	1	**7**	✓	
S8						**0**	✓	
S9		3	3		1	**7**	✓	
S10	1+1=2	1+2=3		0+2=2	1+1=2	**9**	✓	
Total	**11**	**13**	**7**	**11**	**6**	**48**	**7**	**3 misinterpreted; 2 recovered**

* S3 engaged in RR for 3 + 1 = 4 times means that there were 2 episodes of understanding the task and RR happened 3 times in the first episode and 1 time in the second episode; 1 × HI for S3 means that HI happened 1 time in the first episode.

The issue was whether the understanding processes used by the students had helped them to interpret the task correctly or to recover from any misinterpretation, and how did these processes help. From the detailed protocol analysis of the 3 students (S3,S5, S6) who misinterpreted the task, it was observed that the understanding processes did not help them to interpret the task correctly, but re-reading the relevant parts of the task had helped 2 of them (S3,S5) to recover from their misinterpretation.

For example, S5 engaged in 8 instances of understanding processes during the Understanding (U) stage, but she still misinterpreted the task when she rephrased the first part of the task statement wrongly: "I want to cut them into 12 identical sausages." What she meant was that she wanted to cut one sausage into 12 identical parts to share them among the 18 people, so she wrote down her problem: "Problem: How the 18 people share 12 pieces of sausages? Is this method working?" But then she was stuck when she realised that 12 pieces of sausages were not enough to share among the 18 people. She even suggested getting another 6 pieces so that there would be enough to share among the 18 people! At about 8 minutes into the investigation, she re-read the entire task statement to think of how to solve the problem (different from re-reading to understand the task) and paused for 4 seconds. Then she cancelled '12 pieces of sausages' in her problem stated above and wrote '12 identical sausages'. Her subsequent working shows that she had found her mistake from re-reading the relevant parts of the task on '12 identical sausages'. Similarly, the other student (S3) did not realise that he had to cut the sausages despite engaging in the understanding processes 11 times, but he managed to recover from his misinterpretation at about 4 minutes into the test when he re-read the relevant parts of the task 'I need to cut' while thinking of a problem to pose (different from re-reading to understand the task).

For the last student (S6) who misinterpreted the task but did not recover, it was a totally different problem. She had no issue with dividing a sausage with rounded ends into 3 equal parts, but she believed that each person should receive an equal number of rounded ends and an equal number of middle portions. Figure 7.8 shows that she named the 3 parts of a sausage ‘a’, ‘b’ and ‘c’, and there were a total of 12 ‘a’ and 24 ‘b & c’ for the 12 sausages. In other words, she treated the rounded ends as different from the middle portions ‘a’, but the rounded ends ‘b’ and ‘c’ as the same. However, this problem cannot be solved unless one subscribes to the hidden assumption that the shape of the sausages does not matter, as explained in the task analysis of this task in Appendix E. It is possible that the student had confused the word ‘identical’ in the ‘12 identical sausages’ in the task statement with the parts of the sausages to be shared equally with the 18 people: ‘equal’ parts for the 18 people means equal in volume, not necessarily ‘identical’ in shape. This possibility was inferred when she later cut each sausage into 2 parts and then said that the 2 parts are “identical” (in her own word).

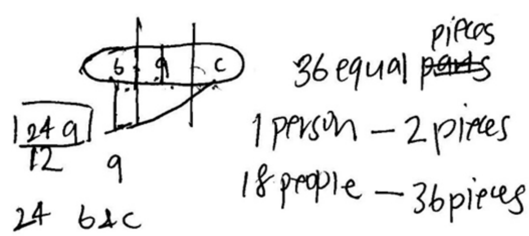


Figure 7.8 Misinterpretation of Task by S6 for Sausage Task

Let us now examine what might have helped the other 7 students understand the task correctly. Three of them (S7,S9,S10) engaged in 7 to 9 instances of understanding processes each. One of them (S7) specifically mentioned that drawing a diagram was “a simpler way to understand” (in her own words). Thus these processes might have helped the 3 students to interpret the task correctly. However, it was possible to engage in very few instances of these processes to understand the task correctly since the other 4 students (S1,S2,S4,S8) engaged in 0 to 2 instances of these processes each. For example, S1 read the task once and immediately posed a problem, suggesting that it is possible to read the task once and understood the task correctly without engaging in any of these processes.

Summary

To summarise, the understanding processes had helped some students in the present study to interpret the task correctly, but these processes did not prevent 3 students from misinterpreting the task. Nevertheless, one of these processes had helped 2 of them to recover from their misinterpretation when they re-read the *relevant* parts of the task (same as the Kaprekar Task discussed in Section 7.2.1). On the other hand, it is possible for some students to interpret the Sausage Task correctly by reading the task only once, which is not possible for the Kaprekar Task since the students need to try examples to understand the latter, which will include re-reading the task for the instructions. That is why the students engaged in the understanding processes for a total of 90 times for the Kaprekar Task, which is a lot more than the 48 times for the Sausage Task.

7.3.2 Problem Posing (Stage 2)

The treatment for the data analysis in this section will be different from the previous section as there is a need to analyse in detail the types of specific problems posed by the 10 students for the Sausage Task before the empirical data could be examined to study the effect of the problem-posing processes on the outcomes in this stage.

(a) Problem-Posing Outcomes

During the task analysis of the Sausage Task in Appendix E, some possible problems that students could pose for this task were identified and classified as trivial or non-trivial. They are reproduced below. Problem 5 is the intended problem for this task.

– *Problem 1 (P1):* Find how to cut the 12 identical sausages to share them equally among the 18 people. [Trivial]

– *Problem 2 (P2):* Find the amount of sausages each person will receive when the 12 identical sausages are shared equally among the 18 people. [Trivial]

– *Problem 3 (P3):* Find the number of cuts needed to share the 12 identical sausages equally among the 18 people. [Non-Trivial]

– *Problem 4 (P4):* Find a few methods to cut the 12 identical sausages to share them equally among the 18 people. [Non-Trivial]

– *Problem 5 (P5):* Find the least number of cuts needed to share the 12 identical sausages equally among the 18 people. [Non-Trivial; Intended Problem]

Table 7.14 shows the TPO for the problem-posing outcomes for the 10 students' investigation of the Sausage Task. The classification of specific problems as trivial or non-trivial had passed the inter-coder reliability test (see Section 5.4). 'Other Trivial Problems' refer to trivial problems other than P1 and P2. No student had posed other non-trivial problems. It was observed that the 10 students posed a total of 22 specific problems for the Sausage Task, or an average of 2.2 problems per student. The actual number of problems posed by a student ranged from 1 to 4 problems. Out of the 22 problems, 12 of them (or 55%) were trivial while the other 10 (or 45%) were non-trivial. P1 was the most common problem that the students posed, but only 2 students

(S1,S5) posed this problem explicitly. The other 6 students just went ahead to find how to cut the sausages without posing it as a problem. Although only one student (S9) posed the problem of finding the amount of sausages each person will get (P2), 6 other students (S1,S2,S4,S5,S8,S10) also found the amount of sausages each person will receive when solving other problems such as P1 or P5. In other words, these 6 students did not set out to solve P2.

Table 7.14 Problems Posed for Sausage Task

	P1 (trivial)	P2 (trivial)	P3 (non-trivial)	P4 (non-trivial)	P5 (non-trivial)	Other Trivial Prob.	Total No. of Trivial Prob.	Total No. of Non-Trivial Prob.	Total No. of Prob.
S1	✓				✓		1	1	**2**
S2	*		*		✓		1	2	**3**
S3	*					1	2	0	**2**
S4	*					1	2	0	**2**
S5	✓			✓			1	1	**2**
S6	*			✓			1	1	**2**
S7	*						1	0	**1**
S8					✓		0	1	**1**
S9		✓	✓		*	1	2	2	**4**
S10	*		✓		✓		1	2	**3**
Total	**8**	**1**	**3**	**2**	**5**	**3**	**12**	**10**	**22**

* Did not pose problem explicitly

It was further observed that 50% of the students (S3-S7) did not find the number of cuts for this task. Only 3 students (S2,S9,S10) posed the problem of finding the number of cuts (P3) and the least number of cuts (P5) needed to share the 12 identical sausages equally among the 18 people, while 2 students (S1,S8) posed P5 directly. Thus a total of 5 students attempted to find the number of cuts, with all 5 of them posing P5 in the end. However, P5 was the intended problem for this task, which means that only 50% of the students in this study were able to pose this problem.

Since a Type B investigative task, such as the Sausage Task, is obtained from a mathematical problem by removing the original intended problem and replacing it with the word 'investigate' (Frobisher, 1994) as explained in Section 2.1.4(b), the implication of this finding is that teachers need to be aware that there is a possibility of 'losing' the intended problem that they might want their students to solve. Thus there is a need for teachers to think about how to guide their students to pose the intended problem for a Type B task, and yet not close up the task by restricting the students' freedom and creativity to pose other types of problems to solve.

A detailed analysis of the students' protocols shows that the students posed P1 to P5 in ascending order of the number. For example, S9 posed P2, then P3, followed by P5. This is not surprising since P1 to P5 had been formulated in a logical sequence during the task analysis of this task. However, it does not mean that the students must pose *all* these problems in order, e.g. S5 posed P1 and P4, but she did not pose P2 and P3.

None of the students posed other non-trivial problems, but 3 students (S3,S4,S9) posed other trivial problems. The first student (S3) posed the trivial problem of finding factors of 18, or "numbers that can be divided by 18 without decimals" (in his own words), at the start of the test. The second student (S4) posed the problem of finding the number of times he needed to cut the sausages, which is different from finding the number of cuts (P3), since he had this idea that the knife might not be long enough to cut all the sausages together at one go, as shown in his protocols below.

> "So maybe the knife is very small, so I can only cut one sausage at a time. So how many times should I need to cut in order to, so each person should, will get 2/3 of sausage [write: 2/3]?" [S4; Sausage Task]

What he meant was made clearer when he then proceeded to cut all the 12 sausages together at one go with a long knife, i.e. he cut a total of only one time, instead of 12 times with a ‘very small’ knife. The third student (S9) posed the problem of finding how to *share* the sausages at the start of the test, which is different from finding how to *cut* the sausages (P1) because his answer was just “each person would get 2/3 of a sausage”.

(b) Effect of Problem-Posing Processes on Outcomes

After analysing the different types of problems (outcomes) posed by the students for the Sausage Task, we now turn our attention to study the effect of the problem-posing processes on the outcomes. Table 7.15 shows the TPO for the problem-posing processes and outcomes for the 10 students’ investigation of the Sausage Task. The ability to pose a problem does not depend only on the final outcome: whether it is non-trivial or trivial. It also depends on whether the student is able to pose the problem ‘naturally’ without struggling, or whether the student struggles to pose the problem ‘eventually’. From the data analysis in part (a) of this section, it was observed that the students posed P1 to P5 in ascending order of the number, so this is how Table 7.15 should be interpreted, e.g. S9 posed P2, then P3, followed by P5. As for the 3 students (S3,S4,S9) who posed one ‘other problem’ each, both S3 and S9 posed the ‘other problem’ as their first problem while S4 posed the ‘other problem’ as his second problem.

Table 7.15 Problem-Posing Processes and Outcomes for Sausage Task

	Processes**				Outcomes***					
	Refer to following to think of problem				P1	P2	P3	P4	P5	Other Prob.
	Task	Current Working	Previous Result	Check -list						
S1	1		1		N				N	
S2	1	1	1		N*		N*		N	
S3	2				E*					E
S4	2				N*					E
S5	1		1		N			N		
S6	1			1	N*			E		
S7	1				N*					
S8	1								N	
S9	1	1	2			N	N		N*	N
S10	3	1	2	2	E*		E		E	
Total	**14**	**3**	**7**	**3**	**8**	**1**	**3**	**2**	**5**	**3**

* Did not pose problem explicitly.
** The students did not analyse the feasibility of the goal or the problem (MG).
*** 'N' indicates posing the problem naturally while 'E' indicates posing the problem eventually.

Table 7.15 shows that 8 students (S1,S2,S4,S5-S9) posed their first problem (either P1, P5 or the 'other problem' for S9) naturally. A detailed analysis of their protocols shows that all the 8 students had referred to the task statement in the understanding stage, before posing their first problem (S1,S5,S8,S9) or going ahead to find how to cut the sausages without posing it as a problem explicitly (S2,S4,S6,S7). The following shows the protocols of a student (S1) who read the task once and then posed the problem naturally:

> "I need to cut 12 identical sausages so that I can share them equally among 18 people. Investigate. The question is [write: Q:] is that [start writing] How am I going to cut the 12 sausages [stop writing]." [S1; Sausage Task]

However, the ability to pose the first problem without much difficulty does not mean that the students could also pose subsequent problems naturally after solving their first problem. Table 7.15 shows that 4 students (S1,S2,S5,S9) posed their subsequent problems naturally. A detailed analysis of their protocols shows that all the 4 students had either referred to (i) their current working, or (ii) a previous result that they had obtained, when thinking of a new problem to pose. For example, S2 did not pose his first problem explicitly but he just went ahead to find how to cut the sausages (P1). He then used a cutting method that involves cutting each of the 12 sausages into 18 equal parts. In the process of cutting (i.e. referring to the current working), he went on to find the total number of cuts (P3) without posing it as a problem explicitly. After solving both P1 and P3, he looked at his solutions (i.e. referring to the previous results) and said that the cutting method was too troublesome, which prompted him to pose the problem of finding the least number of cuts (P5) explicitly. In other words, for students who posed problems naturally for the Sausage Task, referring to the task statement had helped them to pose their first problem, but for subsequent problems, what was more helpful was their current working or previous results, which had acted as a springboard for more problems.

Table 7.15 shows that 15 of the 22 problems (or 68%) were posed naturally while 7 problems (or 32%) were posed eventually. It was observed from the same table that 2 students (S3,S10) struggled to pose all their problems while 2 other students (S4,S6) struggled to pose their second problem. A detailed analysis of their protocols suggests that the processes that had helped them to pose their problem(s) eventually were the reference to (i) the task statement, (ii) a previous result, and (iii) the given checklist of investigation processes (see Appendix H). For example, S10 read the task statement at the start of the test and then proceeded to cut the 12 identical sausages to share them equally among the 18 people, but she did not treat this as solving P1. It was after she had referred to the checklist, re-read the task statement, and checked her current working, before she realised that she had solved a problem, as shown in her protocols below.

> "Um, okay, so this is already the solution." [S10; Sausage Task]

This suggests that she did not pose P1 naturally. Then she struggled to think of a second problem. In the end, she referred to her previous result or solution of P1 before posing her second problem eventually: find the number of cuts (P3). After solving this problem, she was again stuck for some time. Then she referred to the checklist, re-read part of the task statement once, and referred to the previous result where she read part of her solution, before posing her third problem eventually: find the least number of cuts (P5). Thus what had helped her to think of a problem to pose were the reference to the task statement, a previous result, and the checklist. Although she had referred to her current working before she realised that she had solved P1, the process of referring to her current working was not for the purpose of posing *more* problems to solve. In fact, none of the students who struggled to pose problems for this task referred to their current working when thinking of what to pose.

None of the students analysed the feasibility of their goal or problem (MG) to see if it was worth pursuing. If they had, they might have realised that most of their problems, such as finding a cutting method (P1) and the amount of sausage that each person will receive (P2), are trivial. Although the students could still solve these trivial problems, they should try to think of more non-trivial problems to solve, such as finding the number of cuts (P3) and the least number of cuts (P5).

Summary

To summarise, 5 of the 10 students in the present study were unable to pose the intended problem for the Sausage Task, which is to find the least number of cuts. Some of them also encountered difficulty in thinking of other problems to pose. For students who were able to pose problems naturally, referring to the task statement helped them think of their first problem, and their current working or previous results were a fertile source for subsequent problems. For those who struggled to think of problems to pose, referring to the task statement, previous results, or the given checklist of investigation processes, helped in posing a problem eventually. None of the students analysed the feasibility of their goal, which might have helped them to think of non-trivial problems to pose and solve, instead of just solving trivial ones.

7.3.3 Using Other Heuristics (Stage 3)

As posited in the theoretical investigation model for cognitive processes in Section 3.2.1, the students should use other heuristics such as reasoning (coded as RE in the protocols), algebra (AL), and visualising information (VI) by drawing a diagram, to think of how to solve the problem that they had posed for the Sausage Task (Type B). However, none of the students used algebra for this task. Specialising in this stage will be discussed later in Section 7.3.7 after the students had extended the task to generalise. During the task analysis of this task in Appendix E, three cutting methods to share the 12 identical sausages equally among the 18 people were identified. They are reproduced below:

- *Cutting Method A (Usual Method):* Cut each of the 12 sausages into three equal parts. Then each person gets two parts. Total number of cuts = 12 × 2 = 24.

- *Cutting Method B (Shortest Method):* Cut each sausage once at the 2/3-mark to divide it into two parts: 2/3 and 1/3 of a sausage respectively. Then each person either gets 1 × 2/3 of a sausage, or 2 × 1/3 of a sausage. Total number of cuts = 12.

- *Cutting Method C (Long Method[18]):* Cut each sausage into 18 equal parts. Then each person receives one part from each of the 12 sausages, i.e. a total of 12 parts. Total number of cuts = 12 × 18 = 216.

For Type B tasks, the ability to use other heuristics effectively is task dependent. For the Sausage Task, students could make use of the fraction of sausage each person will receive to discover Cutting Method B, or the LCM of the number of sausages and the number of people to help find Cutting Method A. As Method A is what most people would naturally use, students who only discovered this method were reckoned to have used other heuristics ‘ineffectively’. If they struggled but eventually discovered Method B that gives the least number of cuts, they were reckoned to have used other heuristics ‘quite effectively’. But if they were able to discover Method B naturally without struggling, whether right from the beginning or after using Method A, then they were reckoned to have used other heuristics ‘effectively’. It was expected that the students would not use Method C since it is clear that it would give more cuts than Method A, and it is very troublesome to cut each sausage into so many parts.

[18] There is no longest method because the number of parts each of the 12 sausages can be cut, so that they can be shared equally among 18 people, can be any multiple of 18, although there is a practical limit as to how small a sausage can be cut.

Table 7.16 shows the TPO for the processes and outcomes of using other heuristics for the 10 students' investigation of the Sausage Task. The three reasoning processes that the students had used to help them think of how to cut the sausages are: (i) the fraction of sausage each person will receive, (ii) the lowest common multiple (LCM) of 12 and 18, and (iii) the factors of 12 and 18.

Table 7.16 Using Other Heuristics for Sausage Task

	Processes*						Outcomes				
	Reasoning using			VI	MF	MA	Cutting Methods				
	Fraction	LCM	Factors				A	B	C	Others	Total
S1	✓			1			✓	✓			**2**
S2	✓			1			✓		✓		**2**
S3				3			✓			1	**2**
S4	✓	✓		1				✓		1	**2**
S5	✓			1			✓		✓		**2**
S6		✓	✓	11			✓			4	**5**
S7		✓	✓	7	1	1	✓		✓	2	**4**
S8*	✓	✓	✓	7			✓	✓		4	**6**
S9	✓			8			✓	✓			**2**
S10	✓	✓		2			✓	✓			**2**
Total	**7**	**5**	**3**	**42**	**1**	**1**	**9**	**5**	**3**	**12**	**29**

* Although S8 also exhibited metacognitive awareness (MA) in this stage, it was not included in this table because it would be analysed in the checking stage in Section 7.3.10 since it had something to do with her sensing something amiss and checking her working.

Table 7.16 shows that most of the students (i.e. 9 of them) attempted Cutting Method A as expected. The remaining student (S4) did not use Method A but went straight to use Method B. So a total of 5 students used Method B. However, it was surprising that 3 students actually used Method C, which is to cut each of the 12 sausages into 18 equal parts. Five students also tried other cutting methods, such as cutting each sausage into halves or into 6 equal parts. Most of these 'other cutting methods' will not work. In total, the 10 students tried 29 cutting methods, or an average of 2.9

methods per student for the original task. The actual number of cutting methods attempted by a student ranged from 2 to 6 methods.

A detailed analysis of the students' protocols suggests that reasoning using the fraction of sausage each person will receive has helped the 5 students (S1,S4,S8-S10) to think of Method B, which gives the least number of cuts. Table 7.16 shows that 7 students (including the 5 students who discovered Method B) found the fraction of sausage each person will receive. The first student (S1) used Method A first, then thought for a while about the 2/3 of a sausage that each person will receive before discovering Method B, which requires each sausage to be cut at the 2/3-mark. The second student (S4) used the LCM of 12 and 18 to reason that there will be a total of 36 pieces of sausages to be shared equally among the 18 people. But he did not use Method A. Instead, he reasoned further that each person will get 2 pieces or 2/3 of a sausage[19], which led him to discover Method B. The third student (S9) reasoned straightaway that it was enough to cut each sausage at the 2/3-mark since each person will receive 2/3 of a sausage. Therefore, these 3 students (S1,S4,S9) were able to reason 'effectively' using the fraction of sausage each person will receive to help them discover Method B naturally without struggling, whether right from the beginning or after using Method A.

The other two students (S8,S10) discovered Method B only at a later stage when they found the fraction of sausage each person will receive, so their use of other heuristics was 'quite effective'. For example, after using LCM and Method A, S8 then cut the sausages any old how. Her 'other methods' include cutting the first 4 sausages into

[19] This was a longer way to obtain 2/3 of a sausage: 12 divided by 18 will give 2/3 straightaway.

halves and the next 4 sausages into 3 parts each, which did not work out; cutting the first 5 sausages into halves and the remaining 7 sausages into 3 parts each, which also did not work out; and cutting the first 6 sausages into halves, which also did not work out. She did not understand that she could not cut the sausages any old how because the fraction of sausage each person will receive is critical in determining how the sausages should be cut. Finally, she thought of the fraction of sausage each person will receive and discovered Method B. The following shows her protocols when she started to think of 2/3.

> "Ok … I … think eventually right, each person should have 2/3 of a sausage. Yah … So that means … I might have to cut … like … Oh, I know already … Instead of cutting into half right [draw 5 ovals], when I slice one, I can slice it into … I can split it into 1/3 and 2/3." [S8; Sausage Task]

Therefore, finding the fraction of sausage each person will receive had helped these 5 students (S1,S4,S8-S10) to think of Method B. But it was not helpful for two other students (S2,S5) who found the fraction of sausage each person will receive because they did not discover Method B, e.g. one of them (S2) did not reduce the fraction of sausage each person will receive (i.e. 12/18) to the lowest terms (i.e. 2/3), so 12/18 did not lead him to Method B.

On the other hand, finding the LCM of 12 and 18 naturally led to Method A for 3 students (S7,S8,S10). Table 7.16 shows that 5 students (S4,S6-S8,S10) found the LCM of 12 and 18. Four of them (except S6) started to find a cutting method by calculating the LCM of 12 and 18 to give 36. This means that, after cutting, there will be 36 pieces of sausages to be shared equally among the 18 people, i.e. each of the 12 sausages must be cut into 3 pieces. So using the LCM of 12 and 18 naturally led to Method A for 3 of the 4 students (S7,S8,S10). As discussed above, the fourth student (S4) did not use Method A but he went on to reason that each person will get 2 pieces or 2/3 of a sausage, which led him to discover Method B. The fifth student (S6) did not use any of the three reasoning processes but she just cut each sausage into 3 parts (Method A). However, she was concerned that the rounded ends were different from the middle portions, which had been discussed earlier in Section 7.3.1. So she ended up trying different methods of cutting, including cutting each sausage into halves or quarters, which did not work out. In the process, she tried using factors of 18, factors of 12, and then the LCM of 12 and 18, which were not helpful, since there is no solution if the rounded ends are considered different from the middle portions.

Other than S6 described in the preceding paragraph, Table 7.16 shows that 2 other students (S7,S8) also found the factors of 12 and 18, but it did not help them to discover Method A or B. Table 7.16 also shows that one other student (S3) did not use any of the three reasoning processes but he just cut each sausage into 2 equal parts, which did not work out; so he just cut each sausage into 3 equal parts (Method A) and it worked, but he did not discover Method B. Hence, the ability to identify and use critical information, such as the fraction of sausage each person will receive, had helped 5 students (S1,S4,S8-S10) to discover Method B.

Table 7.16 shows that 3 students actually used Method C, which is very troublesome since each sausage has to be cut into 18 equal parts. The first student (S2) started with Method C, which he initially said was “the simplest way to cut” (in his own words). But he soon recovered and found Method A. The second student (S5) tried to find different cutting methods and so she just listed Method C as one possible method after using Method A. The last student (S7) used Method C as she was trying to find out whether the number of parts each sausage should be cut is a factor of 18.

The use of other heuristics includes not only reasoning as described above, but also visualising information. Table 7.16 shows that all the students visualised information by drawing a diagram with a total frequency of 42 times. Generally, the students drew a diagram in this stage for three purposes: (i) to think of a cutting method, (ii) to try out a cutting method that they had thought of to see if it works, and (iii) to present the solution. For example, S7 drew 12 sausages and then said:

> "So these 12 sausages must be divided between 18 people ... So … how should I divide? …" [S7; Sausage Task]

The student was (i) visualising information to think of a cutting method, but it was not helpful. So she used multiples and factors to help her think of how to cut. After a short while, she decided to cut each sausage into 3 equal parts, but she was not sure whether the method will work. So she (ii) tried out the cutting method by drawing two vertical lines to cut each sausage. After cutting the eighth sausage into 3 equal parts, she reasoned that there will be a total of 36 parts, which can be shared among the 18 people. In other words, the student was now certain that her cutting method will work. Then she (iii) presented her solution by drawing two vertical lines for each of the four remaining sausages, and she even shaded the first few sausages as shown in Figure 7.9, although she probably decided that it was too troublesome to continue shading the rest of the sausages.

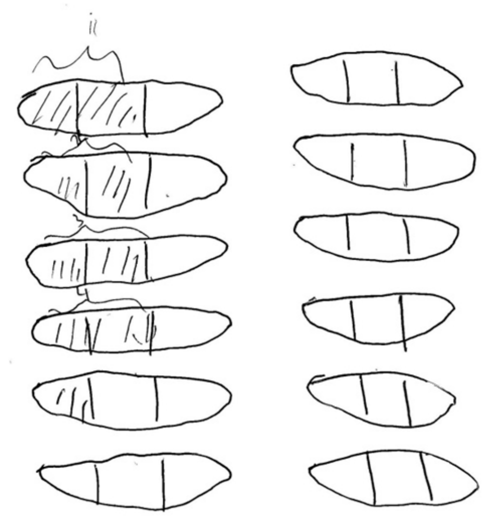


Figure 7.9 Visualising Information by S7 for Sausage Task

Just like S7 above, 5 other students (S3,S5,S6,S8,S10) who drew a diagram to think of how to cut the sausages were unable to find a workable cutting method just by visualising the sausages alone. On the other hand, visualising information was more helpful in checking whether a cutting method would work, e.g. in addition to S7, visualising information had helped 5 other students (S1-S3,S6,S8) to do so. In fact, drawing 17 cuts on a sausage to divide it into 18 equal parts had led one student (S2) to realise that his cutting method (Method C) was too troublesome, so he went on to find a shorter method (Method A).

Table 7.16 shows that only one student (S7) engaged in 2 instances of metacognitive behaviours. She analysed the feasibility of her plan (MF) in using multiples or factors to think of a cutting method on one occasion, but it was not effective because it did not work out. However, she exhibited metacognitive awareness (MA) on another occasion when she realised that she had tried a cutting method before, thus saving her precious time from trying the same cutting method again. None of the other students analysed whether their cutting method will work before trying it out, and thus wasted a lot of time when the method failed. For example, as discussed above, S8 initially cut the sausages any old how without analysing whether the cutting method will work, so she wasted precious time on a wild goose chase until finally, she realised that each person must receive 2/3 of a sausage, which led her to discover Method B.

Summary

To summarise, some of the students in the present study were able to reason effectively by using critical information, such as the fraction of sausage each person will receive, to find the shortest method, while others were not able to do so. Some of them did not even try to analyse whether their cutting method was feasible or worth pursuing, but they just went ahead to cut the sausages any old how. On the other hand, visualising information by drawing a diagram had helped some of the students to check whether their cutting method will work, but visualising information per se did not help them to think of a workable cutting method. Metacognitive awareness had also helped one student not to repeat a cutting method that she had tried earlier.

7.3.4 Conjecturing (Stage 4)

As explained in the theoretical model for cognitive processes in Section 3.2.1, students can solve a problem that they had posed for the original task by using other heuristics with or without formulating any conjecture. For example, students can solve the problem of finding a method to cut the 12 sausages to share them equally among the 18 people for the Sausage Task without formulating any conjecture. But if they want to find a method that gives the least number of cuts, then the method is only a conjecture (coded as FC in the protocols) unless they can justify whether it will indeed give the least number of cuts. Therefore, the processes for the conjecturing stage for Type B tasks are actually using other heuristics, such as reasoning, in the previous stage, and the outcomes in this stage are whether the conjectures formulated are non-trivial or trivial, and whether they are correct or wrong. Table 7.17 shows the TPO for the conjecturing processes and outcomes for the 10 students' investigation of the Sausage Task. Only 5 students were shown because they were the only ones who posed the problem of finding the least number of cuts and so only their cutting method was a conjecture to be proven or refuted. The conjectures were classified as trivial or non-trivial based on the task analysis in Appendix E and the classification had passed the inter-coder reliability test (see Section 5.4).

Table 7.17 Conjecturing Processes and Outcomes for Sausage Task

	Processes*	**Outcomes**		
	Reasoning	**Conjecture**	**Trivial or non-trivial?**	**Correct or wrong?**
S1	Effective	Conjecture 1	Non-trivial	Correct (use Cutting Method B)
S2	Ineffective	Conjecture 1	Non-trivial	Wrong (use Cutting Method A)
S8	Quite Effective	Conjecture 1	Non-trivial	Correct (use Cutting Method B)
S9	Effective	Conjecture 1	Non-trivial	Correct (use Cutting Method B)
S10	Quite Effective	Conjecture 1	Non-trivial	Correct (use Cutting Method B)
Total	**1 ineffective**	**5 conjectures**	**5 non-trivial**	**4 correct; 1 wrong**

* The students did not exhibit the metacognitive behaviours, MF and MA, in this stage.

Table 7.17 shows that 4 of the 5 conjectures were correct because 4 of the students (S1,S8-S10) were able to use reasoning 'effectively' or 'quite effectively' to discover Cutting Method B, which will give the least number of cuts. Although another student (S4) also used Method B as discussed in the previous section, he did not pose the problem of finding the least number of cuts, so his solution was not a conjecture to be proven or refuted. All the conjectures were non-trivial although the conjecture for S2 was wrong because he used Method A, which will not give the least number of cuts.

7.3.5 Justifying and Generalising (Stages 5 and 6)

Table 7.18 shows the TPO for the justifying and generalising processes and outcomes for the 10 students' investigation of the Sausage Task. Only 5 students were shown because they were the only ones who posed the problem of finding the least number of cuts and so only their cutting method was a conjecture to be proven or refuted.

Table 7.18 Justifying / Generalising Processes and Outcomes for Sausage Task

	Conjecture	Justifying Processes*			Justifying and Generalising Outcomes
		Naïve Testing	Non-proof Argument	Formal Proof	
S1	Correct Non-trivial Conjecture 1		✓		**Proven conjecture correctly (no generalisation)**
S2	Wrong Non-trivial Conjecture 1				Wrongly accepted conjecture as true without testing
S8	Correct Non-trivial Conjecture 1				Wrongly accepted conjecture as true without testing
S9	Correct Non-trivial Conjecture 1				Wrongly accepted conjecture as true without testing
S10	Correct Non-trivial Conjecture 1		✓		**Proven conjecture correctly (no generalisation)**
Total	**4 correct; 1 wrong**	**0**	**2**	**0**	**2 proven; 0 generalisation****

* The students did not exhibit the metacognitive behaviours, MF and MA, in this stage.
** There was no generalisation for all the conjectures since these conjectures were not general results.

Table 7.18 shows that 3 of the 5 students wrongly accepted their conjecture as true without testing while the other 2 students proved their conjecture correctly without leading to any generalisation since their conjecture was not a general result. One of the students (S2) only managed to discover Cutting Method A, so he wrongly accepted that the least number of cuts is 24 without any justifiable basis. The other 4 students used Cutting Method B, but 2 of them (S8,S9) accepted that the least number of cuts is 12 without any basis. In fact, another student (S1) did not prove his conjecture initially, but he decided to review his solution for the original task when he was stuck with an extension. In the process, he showed that the least number of cuts is 12 by using a non-proof argument as follows:

> "But you have to share … *every sausage have to cut* [italics mine]. How about … not cut? How about leave the whole sausage? Leaving the whole sausages, yes that will save … a lot of them, quite a lot of the number of cuts … Um, but how am I going to do that? … How to cut? … How to avoid, cutting a sausage, that means, that leaving some sausages as a whole? … So that means that 1, 12 among 18 people, but have to involve *fraction* [italics mine]. Yes." [S1; Sausage Task]

Although he did not phrase his argument properly, he did say the essence of it: each sausage must be cut at least once because no one will receive one whole sausage as each person will receive only 2/3 of a sausage; and since he had found a method that involves cutting each of the 12 sausage only once (Method B), then the least number of cuts is 12. A more detailed argument along this line is given in the task analysis of this task in Appendix E.

The last student (S10) tried to prove her conjecture immediately after formulating it, but her non-proof argument was not as rigorous as that of the previous student because she did not mention about the need to cut each sausage at least once. Instead, she tried to see if she could cut the 12 sausages using 6 cuts instead of 12 cuts. The reason why she used 6 cuts was because there were only 12 sausages but there were 18 people, so there was a need to cut the sausages 6 times to give a total of 18 pieces in order that each person will get one piece. In the end, the student realised that it was not possible to cut the sausages only 6 times. Mason et al. (1985) discussed the need to convince oneself before convincing others. So what S10 had done was that she had convinced herself that the least number of cuts is 12, but her reasoning was not rigorous enough to convince others. Table 7.18 shows that none of the students used naïve testing or a formal proof in this stage.

7.3.6 Extension (Stage 8)

The next stage should be the checking stage in the review phase before extending the task to generalise. Since the processes and outcomes of the checking stage do not depend on whether the students are solving the original task or the extended task, it is more appropriate to analyse all the checking processes and outcomes together in Section 7.3.10 later. So we will proceed directly to the extension stage. The treatment for the data analysis in this section will be the same as that for problem posing in Section 7.3.2 because there is a need to analyse in detail the types of extensions posed by the 10 students for the Sausage Task before the empirical data could be examined to study the effect of the extension processes on the outcomes in this stage.

(a) Extension Outcomes

During the task analysis of the Sausage Task in Appendix E, some possible extensions that students could pose for this task were identified and classified as trivial or non-trivial. They are reproduced below. Extension 2 is the intended extension for this task. Both extensions are for the purpose of generalising.

- *Extension 1 (E1):* Find the amount of sausages that each person will receive when n identical sausages are shared equally among m people. [Trivial]

- *Extension 2 (E2):* Find the least number of cuts needed to share n identical sausages equally among m people. [Non-Trivial; Intended Extension]

Table 7.19 shows the TPO for the extension outcomes for the 10 students' investigation of the Sausage Task. The classification of extensions as trivial or non-trivial had passed the inter-coder reliability test (see Section 5.4). 'Other Extensions' refer to extensions other than E1 and E2. As explained in the theoretical model for cognitive processes in Section 3.2.1, the students were expected to extend Type B tasks to generalise, but they could also extend the tasks without generalising.

Table 7.19 Extensions for Sausage Task

	E1 (trivial)	**E2 (non-trivial)**	**Other Extensions to generalise**		**Other Extensions but not to generalise**		**Total No. of Extensions**		
			Trivial	**Non-Trivial**	**Trivial**	**Non-Trivial**	**Trivial**	**Non-Trivial**	**All**
S1		✓			1		1	1	**2**
S2	✓				2		3	0	**3**
S3				1*	1		1	1	**2**
S4	✓				2		3	0	**3**
S5					1		1	0	**1**
S6					3		3	0	**3**
S7			1				1	0	**1**
S8				1			0	1	**1**
S9		✓					0	1	**1**
S10				1	1		1	1	**2**
Total	**2**	**2**	**1**	**3**	**11**	**0**	**14**	**5**	**19**

* Did not pose extension explicitly

Table 7.19 shows that the 10 students posed a total of 19 extensions for the Sausage Task, or an average of 1.9 extensions per student. The actual number of extensions posed by a student ranged from 1 to 3 extensions. Out of the 19 extensions, 14 of them (or 74%) were trivial while the other 5 (or 26%) were non-trivial. Only 8 extensions (or 42%) were for the purpose of generalising (including E1 and E2), while the other 11 (or 58%) were not for the purpose of generalising. Two students (S5,S6) did not attempt to generalise at all.

It was observed that only 2 students (S1,S9) posed the intended extension (E2) of finding a formula for the least number of cuts to share *n* identical sausages equally among *m* people. From the data analysis in Section 7.3.2, it was found that 5 of the students (S3-S7) did not even find the number of cuts for the original task (because they did not pose P3 or P5), so they could not possibly pose E2. Out of the other 5 students (S1,S2,S8-S10) who posed the problem of finding the least number of cuts (P5) for the original task, only 2 of them (S1,S9) tried to generalise the result (E2), while the other three (S2,S8,S10) tried to generalise in different manners: S2 tried to find a formula for the amount of sausages that each person will receive (E1), S8 tried to find the conditions for her cutting method to work (which will be discussed later in this section), while S10 tried to find any formula but decided that this type of tasks could not have any formula. Another student (S4) also posed E1.

Table 7.19 shows that 8 students (S1-S7,S10) also posed other trivial extensions, but most of these extensions were not for the purpose of generalising. Although 7 students (S1-S6,S10) changed the number of sausages and / or the number of people in the task statement, they did not have the intention to generalise. For example, S2 changed the numbers of sausages and people to 3 sausages and 20 people respectively, and he just found how to cut the sausages and the amount of sausages each person will receive. Then he changed the numbers to 20 sausages and 3 people. This time, he only found the amount of sausages each person will receive, without even finding how to cut the sausages. Next, he tried to think of other extensions but he was stuck. This shows that he did not have the intention to generalise his first two extensions, which were considered as trivial. It was only later that he decided to generalise by finding a general formula for sharing things (i.e. divide the number of items to be shared by the number of recipients) when he extended the task for the third time by changing the numbers to 10 sausages and 17 people. This last extension was actually similar to E1. Although another 4 of the 7 students decided to generalise eventually, just like S2, the remaining 2 students (S5,S6) did not generalise at all.

Two students (S4,S6) also changed other variables. The first one (S4) changed the context to money: find how much each person had to pay for his share of sausages if the cost of each sausage was $1. Surprisingly, the student believed that "the money cannot be divided" (in his own words), i.e. each person cannot pay $0.66 (he did not round 2/3 = $0.\dot{6}$ up to $0.67). So he decided that the "simplest way" was for 3 people to share $2 to buy 3 sausages. In the end, he rounded up to $0.70 per person, but there will be an extra 10 cents for the 3 people when they paid $2.10, instead of $2, for the 2 sausages. The second student (S6) changed the shape of the sausages to rectangle and hexagon since she could not solve the original task as she wrongly believed that the rounded ends of the sausages were different from the middle portions, which had been described earlier in Section 7.3.1. As a result, she experimented with hexagonal sausages where it was possible to cut each hexagon[20] into 6 equal parts to share them equally among the 18 people.

Table 7.19 shows that 3 students (S3,S8,S10) also posed other non-trivial extensions to generalise. A detailed analysis of their protocols shows that all the 3 students tried to generalise *how* to cut *n* sausages to share them equally among *m* people. The first one (S3) extended the task to generalise Cutting Method A for *x* sausages and *y* people. The second student (S8) tried to generalise Cutting Method B for different

[20] She did not mention that the hexagons must be regular.

numbers of sausages and people, but in a different manner. She wanted to know whether Method B would work for which numbers of sausages and people. In the end, she found a general cutting method (which is not generalised Method B) that allowed her to share $6n$ sausages equally among $6n + 6$ people, although this method will not give the least number of cuts (her conjecture will be discussed later in Section 7.3.8). The third student (S10) wanted to find a general formula for a cutting method. Initially, she said that her cutting method used the idea of LCM, but she then realised that there was actually no need to use the LCM to find the cutting method. So she decided that it was not possible to find a general formula for a cutting method.

(b) Effect of Extension Processes on Outcomes

After analysing the different types of extensions (outcomes) posed by the students for the Sausage Task, we now turn our attention to study the effect of the extension processes on the outcomes. Table 7.20 shows the TPO for the extension processes and outcomes for the 10 students' investigation of the Sausage Task. The ability to extend does not depend only on the final outcome: whether it is non-trivial or trivial. It also depends on whether the student is able to extend 'naturally' without struggling, or whether the student struggles to extend the task 'eventually'. It was observed that only one student posed the intended extension (E2) naturally, while the other 3 students posed E1 or E2 eventually. In fact, out of the 19 extensions, only a small proportion (5 extensions or 26%) were posed naturally. This was unlike the problem-posing stage where most of the problems (68%) were posed naturally (see Section 7.3.2). This suggests that the students had more difficulty extending the task than posing problems for the original task.

Table 7.20 Extension Processes and Outcomes for Sausage Task

	Processes					Outcomes**			
	Refer to following to think of extension				MG	E1	E2	Other Extensions to generalise	Other Extensions but not to generalise
	Task	Current Working	Previous Result	Check-list					
S1	1	1					N		1N
S2			2	1		E			2E
S3	2		1					1E*	1E
S4	1		1	1		E			2E
S5	1								1N
S6	1			3					3E
S7	1							1N	
S8			1					1N	
S9	1		1				E		
S10				2	1			1E	1E
Total	**8**	**1**	**6**	**7**	**1**	**2**	**2**	**4**	**11**

* Did not pose extension explicitly
** 'N' indicates extending the task naturally while 'E' indicates extending the task eventually.

Table 7.20 shows that 4 students (S1,S5,S7,S8) extended the task naturally. A detailed analysis of their protocols shows that all the 4 students either referred to (i) the task statement, (ii) the current working, or (iii) a previous result, to think of a problem to extend. For example, the first student (S1) said that he wanted to extend the task, then re-read the task statement once and immediately swapped the two given numbers around to give 18 sausages and 12 people. While he was writing down his first extension (i.e. referring to his current working), he suddenly realised that he needed to find some formula, which led him to his second extension to find a general formula for the least number of cuts (he did not continue his first extension anymore). Two other students (S5,S7) also re-read the task statement once before extending the task naturally by changing the number of sausages and / or the number of people. The last student (S8) used her previous result to extend the task naturally by asking whether her cutting method for the original task could be applied to share what other number of sausages equally among how many other people. For the other 6 students (S2-S4,S6,S9,S10) who extended the task eventually, a detailed analysis of their protocols suggests that the processes that had helped them were the reference to (i) the task statement (S3,S4,S6,S9), (ii) a previous result (S2-S4,S9), and (iii) the given checklist of investigation processes (S2,S4,S6,S10). Interestingly, the findings tally exactly with the processes that had helped the students to pose their problems for the original task (see Section 7.3.2).

It was further observed from Table 7.20 that one student (S10) analysed the feasibility of her goal (MG) when extending the task. This was the only instance in the pretest and the posttest that a student engaged in MG. The student was struggling to extend the task when she thought of finding a formula:

> "No problem already … Let me think for a while … No problem … Formula? But this one, how to have formula? … Um, this one is find the LCM first. Actually no need. Yah. Eh? …" [S10; Sausage Task]

The student analysed the feasibility of extending the task to find a formula. At first, she believed that there would not be a formula for this kind of tasks: "But this one, how to have formula?" Then she thought that the formula might be to find the LCM first, but she realised that it was not necessary to find the LCM in order to find a cutting method. However, she was unable to see the need to find a formula for the least number of cuts, which she had obtained for the original task. In the end, she did not pursue the extension to find a formula, so her analysis of the feasibility of extending the task to find a formula was not effective.

Summary

To summarise, 8 of the 10 students in the present study were unable to pose the intended extension for the Sausage Task, which is to find a general formula for the least number of cuts, partly because 5 of them did not even find the number of cuts for the original task. Instead, they extended the task to find a general formula for the amount of sausages each person will receive, or to find a general cutting method. Most of the students also encountered greater difficulty in extending the task than in posing problems for the original task. In fact, some students just changed the variables in the task statement without realising that the intention is to generalise. Nevertheless, all the students actually extended the task in one way or another. Just like the findings in the problem-posing stage, referring to the task statement, current working, or previous results, had helped some students to extend the task naturally, but for those who struggled to extend the task, referring to the task statement, previous results, or the given checklist of investigation processes, had helped some of them to extend the task eventually. There was only one instance when a student analysed the feasibility her goal to extend the task, but it was not effective.

7.3.7 Specialising and Using Other Heuristics (Stage 3) for Extension of Task

As posited in the theoretical investigation model for cognitive processes in Section 3.2.1, after extending the task, the students should specialise at this stage in order to generalise. But for each specific example that they specialise, they might have to use other heuristics to solve, just like for the original task.

(a) Specialising

Unlike the Kaprekar Task analysed in Section 7.2.3, the examples generated for the Sausage Task could not be classified as representative because there was only one type of pattern. Therefore, the specialising outcomes for the Sausage Task are actually the same as the conjecturing outcomes: whether the students were able to formulate a correct conjecture for their extension. Table 7.21 shows the TPO for the specialising processes and outcomes for the 10 students' extension of the Sausage Task. It does not include the first example for specialising, which is actually the original task: 12 sausages to be shared equally among 18 people. However, the table includes all the examples where the students changed the number of sausages and / or the number of people without the intention to generalise, and where they changed the context.

Table 7.21 shows that the 10 students generated a total of 29 examples to specialise, or an average of 2.9 examples per student, for the Sausage Task. This is lower than the average of 4.5 examples per student for the Kaprekar Task (see Section 7.2.3), which is not surprising since it will take a longer time to solve an example generated for the Sausage Task than to generate a sequence for the Kaprekar Task. The actual number of examples generated by a student for the Sausage Task ranged from 1 to 4 examples each, with an outlier of 10 examples generated by one student (S8). Just like the Kaprekar Task, most of the examples (55%) generated by the students for the Sausage Task were random. It was further observed that only 17 of the 29 examples were for the purpose of generalising (indicated by * in the table) and they were generated by 5 students (S1,S2,S7-S9), but most of them (i.e. 10 examples) were from the same student (S8). Although there were 3 other students (S3,S4,S10) who also extended the task to generalise, they did not generate examples for that purpose because they could use the example in the original task.

Table 7.21 Specialising Processes and Outcomes for Extension of Sausage Task

	Processes***			Outcomes	
	Random	**Purposeful**	**Systematic**	**Total No. of Examples**	**Formulated correct conjecture to generalise?**
S1	1 (E.g. 2)	1 (E.g. 3*)		**2**	Extend to generalise but no conjecture
S2	1 (E.g. 4*)	2 (E.g. 2,3)		**3**	Formulated trivial correct conjecture to generalise
S3	1 (E.g. 2)			**1**	Formulated non-trivial correct conjecture to generalise**
S4	2 (E.g. 2,3)			**2**	Formulated trivial correct conjecture to generalise**
S5	1 (E.g. 2)			**1**	Extend but did not try to generalise
S6	4 (E.g. 2-5)			**4**	Extend but did not try to generalise
S7	4 (E.g. 2-5)*			**4**	Extend to generalise but no conjecture
S8		10 (E.g. 2-11)*		**10**	Formulated one trivial wrong conjecture and one non-trivial correct conjecture to generalise
S9	1 (E.g. 2*)			**1**	Formulated two non-trivial conjecture (one correct and one wrong) to generalise
S10	1 (E.g. 2)			**1**	Extend to generalise but no conjecture**
Total	**16 (55%)**	**13 (45%)**	**0**	**29**	**5 formulated correct conjecture to generalise**

* Example generated for purpose of generalising.
** Did not generate examples to generalise because the students used the example in the original task.
*** The students did not exhibit the metacognitive behaviours, MF and MA, in this stage.

A detailed analysis of the students' protocols suggests that the students generally did not have any preference as to which variable to change first: the number of sausages or the number of people. For example, 4 students (S6-S9) changed the number of sausages while keeping the number of people constant at 18, 3 students (S3,S4,S7) changed the number of people while keeping the number of sausages constant at 12, and 7 students (S1,S2,S5-S8,S10) changed both the number of sausages and the number of people at the same time. This finding is not surprising as it really does not matter which variable of the Sausage Task is changed first since the students were expected to specialise systematically by keeping one of the two variables constant in order to try to find the formula for the least number of cuts. The task analysis in Appendix E shows two tables of values as exemplars on page 507: the first one for the case where the number of sausages is kept constant at $n = 12$, and the second one for the case where the number of people is kept constant at $m = 18$. Admittedly, it is not easy to observe the formula just by looking at these systematic examples because the formula m – HCF(m, n) is not easily discernible. But one student (S9) was able to discover the formula without even specialising systematically: he just tried a random example, and together with the example in the original task, he managed to observe the formula (see his investigation which had been discussed in detail in Section 6.3a).

Let us now examine the 13 purposeful examples generated by the 3 students (S1,S2, S8). The first student (S1) was more concerned about whether the number of sausages was greater or less than the number of people. This is a valid concern because if the number of sausages is greater than the number of people, then there is no need to cut some of the sausages. For example, if 18 identical sausages are to be shared equally among 12 people, each person will get one whole sausage first, and then what are left will be 6 sausages to be shared equally among the 12 people. In fact, after sharing all the possible whole sausages equally among the people, it would reduce to the case where the number of sausages is less than the number of people. But none of the students were able to argue in this manner.

Instead, S1 started with a random example (his Example 2) by swapping the given numbers around to give 18 sausages and 12 people, before realising immediately that he should generalise to find a formula for the least number of cuts. He then chose his next example (his Example 3) to be similar to the original task (which is considered as Example 1) where the number of sausages is less than the number of people: he purposefully chose 4 sausages and 10 people for his Example 3. After solving this example, he tried to search for patterns to find the formula but failed to observe any.

The second student (S2) observed that the numbers in the original task, 12 and 18, have common factors (other than 1), so he extended the task by trying numbers that are co-primes: he purposefully chose 3 sausages and 20 people for his Example 2. But he just solved this example by finding the fraction of sausage each person will receive without finding the number of cuts at all, unlike the original task where he found the least number of cuts. Then he purposefully selected more sausages than people by swapping the two numbers around to give 20 sausages and 3 people (his Example 3). Similarly, he solved this example by finding the fraction of sausage each person will receive without finding the number of cuts at all. He then went into a long period of hesitation, not knowing what else to investigate. It is indeed puzzling how the student decided that the key criteria in deciding what numbers to change were whether the numbers are co-primes, or whether the number of sausages is greater or less than the number of people.

The third student's (S8) extension is totally different from all the other 9 students. She wanted to know whether her Cutting Method B for the original task would work for which number of sausages, where the number of people is kept constant at 18. She suspected that Method B might work for multiples of 6, so she purposefully chose 24 sausages for her Example 2. She used the same Method B (cut each sausage at the

2/3-mark) but she did not realise that it will no longer give the least number of cuts. She then purposefully chose 9 sausages (multiples of 9) for her Example 3, 27 sausages (multiples of 3) for Example 4, and 32 sausages (multiples of 2) for Example 5, but she found out that Method B did not work for these three examples. So she went back to test whether Method B will only work for multiples of 6: she chose 54 sausages for Example 6, but this time, Method B failed. She then purposefully chose 144 sausages (multiples of 12) for Example 7: although Method B works, it no longer gives the least number of cuts. She did not realise that there was actually no need to cut the sausages for both Examples 6 and 7.

The student then tried to find out why Method B worked for multiples of 6 and 12. She observed that the number of people is 6 more than the number of sausages in the original task. So she purposefully chose 14 sausages and 20 people for Example 8, where she changed the number of people for the first time, but Method B did not work. She then tried a new cutting method (her 7^{th} one for the investigation): cut each sausage at the 3/4-mark, instead of at the 2/3-mark for Method B, which also did not work for Example 8. So she tried multiples of 6 again: she chose 18 sausages and 24 people for Example 9, but this time, her 7^{th} cutting method worked. At this stage, she was no longer interested in a cutting method that gives the least number of cuts, but a general cutting method that can work for certain numbers of sausages and people. She then suspected that her 7^{th} cutting method worked for multiples of 18, so she chose 36 sausages but still 24 people for Example 10, and discovered that her 7^{th} cutting method still worked. Then she wrongly accepted that her 7^{th} cutting method worked for multiples of 18 based on only two examples, but this trivial conjecture was false.

The student decided to go back to try the case where the number of people is 6 more than the number of sausages, but her examples that followed suggest that she had another condition: both numbers must be multiples of 6. In other words, she was trying to find a general cutting method to share $6n$ sausages equally among $6n + 6$ people. So she purposefully chose 30 sausages and 36 people for Example 11. With a sudden insight, she decided to cut each sausage at the 4/5-mark (her 8th cutting method), instead of at the 3/4-mark, but it did not work. Then she realised that she should cut each sausage at the 5/6-mark (her 9th cutting method) and it worked this time. This led her to formulate a non-trivial conjecture about a general cutting method to share $6n$ sausages equally among $6n + 6$ people, which is to cut each sausage at the $\frac{n}{n+1}$-mark, although she did not use algebraic notations to describe her conjecture.

Therefore, it was observed that the 3 students (S1,S2,S8) specialised purposefully for completely different reasons, and that purposeful specialising had helped one of them (S8) to formulate a non-trivial conjecture. It did not help the other two students to formulate their conjectures because their types of conjectures do not depend on the way they purposefully chose the number of sausages or the number of people. Similarly, the types of conjectures formulated by S3 and S4 also do not depend on how they chose the number of sausages or the number of people. These conjectures will be discussed in more detail in Section 7.3.8.

(b) Using Other Heuristics

For each specific example that the students specialised for the Sausage Task in order to generalise, they had to use other heuristics to solve, just like for the original task.

Table 7.22 shows the TPO for the processes and outcomes of using other heuristics for the 10 students' extension of the Sausage Task (the reader should refer to Section 7.3.3 to be familiar with the three reasoning processes and the three cutting methods that the students had used for the original task). There is a need to include a new metacognitive behaviour, incubation (MI), in this table. Although all the students extended the task, not all of them did it to find the least number of cuts.

Table 7.22 Using Other Heuristics for Extension of Sausage Task

	Processes						Outcomes				
	Reasoning using			VI	MF	MI	Cutting Methods				
	Fraction	LCM	Factors				A	B	C	Others	Total
S1	✓	✓		2	1		✓	✓		1	**3**
S2				4			✓		✓		**2**
S3				2			✓				**1**
S4	✓										**0**
S5	✓			2			✓		✓		**2**
S6				4						1	**1**
S7		✓	✓	1	1	3					**0**
S8	✓			9				✓		3	**4**
S9	✓			10				✓			**1**
S10	✓			5				✓			**1**
Total	**6**	**2**	**1**	**39**	**2**	**3**	**4**	**4**	**2**	**5**	**15**

* Although S9 also exhibited metacognitive awareness (MA) in this stage, it was not included in this table because it would be analysed in the checking stage in Section 7.3.10 since it had something to do with him sensing something amiss and checking his working.

Table 7.22 shows that 2 students (S4,S7) did not use any cutting method for their extension at all. This was because one of them (S4) was no longer interested in finding the least number of cuts but the fraction of sausage each person will receive in order to generalise, while the other one (S7) did not know how to cut the sausages: she tried using the LCM and factors of the number of sausages and the number of people, which led her nowhere. It was observed that 4 students used Cutting Method

A, 4 students used Method B and 2 students used Method C for their extension. A check with Table 7.16 in Section 7.3.3 shows that these students had used the same method(s) for the original task. This means that the students just applied the same method(s) for the original task to different examples in their extension, but it was not a straightforward application, as explained below.

For example, one student (S1) was initially clueless about how to cut 4 sausages to share them equally among 10 people in his extension although he had discovered Methods A and B when solving the original task. So he began by cutting each sausage into 2 equal parts (which falls under 'Others' in Table 7.22), and then realised that there were not enough pieces to share. It took him quite a while (about 5 minutes) to refer back to his solution for the original task before he discovered how to solve his extension: he found the LCM of 4 and 10 to give 20, which led him to cut each of the 4 sausages into 5 equal parts, so that there will be a total of 20 pieces to be shared equally among the 10 people (Method A). As discussed earlier in Section 7.3.3, finding the LCM will naturally lead to Method A. He then found the fraction of sausage each person will receive to be 2/5, which led him to cut each sausage twice: at the 2/5-mark and the 4/5-mark (Method B). As discussed in Section 7.3.3, there is a need to find the fraction of sausage each person will receive in order to apply Method B, which gives the least number of cuts at 8 in this case. Similarly, 2 other students (S9,S10) also found the fraction of sausage each person will receive before applying Method B to find the least number of cuts for their extension. However, one student (S8) was unable to see beyond the specifics (i.e. cut at the 2/3-mark) to the general idea (i.e. cut at the fraction of sausage each person will receive) behind Method B, so she ended up cutting all the sausages at the 2/3-mark for 7 of her examples in her extension, and it did not work in most cases.

Table 7.22 shows that 9 students visualised information (VI) by drawing a diagram in this stage. Only S4 did not draw any diagram because he wanted to generalise the fraction of sausage each person will receive, so there was no need for him to visualise how to cut the sausage. Similar to the data analysis in Section 7.3.3 on visualising information when solving the original task, the students drew a diagram during extension for the same three purposes: (i) to think of a cutting method, (ii) to try out a cutting method that they had thought of to see if it works, and (iii) to present the solution. Similar to the findings in Section 7.3.3, visualising information did not help the students to think of a cutting method during extension, but it was more useful in helping the students test whether a cutting method would work.

It was further observed from Table 7.22 that only 2 students (S1,S7) analysed the feasibility of their plan to use other heuristics (MF). As discussed above, the first student (S1) was having difficulty thinking of a cutting method for his extension of sharing 4 sausages equally among 10 people, so he referred back to Cutting Method A for the original task. He then analysed the feasibility of using the same method for his extension, but realised that something was missing:

> "Um … that the above cutting 18 among 12 … Okay, yes, it may be work, it may work very … very well. But I need to find the formula first before I go to the different cases, yah." [S1; Sausage Task]

This was because he did not find the LCM when using Method A to cut the sausages for the original task, so he was unable to apply the idea behind Method A to the extension. At first, he thought that the 'formula' had something to do with multiples of 10. In the end, after struggling for another 2 minutes, he realised that the total number of sausages after cutting had to be a multiple of 10, so he discovered the LCM of 4 and 10, which enabled him to apply Method A to his extension. In other words, analysing the feasibility of his plan to use the same cutting method as the original task for his extension had helped him to solve his extension. But MF was not helpful for the other student (S7) because it led her nowhere: she analysed the feasibility of using multiples or factors to think of a cutting method for her extension, just like for the original task in Section 7.3.3, but both multiples and factors did not help her.

S7 also incubated (MI) 3 times in this stage[21]. These were the only instances in the pretest and the posttest that a student engaged in MI. Incubation is a stage of creativity in Wallas' Creativity Model (see literature review in Section 2.2.2e) where a person takes a break from getting stuck in problem solving and relaxes the mind: think about the problem in a more relaxed state and environment and let the images from the subconscious surface. It was taught to the students during Lesson 6 of the teaching experiment (see Appendix C), but incubation and thinking aloud are diametrically opposite: you cannot incubate when you are thinking aloud. So it was surprising that the student tried to incubate during the test. She incubated for only 3 seconds during the first instance, 6 seconds during the second instance 13 minutes later, and another 6 seconds during the last instance 3 minutes later. Her incubation period was very short because she needed to think aloud. Therefore, her incubation was not effective. The following shows her protocols during the second incubation period:

> "I think, perhaps, I should let my mind rest for a while, then maybe an idea will just pop up. [Pause 6 s]" [S7; Sausage Task]

---

[21] Incubation was coded as XC (Unable to Code) in the protocols because this behaviour was not previously observed in the other posttest transcripts used to develop the Final Coding Scheme.

Summary

To summarise, none of the students in the present study specialised systematically to try to find a formula for the least number of cuts. Most of them specialised randomly and it was not effective in general, although one student did discover the formula for the least number of cuts by just looking at two random examples. Three students also specialised purposefully for different purposes. Most of the students used the same cutting method(s) that they had used to solve the original task for the examples in their extension, but it was not easy for some students to apply the same method(s) directly because they did not understand the underlying idea behind the method(s). One student also invented three other cutting methods, which proved to be special cases of a more general cutting method to share $6n$ sausages equally among $6n + 6$ people. Only 2 students analysed the feasibility of their plan, which was useful only for one of them. One student also incubated 3 times, but it was not effective.

7.3.8 Conjecturing (Stage 4) for Extension of Task

Table 7.23 shows the TPO for the conjecturing processes and outcomes for the 10 students' extension of the Sausage Task. The conjectures were classified as trivial or non-trivial based on the task analysis in Appendix E and the classification had passed the inter-coder reliability test (see Section 5.4). Only 5 students were shown as they were the only ones who formulated conjectures for the extension. None of the 10 students analysed the feasibility of their plan (MF) or exhibited any metacognitive awareness (MA) in this stage. It was observed that the 5 students formulated a total of 7 conjectures, out of which 4 of them were non-trivial and 5 of them were correct. Only one student (S9) formulated 2 conjectures to find a general formula for the least number of cuts while 2 other students (S2,S4) formulated a conjecture to find a general formula to share *n* items equally among *m* recipients. The last 2 students (S3,S8) formulated a total of 3 conjectures to generalise a cutting method.

Table 7.23 Conjecturing Processes and Outcomes for Extension of Sausage Task

	Processes		Outcomes		
	MF	**MA**	**Conjecture**	**Trivial or non-trivial?**	**Correct or wrong?**
S2			Conjecture 2*	Trivial	Correct: general formula to share *n* items equally among *m* recipients**
S3			Conjecture 1	Non-trivial	Correct: generalised Cutting Method A
S4			Conjecture 1	Trivial	Correct: generalised fraction of sausages that each person will receive**
S8			Conjecture 2	Trivial	Wrong: generalised cutting method to share 18*n* sausages equally among 24 people
			Conjecture 3	Non-trivial	Correct: generalised cutting method to share 6*n* sausages equally among 6*n* + 6 people
S9			Conjecture 2	Non-trivial	Wrong: general formula for least number of cuts, but made a mistake in the number of sausages in the conjecture
			Conjecture 3	Non-trivial	Correct: general formula for least number of cuts
Total	**0**	**0**	**7 conjectures**	**3 trivial; 4 non-trivial**	**5 correct; 2 wrong**

* Conjecture 2 means the student had formulated Conjecture 1 for the original task.
** These two results were clearly true (i.e. no need to prove) but the students treated it as a conjecture.

From his protocols, it was observed that S9 extended the task by sharing 10 sausages equally among 18 people, but he solved the problem wrongly for only 5 sausages, so his Conjecture 2 based on these 5 sausages was false. After he had discovered the mistake, he was able to formulate his Conjecture 3 on the general formula for the least number of cuts for sharing x identical sausages equally among y people, which is $y - \text{HCF}(x,y)$. His full investigation had already been described in Section 6.3(a).

The two conjectures from S2 and S4 were actually very trivial because they were clearly true, but the two students actually tested their conjecture using naïve testing. The first student (S2) generalised the amount of sausages each person will receive by using the formula of dividing the number of items to be shared by the number of recipients. He even tested the formula on a different context: how to share the usage of a computer with 9 brothers over a week. In the end, he concluded:

> "I think it does not only work on food stuff … [start writing] works on other things with numerical value [stop writing]." [S2; Sausage Task]

The other student (S4) decided to test what he called the 'relationship ratio', i.e. the ratio of the number of sausages to the number of people, on three examples: he randomly chose 68 sausages and 42 people for his first example, purposefully chose 9 sausages and 3 people for his second example so that the 'relationship ratio' will be whole numbers, and randomly chose 18 sausages and 12 people for his third example by swapping the two given numbers in the original task. In the end, he verified that the 'relationship ratio' worked. It is indeed puzzling why the two students (S2,S4) took so long to develop this simple formula for the amount of sausages each person will receive because the formula could easily be obtained by simple reasoning which they had learnt in primary schools. For example, a common question in primary school mathematics on fractions is:

> 12 identical sausages are shared equally among 18 people. Find the fraction of sausage that each person gets.

Since these two students were from a high-performing secondary school, they would have solved this question easily when they were in primary schools: the answer is just the number of sausages divided by the number of people, which is 12/18, or 2/3. So it came as a surprise when the two students took such a long time to develop the general formula of dividing the number of sausages by the number of people, and they even had to test it on some examples to be more certain of the formula. Therefore, by opening up this kind of questions into open investigative tasks, the two students were thrown off balance and they were unable to do a simple direct application of what they had learnt in primary schools.

The last 2 students (S3,S8) formulated a total of 3 conjectures to generalise a cutting method. One of them (S8) started with the intention to generalise Cutting Method B by finding out whether the method would work for which numbers of sausages and people. Along the way, she realised that Method B did not work most of the time, so she tried another cutting method (her 7^{th} one for the investigation) and generalised the method to share $18n$ sausages equally among 24 people, but her Conjecture 2 was false. Finally, she formulated her non-trivial Conjecture 3 on a general cutting method to share $6n$ sausages equally among $6n + 6$ people. Her two conjectures had already been described in detail in Section 7.3.7(a) on page 356. The other student (S3) tried to generalise Cutting Method A for x sausages and y people. He did not use the LCM of 12 and 18 to help him find Method A for the original task: instead he just cut each sausage into 2 equal parts, which did not work out; so he just cut each sausage into 3 equal parts (Method A) and it worked. Then he extended the task by finding a method to cut the 12 sausages to share them equally among 30 people. After drawing 12 sausages, he just cut each sausage into 5 equal parts and found that there were a total of 60 parts, which is divisible by 30, so each person will receive 2 parts. Then he realised that he needed to find the LCM:

> "Ok, it seems that, um … if I've a certain number of sausages and a certain number of people, and I want to share the sausages equally among the number of people, I've to find the, um … lowest common multiple." [S3; Sausage Task]

So he decided to generalise Cutting Method A for *x* sausages and *y* people. Using the LCM of 12 and 30, which is 60, as an example, he reasoned that the number of parts that each sausage should be cut is the LCM of the two numbers, *x* and *y*, divided by the number of sausages, *x*. He also reasoned that the number of parts of sausages that each person will get is the total number of parts divided by the number of people, *y*.

7.3.9 Justifying and Generalising (Stages 5 and 6) for Extension of Task

Table 7.24 shows the TPO for the justifying and generalising processes and outcomes for the 10 students' extension of the Sausage Task. Only 5 students were shown because they were the only ones who had formulated conjectures for the extension. The other 5 students did not analyse the feasibility of their plan (MF) and none of the 10 students exhibited metacognitive awareness (MA) in these stages.

Table 7.24 Justifying / Generalising Processes and Outcomes for Extension of Sausage Task

	Conjecture	Justifying Processes**				Justifying and Generalising Outcomes
		Naïve Testing	Non-proof Argument	Formal Proof	MF	
S2	Correct Trivial Conjecture 2*	✓				Accepted conjecture as true based on naïve testing
S3	Correct Non-trivial Conjecture 1	✓				Accepted conjecture as true based on naïve testing
S4	Correct Trivial Conjecture 1	✓				Accepted conjecture as true based on naïve testing
S8	Wrong Trivial Conjecture 2					Accepted conjecture as true without testing
	Correct Non-trivial Conjecture 3	✓	✓			Tried but failed to justify conjecture
S9	Wrong Non-trivial Conjecture 2		✓		1	Tried but failed to justify conjecture
	Correct Non-trivial Conjecture 3	✓	✓			Tried but failed to justify conjecture
Total	**5 correct; 2 wrong**	**5**	**3**	**0**	**1**	**0 proven; 0 generalisation*** **

* Conjecture 2 means the student had formulated Conjecture 1 for the original task.
** The students did not exhibit metacognitive awareness (MA) in this stage.
*** There was no generalisation either because the conjecture was wrong or it had not been proven.

Table 7.24 shows that 3 of the 5 students accepted their conjecture as true based on naïve testing while one of them (S8) accepted one of her two conjectures as true without testing. However, the latter did try to justify her other conjecture by using a non-proof argument in the middle of naïve testing using three examples, but failed. The last student (S9) also tried to justify his two conjectures using a non-proof argument but failed (his full investigation had already been described in Section 6.3a). It was observed from Table 7.24 that S9 also analysed the feasibility of his plan to justify his conjecture (MF):

> "But I don't know how to prove it. I think it is a little off my limits. How do I prove this conjecture? The conjecture is uh, a little bit pretty much complicated. Eh, no, the conjecture is not complicated, but is very hard to prove." [S9; Sausage Task]

In the end, he still decided to try to prove it by using a non-proof argument, but he was unable to explain why the formula worked[22]. No student used formal proofs. Although 5 of the 7 conjectures formulated were correct and were general results, none of the students was able to prove any of them, so there was no generalisation.

7.3.10 Checking (Stage 7)

Table 7.25 shows the TPO for the checking processes and outcomes for the 10 students' investigation of the Sausage Task. As explained in Section 7.2.6, it is more appropriate to analyse all the checking and monitoring processes and outcomes, including metacognitive awareness (MA) where students sensed something amiss[23],

---

[22] The students were not expected to prove this formula because the proof might be beyond them (see the proof in the task analysis of this task in Appendix E).

[23] Metacognitive awareness (MA) that helped students to be aware of what they were doing, which might have helped them save time by not repeating what they had done before, were dealt with in the stages that it had occurred, e.g. in Section 7.3.3 for the Sausage Task.

together in the same section although some of them could occur in other stages. The classification of errors made as major or minor had been explained in Section 7.2.6.

Table 7.25 Monitoring / Checking Processes and Outcomes for Sausage Task

	Processes								**Outcomes**	
	Check Working					**MP**	**MR**	**MA**	**Errors Made**	**Errors Discovered**
	Most Parts	**Some Parts**	**Glance Briefly**	**Others**	**Total**					
S1				1	**1**	1	1		1 minor	0
S2			1	2	**3**		2		3 minor	0
S3					**0**		1		0	0
S4	1				**1**		1		1 major + 3 minor = 4	2 minor
S5			4		**4**		3		0	0
S6					**0**				2 major + 1 minor = 3	1 minor
S7		1		1	**2**	9			6 minor	3 minor
S8				1	**1**			1	2 major + 3 minor = 5	1 major (2 min later) + 1 minor = 2
S9		2			**2**		2	1	1 major + 3 minor = 4	1 major (4 min later) + 3 minor = 4
S10		2	1	1	**4**		2		0	0
Total	**1**	**5**	**6**	**6**	**18**	**10**	**12**	**2**	**6 major + 20 minor = 26**	**2 major + 10 minor = 12**

* Errors due to misinterpretation had been dealt with in Table 7.13, and errors due to accepting conjectures as true without testing or based on naïve testing had been dealt with in Tables 7.18 and 7.24, so these errors were omitted from this table. Formulating incorrect conjectures, as discussed in Tables 7.17 and 7.23, were *not* errors.

** The time indicated in brackets for 'Errors Discovered' refers to the time interval between making the major error and discovering the error. No time is indicated for the discovery of a minor mistake because the discovery time is not an important factor when the mistake is minor.

Table 7.25 shows that 8 students checked their working on 18 occasions, or between 1 to 4 times each. Only 2 students (S3,S6) did not appear to check their working at all. Unlike the Kaprekar Task discussed in Section 7.2.6 where the frequencies for the three processes of checking (most parts, some parts and glance briefly) happened to be the same and no student checked their working by other means such as checking backwards, the students seldom checked most parts of their working step by step for the Sausage Task, and some of them also checked their working backwards. For example, the following shows the protocols of a student (S1) who checked his working backwards for using Cutting Method B for the original task by verifying that it was possible to share the 12 sausages equally among the 18 people from the fact that each one of them will receive 2/3 of a sausage:

> "When you cut, there's 12 people already have 2/3 of the … 2/3, that means that 12 people have … their sausages, left with 6 people who don't have, and the remaining 12 1/3's is 6 2/3's that means that 6 people, yah that means that 6 people will also have their sausages. And everybody share the equal amount of the sausages, that is 2/3 of one piece… okay. [Turn to p. 2] It's done, this question." [S1; Sausage Task]

The issue is which checking processes were effective in helping the students discover their mistakes. A detailed analysis of their protocols suggests that metacognitive awareness (MA), which includes the ability to sense something amiss, had played a more important role than any of the checking processes in helping the students to discover their errors, just like the Kaprekar Task discussed in Section 7.2.6. For example, one student (S8) was using Cutting Method B for the Sausage Task as shown in Figure 7.10: she had divided each sausage at the 2/3-mark in order to share the 12 sausages equally among the 18 people. But she verbally grouped the last 1/3 part of every *three* sausages to give to each person, when she should have grouped the 1/3 part of every *two* sausages since each person receives 2/3 of a sausage. Thus she could only share the 12 sausages among 16 people and so she wrongly believed that Cutting Method B failed. Later, she cut each sausage into 3 equal parts (Cutting

Method A) as shown in the same figure. She then grouped the first two 1/3 parts of the first 6 sausages to give to 6 people by circling them. Then she grouped the last 1/3 part of the first *two* sausages to give to the 7th person. As she grouped the last 1/3 part of the third and fourth sausages for the 8th person, she paused halfway as she sensed something amiss. After grouping the last 1/3 part of the fifth and sixth sausages for the 9th person, she concluded that she had earlier counted wrongly for Cutting Method B, i.e. she should have grouped the last 1/3 part of every *two* sausages instead of every three sausages. Therefore, her metacognitive awareness in sensing something was amiss had helped her to discover her first major error.

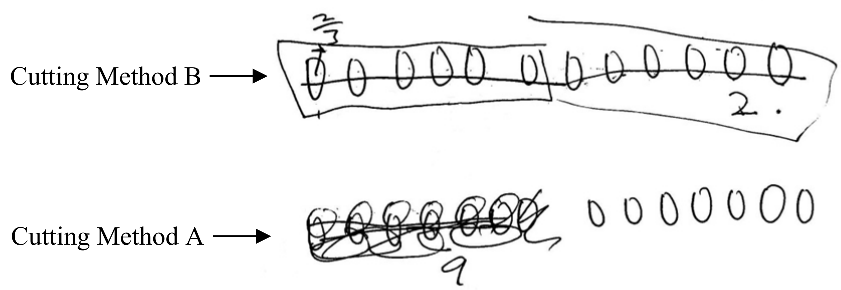


Figure 7.10 Discovery of Error by S8 for Sausage Task

Table 7.25 shows that 7 students reviewed their solution (MR) for the Sausage Task on 12 occasions. For example, one student (S1) reviewed his solution to check whether it had met the goal of finding the least number of cuts. Then he used a non-proof argument to reason why 12 was the least number of cuts (his argument had been discussed in Section 7.3.5 on page 344), which he should have done earlier. Another student (S9) reviewed his solutions for the first three problems that he had posed, and realised that the solution for his second problem was actually to explain why the solution for his first problem worked, so he concluded, “Problem 2 is actually not really called a problem.” The same student later reviewed his solution for his first extension and realised that the result “is still a conjecture” (in his own words), i.e. he had not proven the conjecture yet. Three of the 7 students (S2,S3,S5) also evaluated the efficacy of their method of solution and looked for alternative methods. For example, one student (S5) reviewed her solution for the original task and concluded that her cutting method was “fair” (in her own word). Then she realised that she could look for alternative cutting methods, which led her to amend her original problem of finding how to share the 12 sausages equally among the 18 people to include finding how many cutting methods and evaluating which cutting method is better.

Table 7.25 shows that only 2 students exhibited overt signs of monitoring progress (MP) during their investigation, for a total of 10 times. One of them (S7) monitored her progress on 9 occasions because she was stuck, but just like the Kaprekar Task described in Section 7.2.6, her metacognitive behaviour for the Sausage Task was not effective because she did not know what else to do other than to continue in the same direction, which was to use factors and multiples to help her think of a suitable cutting method. She did not realise that the critical factor in cutting the sausages was the fraction of sausage each person will receive, not factors and multiples. It was further observed that the students monitored their progress a lot more often for the Kaprekar Task (39 times) than the Sausage Task (10 times). This was because the students had more trouble finding an underlying pattern for the Kaprekar Task than solving a problem in the Sausage Task. For example, one student (S8) monitored her progress for the Kaprekar Task 10 times because she had difficulty observing the underlying patterns, but she was able to solve some specific problems for the Sausage Task and so she did not pause to monitor progress at all.

Summary

To summarise, metacognitive awareness (MA), which includes the ability to sense something amiss when a questionable result is obtained, had played a more important role than the checking processes in helping the students to discover their mistakes. But unlike the Kaprekar Task, some students also checked their working backwards, evaluated the efficacy of their method of solution, and looked for alternative methods for the Sausage Task. The students also monitored their progress a lot less often for the Sausage Task than for the Kaprekar Task because they were able to progress in the Sausage Task and so most of them did not pause to monitor their progress.

7.4 SUMMARY OF ANSWER TO RESEARCH QUESTION 2

This section will summarise the main findings from analysing the data collected for the present study in order to answer Research Question 2. Section 7.4.1 will sum up the data analysis of the processes and outcomes of the 10 students when they attempted the Kaprekar Task (Type A) and the Sausage Task (Type B): whether these processes had helped them to produce significant outcomes in their investigation, such as posing the intended problem and formulating non-trivial conjectures. Section 7.4.2 will use the empirical data collected to validate and refine the two theoretical investigation models for cognitive and metacognitive processes.

7.4.1 Processes and Outcomes of Mathematical Investigation

(a) Understanding the Task

Understanding the task correctly is very important. If students misinterpret the task, they will end up on a wild goose chase (Schoenfeld, 1987). Some students in the present study had wasted precious time because they misinterpreted the task. If only they had spent a bit more time trying to understand the task at the beginning, they would have saved a lot of time if they had embarked on the correct track right from the start of the investigation. The main understanding processes that had helped the students in the present study to interpret both types of tasks correctly, or to recover from any misinterpretation, were the re-reading or the rephrasing of the *relevant* parts of the task statement. The difference between understanding the two types of tasks is that the students needed to spend a lot more time to understand the Kaprekar Task (Type A) by trying examples, while some students understood the Sausage Task (Type B) correctly by reading the task only once. The metacognitive process of monitoring the understanding did not seem to be helpful.

(b) Problem Posing

The ability to pose problems to solve was an integral part of investigation, but it affected Type B tasks more than Type A tasks since students could just search for any pattern for Type A tasks but they had to pose specific problems to solve for Type B tasks. For the Kaprekar Task (Type A) in the present study, most of the students did not see the need to pose the general problem of searching for any pattern explicitly, but they just went ahead to search for patterns. One student even used a conjecture as a springboard to pose two non-trivial specific problems with analogous results. This is called problem posing by analogy, which was advocated by Kilpatrick (1987) and discussed in detail in Section 2.2.3(h). For the Sausage Task (Type B), only 5 students were able to pose the intended problem of finding the least number of cuts. This suggests that teachers could not just design an investigative task by removing the intended problem from a mathematical problem, as proposed by Frobisher (1994), without considering the possibility of 'losing' the intended problem that they might want their students to solve. Thus there is a need for teachers to think about how to guide their students to pose the intended problem, and yet not close up the task by restricting the students' freedom to pose other types of problems to solve.

Unlike understanding the task where re-reading or rephrasing the task statement had helped most students to interpret the task correctly, most of the specific problems posed for both types of tasks were the result of referring to their current working, previous results, or the given checklist of investigation processes (see Appendix H). There were some exceptions. For students who were able to pose specific problems naturally for the Sausage Task, referring to the task statement was enough for them to think of their first problem, but for subsequent problems, their current working or previous results were still a fertile source of new problems. But for those who struggled to think of specific problems to pose for the Sausage Task, referring to the task statement or the checklist of investigation processes had helped some of them to pose a problem eventually. None of the students analysed the feasibility of the goal, which might have helped prevent some of them from pursuing goals that were too trivial or too difficult to achieve.

(c) Extension

Students are not expected to extend a Type A task by changing the given because it will usually produce a completely new task with different patterns, but they are expected to extend a Type B task in order to generalise. Most of the students in the present study were unable to pose the intended extension of finding a general formula for the least number of cuts for the Sausage Task (Type B) because of two reasons:

half of them did not even find the number of cuts for the original task, and some of them did not fully understand that the intention of changing the variables in the task statement is to generalise. Instead, they extended the task to find a general formula for the amount of sausages each person will receive, or to find a general cutting method. Most of the students also encountered greater difficulty in extending the task than in posing problems for the original task, but all of them actually extended the task in one way or another. Just like the findings in the problem-posing stage, referring to the task statement, current working, or previous results, had helped some students to extend the task naturally, but for those who struggled to extend the task, referring to the task statement, previous results, or the given checklist of investigation processes, had helped some of them to extend the task eventually. There was only one instance when a student analysed the feasibility her goal to extend the task, but it was not effective.

(d) Specialising

Specialising is one of the four main mathematical thinking processes (Mason et al., 1985). For Type A tasks, students need to specialise to look for patterns, but for Type B tasks, students need to specialise only later when they extend the task to generalise. For both the Kaprekar Task (Type A) and the Sausage Task (Type B), most of the students in the present study specialised randomly instead of systematically to search for patterns. As a result, some students were unable to observe the underlying patterns for the Kaprekar Task as they did not generate all the types of sequences, and most students did not discover the general formula for the least number of cuts for the Sausage Task although one student did discover the formula just by looking at two random examples. Some students were also able to choose examples purposefully to search for any pattern or to be more certain of a pattern. There were few instances of metacognitive behaviours in this stage and most of them were not effective. However, metacognitive awareness had helped one student save a lot of time, but the lack of it had resulted in two students repeating examples that they had tried earlier.

(e) Using Other Heuristics

Using other heuristics is mainly for Type B tasks, where the students can solve the problems posed for the original task or the extended task using heuristics other than specialising, such as reasoning and visualising information. For the Sausage Task (Type B) in the present study, some students were able to reason 'effectively' by using critical information, such as the fraction of sausage each person will receive, to find the shortest method, while others were not able to do so. Some of them did not even try to analyse whether their cutting method was feasible or worth pursuing, but they just went ahead to cut the sausages any old how. Most of the students used the same cutting method(s) that they had used to solve the original task for the examples in their extension, but it was not easy for some students to apply the same method(s) directly because they did not understand the underlying idea behind the method(s). On the other hand, visualising information by drawing a diagram did not help the students to think of a workable cutting method, but it was more helpful for them to check whether their cutting method would work. Metacognitive awareness had also helped one student not to repeat a cutting method that she had tried earlier. Only 2 students analysed the feasibility of their plan, which was useful only for one of them. One student also incubated 3 times, but it was not effective.

(f) Conjecturing

Conjecturing is one of the four main mathematical thinking processes (Mason et al., 1985). For Type A tasks, students need to specialise in order to search for patterns so as to formulate conjectures, but for Type B tasks, students can solve some specific problems with or without formulating any conjecture. For the Kaprekar Task (Type A) in the present study, all the students searched for patterns within a sequence, but none of them searched for patterns across sequences because they did not analyse where to search for patterns. As a result, they were unable to observe patterns that exist across sequences, such as self numbers. On the other hand, generating all the representative sequences and searching for patterns within each sequence did not help most of the students to observe the Types 1 and 2 patterns as these patterns were not the usual types of patterns that they had encountered. However, two students were able to actively apply their knowledge, or what Schoenfeld (1985) called 'resources', to discover the Type 1a 'multiples' pattern or the complicated Type 2 'digital roots' pattern (see the task analysis in Appendix E for a description of these patterns).

For the Sausage Task (Type B), only 5 of the 10 students posed the intended problem of finding the least number of cuts for the original task, but only 4 of them were able to use reasoning 'effectively' to formulate the correct conjecture that the least number of cuts is 12. For the extension, only 2 students posed the intended extension of finding a general formula for the least number of cuts for sharing x identical sausages equally among y people, but only one of them was able to formulate the correct conjecture that the formula is $y - \text{HCF}(x,y)$. Four other students also formulated at least one conjecture each, but their conjectures were different in nature, depending on the types of extensions that they had posed. Two of them formulated a conjecture about a formula for the fraction of sausage each person will receive, but the result was clearly true, meaning that it should not be a conjecture in the first place. The last two students formulated a conjecture about a general cutting method, e.g. one of them found a cutting method to share 6*n* sausages equally among 6*n* + 6 people, which is to cut each sausage at the $\frac{n}{n+1}$ -mark.

(g) Justifying

Justifying is one of the four main mathematical thinking processes (Mason et al., 1985). Students need to recognise that a result obtained from specialising has to be justified because it is only a conjecture. For the Kaprekar Task (Type A), 2 of the 10 students still wrongly accepted their conjecture as true without testing, even after the teaching experiment, but none of them accepted their conjectures as true based on naïve testing. For the Sausage Task (Type B), 3 students accepted their conjecture for the original task as true without testing, and one of them also accepted her conjecture for the extension as true without testing. In addition, 3 students also wrongly accepted their conjectures for the extension as true based on naïve testing. Some students tried to prove their conjectures using a non-proof argument, with 2 of them succeeding in proving a total of 3 conjectures for the Kaprekar Task, and another 2 of them in proving their conjecture about the least number of cuts for the original Sausage Task. No student managed to prove any conjecture for the extension of the Sausage Task. Only one student attempted a formal proof using algebra (this is for the Kaprekar Task), but failed. It seems that formal proofs are beyond the level of Secondary 2 students. Only one student analysed the feasibility of proving his conjecture (this is for the general formula for the least number of cuts for the Sausage Task) and concluded that it was beyond him.

(h) Generalising

Generalising is one of the four main mathematical thinking processes (Mason et al., 1985). When a conjecture has been proven correctly, generalisation has taken place if the conjecture is a general result. For the Kaprekar Task (Type A), there were 3 generalisations when 2 of the students succeeded in proving their 3 conjectures, which are general results. In particular, one of them was the Type 1 'multiples' pattern. For the Sausage Task (Type B), there was no generalisation because the 2 proven conjectures about the least number of cuts for the original task are not general results. Although most of the conjectures for the extension of the Sausage Task are general results, the students were unable to prove them and so there was no generalisation.

(i) Checking

Checking working and monitoring progress are important processes that should occur throughout the investigation, not just at the end after proving a conjecture or solving a problem. It was found that metacognitive awareness (MA), which includes the ability to sense something amiss when a questionable result is obtained, had played a more important role than the checking processes in helping the students to discover their mistakes for both the Kaprekar Task (Type A) and the Sausage Task (Type B). It was also discovered that most of the students monitored their progress only when they were stuck. Some of them did not know what else to do when monitoring their progress, except to continue in the same direction, while others were able to monitor their progress more effectively for the Kaprekar Task by changing the approach to search for patterns elsewhere or by checking their working to see if they had made any mistake. However, the students monitored their progress a lot less often for the

Sausage Task than for the Kaprekar Task as they were able to progress in the Sausage Task and so most of them did not pause to monitor their progress. It was further observed that some students were able to review their solution effectively when they realised that their solution had not fully met the goal of the task. Some students also evaluated the efficacy of their method of solution, and looked for alternative methods for the Sausage Task.

7.4.2 Validation and Refinement of Mathematical Investigation Models

From the data analysis above, it was found that the two theoretical investigation models for cognitive and metacognitive processes have provided a fairly accurate description of the main processes and their interactions among one another according to the pathways shown in the models. However, there was a need to refine both models slightly, based on new insights gleaned from the students' investigation.

(a) Refined Investigation Model for Cognitive Processes

The Theoretical Investigation Model for Cognitive Processes developed for the present study is shown in Figure 3.1 in Section 3.2.1, while the Refined Investigation Model for Cognitive Processes is shown in Figure 7.11 in this section. As described in Section 7.2.4, the analysis of empirical data for the Kaprekar Task had revealed that all the students in the present study went from 'Specialising' to 'Searching for Patterns' and then back to 'Specialising' very often, either because they could not find a pattern and so they had to specialise some more, or they tried to search for patterns in every step of the process that they repeated for each example. Thus there was a need to include in the investigation model this new pathway of going back from

‘Searching for Patterns’ to ‘Specialising’, as indicated by the thick pathway, in the Refined Investigation Model for Cognitive Processes.

A new outcome called ‘Observed Pattern’ was also discovered and discussed in Section 7.2.4. In the theoretical model, it is posited that the patterns observed by the students will be considered as conjectures, so their protocols will be coded as ‘Formulated Conjecture’ (FC). But it was discovered that there was generally a time gap between observing a pattern and formulating it as a conjecture since the students were usually unsure of their observed pattern and so they would specialise some more to be more certain of the observed pattern first, before formulating it as a conjecture. Sometimes, they even found counter examples to reject the observed pattern before they could formulate it as a conjecture. Thus there was a need to distinguish between observing a pattern and formulating a conjecture. The Refined Investigation Model for Cognitive Processes in Figure 7.11 shows the new outcome ‘Observed Pattern’ and the corresponding pathways, which are indicated by the darker box and the thicker pathways respectively. In other words, when a student first observes a pattern, he or she might treat the observed pattern as a conjecture to be proven or refuted, or he or she might try more examples to be more certain of the pattern first (first level of testing), before formulating it as a conjecture. Then the student should try to justify the conjecture. However, the student might not know how to prove the conjecture by using a non-proof argument or a formal proof, so he or she might end up trying more examples in naïve testing to see if the conjecture could be refuted by counter examples (second level of testing). This is consistent with Frobisher’s (1994) model where he differentiated between ‘conjecturing’ and ‘hypothesising’ with two levels of testing using empirical data (see Section 2.2.2g on page 59), although most educators (e.g. Lampert, 1990; Mason et al., 1985) do not distinguish between them.

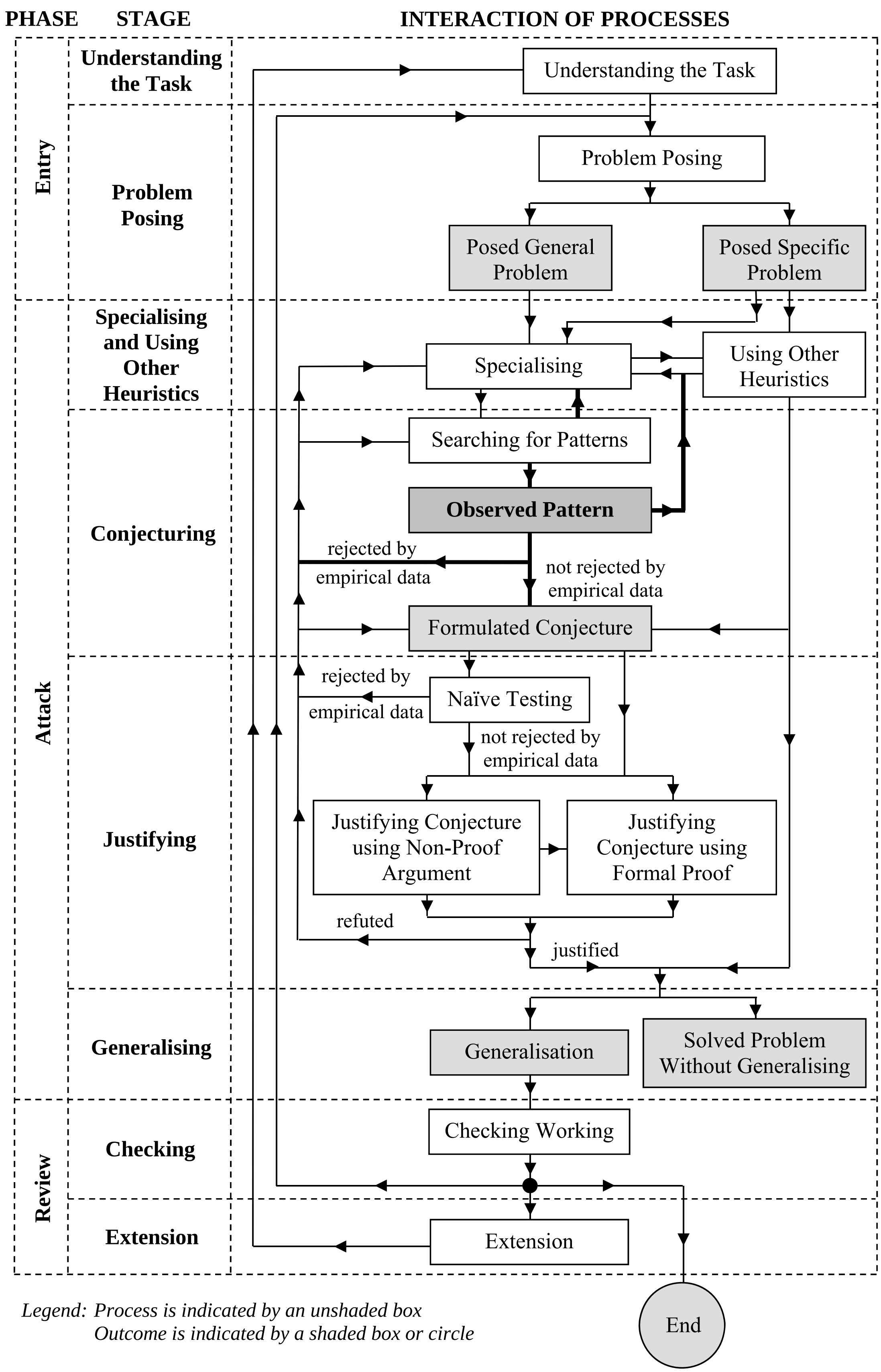


Figure 7.11 Refined Investigation Model for Cognitive Processes

(b) Refined Investigation Model for Metacognitive Processes

Table 7.26 shows a summary of the types and frequencies of metacognitive processes engaged by the students for the Kaprekar Task (Type A) and the Sausage Task (Type B) in the present study. The number in each square bracket indicates the frequency of the particular metacognitive behaviour observed when the student was attempting the task. The data for the table were obtained from the respective tables in Sections 7.2 and 7.3. However, there was a need to combine the data in some tables, e.g. the frequency for 'analysing feasibility of plan' (MF) during 'using other heuristics' for the original task in Table 7.16 was combined with the frequency for MF during 'using other heuristics' for the extension in Table 7.22. Although 'monitoring progress' (MP) and 'metacognitive awareness' (MA) were analysed under the checking stage for the Kaprekar Task and the Sausage Task in Sections 7.2.6 and 7.3.10 respectively, these metacognitive processes generally do not occur in the checking stage, so there was a need to examine the students' protocols to find out which cognitive processes that these metacognitive processes interact with.

It was observed from Table 7.26 that all 10 students engaged in at least one metacognitive process for the Kaprekar Task, but only 9 students (except S6) did so for the Sausage Task. In fact, S6 showed only one overt sign of metacognitive behaviour for the Kaprekar Task. Although S7 exhibited metacognitive behaviours most frequently with a total of 14 + 13 = 27 occasions for both tasks, her metacognition was not effective as analysed previously in this chapter.

Table 7.26 Metacognitive Processes for Kaprekar and Sausage Tasks

Metacognitive Process	Cognitive Process	Kaprekar Task		Sausage Task		Total
		Student	Sub-total	Student	Sub-total	
Monitoring Understanding (MU)	Understanding the Task	S1[1], S3[2], S4[1], S7[2], S9[2], S10[1]	6[9]	S3[2], S7[1], S9[1], S10[2]	4[6]	**6[15]**
Analysing Feasibility of Goal (MG)	Problem Posing		0		0	0
	Extension		0	S10[1]	1[1]	1[1]
	Sub-total		0	S10[1]	1[1]	**1[1]**
Analysing Feasibility of Plan (MF)	Specialising	S2[1], S5[2]	2[3]		0	2[3]
	Using Other Heuristics		0	S1[1], S7[2]	2[3]	2[3]
	Conjecturing		0		0	0
	Justifying / Generalising		0	S9[1]	1[1]	1[1]
	Sub-total	S2[1], S5[2]	2[3]	S1[1], S7[2], S9[1]	3[4]	**5[7]**
Monitoring Progress (MP)	Specialising	S2[1], S7[3], S8[1]	3[5]		0	3[5]
	Using Other Heuristics		0	S7[9]	1[9]	1[9]
	Conjecturing	S2[3], S3[4], S5[5], S7[8], S8[9]	5[29]	S1[1]	1[1]	6[30]
	Justifying / Generalising	S9[5]	1[5]		0	1[5]
	Sub-total	S2[4], S3[4], S5[5], S7[11], S8[10], S9[5]	6[39]	S1[1], S7[9]	2[10]	**7[49]**
Reviewing Solution (MR)	Checking	S5[2], S9[2]	2[4]	S1[1], S2[2], S3[1], S4[1], S5[3], S9[2], S10[2]	7[12]	**7[16]**
Metacognitive Awareness (MA)	Specialising	S2[2], S4[1], S6[1], S7[1]	4[5]		0	4[5]
	Using Other Heuristics		0	S7[1], S8[1], S9[1]	3[3]	3[3]
	Conjecturing	S1[1], S5[2], S9[2], S10[4]	4[9]		0	4[9]
	Justifying / Generalising		0		0	0
	Sub-total	S1[1], S2[2], S4[1], S5[2], S6[1], S7[1], S9[2], S10[4]	8[14]	S7[1], S8[1], S9[1]	3[3]	**9[17]**
Metacognitive Processes	**Total**	S1[2], S2[7], S3[6], S4[2], S5[11], S6[1], S7[14],S8[10], S9[11], S10[5]	**10[69]**	S1[3], S2[2], S3[3], S4[1], S5[3], S7[13], S8[1], S9[5], S10[5]	**9[36]**	**10[105]**

* The number in each square bracket indicates the frequency of the particular metacognitive behaviour observed when the student was attempting the task.

Table 7.26 shows that the 10 students exhibited about twice as many metacognitive behaviours for the Kaprekar Task as those for the Sausage Task. The main bulk came from ‘monitoring progress’ (MP), which occurred 39 times for the Kaprekar Task, compared with only 10 times for the Sausage Task. As analysed earlier in Sections 7.2.6 and 7.3.10, the students monitored their progress a lot more often for the Kaprekar Task because they were stuck, but most of them were able to progress more smoothly for the Sausage Task and so they paused less often to monitor their progress. It was observed that the two metacognitive processes least engaged in by the students were ‘analysing feasibility of goal’ (MG) and ‘analysing feasibility of plan’ (MF). In fact, MG occurred only once when a student (S10) analysed the feasibility of her goal during the extension of the Sausage Task (see Section 7.3.6).

It was discovered that the interactions between some metacognitive processes and their corresponding cognitive processes as posited in the Theoretical Investigation Model for Metacognitive Processes (see Fig. 3.2 in Section 3.2.2) were not observed for the students in the present study (indicated by 0 in Table 7.26), although it does not mean that these interactions are not possible. Figure 7.12 shows the Refined Investigation Model for Metacognitive Processes based on the data analysis above. A dotted arrow from a metacognitive process to a cognitive process indicates that the interaction had not been observed in the current study, while a solid arrow indicates that the interaction had been observed. The shaded box indicates that the metacognitive process was not hypothesised in the theoretical model, but had been observed in the present study. The thick arrows indicate the interactions between the newly found metacognitive process and the various cognitive processes. For example, the students were supposed to analyse the feasibility of their goal (MG) and their plan

(MF), but none of them engaged in MG in the problem-posing stage, and in MF in the conjecturing stage (indicated by the dotted arrows). Thus there was a need to pay more attention to develop these two metacognitive processes in the students.

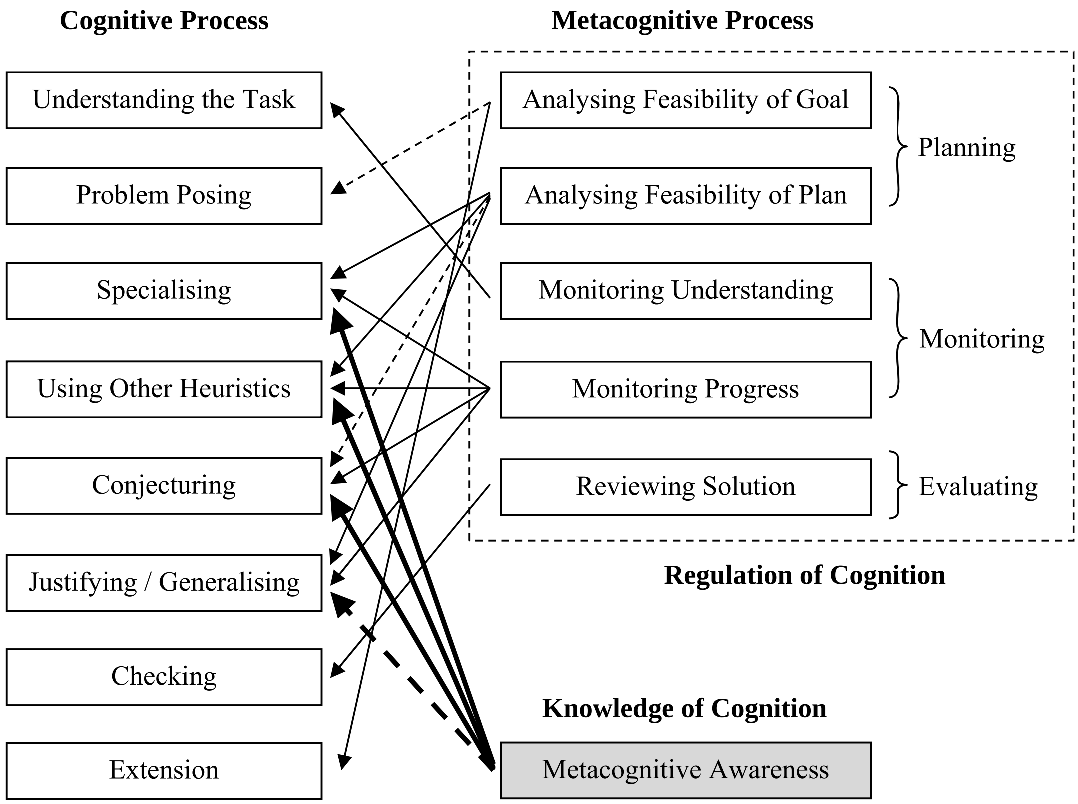


Figure 7.12 Refined Investigation Model for Metacognitive Processes

One of the significant findings in this study was the discovery of metacognitive awareness (MA), as shown in Figure 7.12. In literature on metacognition reviewed in Section 2.3.1, 'knowledge of cognition' or 'metacognitive awareness' was usually portrayed as something that was more passive than active (e.g. Desoete & Veenman, 2006; Schraw & Moshman, 1995). That was why most research studies on knowledge of cognition (e.g. Mevarech & Fridkin, 2006; Wong, 1989) used a paper-and-pencil test to try to understand the knowledge in the minds of the subjects. Since the focus of the present research was on how students regulated their cognition during investigation, my original intention was not to study the students' knowledge of cognition. But in the course of the analysis of the data collected for the present research, it was discovered that some students exhibited a more active awareness of their cognition, rather than a passive knowledge of cognition. This metacognitive awareness had proven to be useful to help some students discover their mistakes when they sensed something amiss, or to save time by being aware that they had generated a sequence or tried a cutting method before, and so they should not redo the investigation.

Figure 7.12 shows that MA had manifested itself during specialising, using other heuristics and conjecturing during the investigation of the two tasks. Although MA was not observed during justifying and generalising in the present study, it does not mean that there is no interaction between this metacognitive process and the cognitive processes of justifying and generalising. On the contrary, more research needs to be done to study the possibility of this interaction.

7.5 CONCLUDING REMARKS FOR THIS CHAPTER

Chapter 7 has answered Research Question 2 on the effect of the cognitive and metacognitive processes of Secondary 2 students on the outcomes of their investigation. Certain processes that had helped the students to produce significant outcomes had been identified. The two theoretical investigation models developed for the present study had also been validated and refined by the empirical data collected for the current research. Chapter 8 will then answer Research Question 3 on the development of investigation processes.

CHAPTER 8: DATA ANALYSIS OF DEVELOPMENT OF MATHEMATICAL INVESTIGATION PROCESSES

In this chapter, the 40 sets of students' thinking-aloud protocols and answer scripts obtained from the pretest and the posttest in the present research will be analysed to study the development of investigation processes of the 10 Secondary 2 students across the two types of investigative tasks in order to answer Research Question 3. There is also a need to develop the fourth data analysis tool for the current study, the Investigation Scoring Rubric (ISR), to aid in the analysis.

8.1 THE THIRD RESEARCH QUESTION

Research Question 3 is reproduced below:

RQ3: What is the effect of the teaching experiment on the development of Secondary 2 students' mathematical investigation processes?

Scope of Data Analysis

This chapter will study the development of mathematical investigation processes of the 10 students via two methods. The first method is to assign a score to the test of the students using an Investigation Soring Rubric (ISR) and then compare the test scores of the pretest and the posttest for the two types of tasks quantitatively using descriptive statistics (Section 8.2). The second method is to follow up from the test scores to study which processes that the students had or had not developed, and the extent of the development, by analysing the data obtained from their thinking-aloud protocols and answer scripts for the pretest and the posttest qualitatively (Section 8.3).

8.2 PROFICIENCY IN MATHEMATICAL INVESTIGATION

The proficiency of a student's performance in the pretest and posttest was measured using an Investigation Scoring Rubric (ISR). On one hand, students who were well versed in processes might not discover anything significant because they might lack what Schoenfeld (1985) called resources, e.g. mathematical conceptual knowledge and procedural skills. On the other hand, students who were poor in processes might discover some non-trivial patterns as they had more mathematical resources at their disposal. Thus there was a need to evaluate both the processes and outcomes in order to assess the quality of an investigation. Therefore, the ISR was developed to evaluate a student's proficiency in investigation based on the quality of his or her processes and outcomes as explicated in the data analysis in the previous chapter.

8.2.1 Investigation Scoring Rubric (ISR)

Table 8.1 shows the ISR. There were six categories based on the stages of the theoretical investigation models, but with some modifications as follows:

- Problem Posing (P) and Extension (E) were combined as one category for two reasons: (i) the two processes were similar in that both involved posing problems, and (ii) there was no need for the students to extend Type A tasks, so they might get a zero if there was a separate category for Extension (E), which would not be reflective of their actual performance.
- Justifying (J) and Generalising (G) were combined as one category, just as in the first two data analysis instruments (i.e. the Investigation Pathway Diagram and the Investigation Timeline Representation) described in Section 5.2.

Table 8.1 Scoring Rubric for Evaluating Student's Proficiency in Investigation

Category	Level 0	Level 1	Level 2
Understanding the task (U)	• Misinterpreted task, but did not recover or monitor understanding	• Misinterpreted task but recovered late (i.e. after more than 5 min); • *or* misinterpreted task and monitored understanding, but did not recover	• Interpreted task correctly; • *or* misinterpreted task but recovered on time (i.e. within 5 min)
Problem Posing and / or Extension (P/E)	• Did not understand what to investigate and so did not search for any pattern or pose any specific problem to solve or extend	• Posed general problem to search for patterns explicitly or otherwise; • *or* posed at least one trivial specific problem to solve or extend; • *or* posed at least one non-trivial specific problem to solve or extend, without any intention to generalise	• Posed at least one non-trivial specific problem to solve or extend, with the intention to generalise
Specialising / Using Other Heuristics (S/H)	• Chose examples randomly; • *or* did not use other heuristics effectively	• Chose at least one example purposefully; • *or* used other heuristics quite effectively	• Chose at least one set of systematic examples; • *or* used other heuristics effectively
Conjecturing (C)	• Did not observe any pattern or formulate any conjecture; • *or* observed incorrect patterns or formulated incorrect conjectures; • *or* observed correct trivial patterns	• Observed at least one correct non-trivial pattern; • *or* formulated at least one correct trivial conjecture	• Formulated at least one correct non-trivial conjecture
Justifying and Generalising (J/G)	• Did not formulate any conjecture to justify; • *or* did not prove any conjecture due to lack of time; • *or* tried but failed to justify any conjecture; • *or* correctly proved trivial conjectures that will not lead to any generalisation; • *or* refuted incorrect trivial conjectures; • *or* wrongly accepted all conjectures as true without testing or based on naïve testing	• Correctly proved at least one trivial conjecture that will lead to generalisation; • *or* refuted at least one incorrect non-trivial conjecture; • *or* tried but failed to prove at least one correct non-trivial conjecture	• Correctly proved at least one non-trivial conjecture
Checking (R)	• Made major errors but did not discover; • *or* did not make any major error, but never or seldom checked working, reviewed solution and monitored progress	• Made major errors but discovered only some of them, or discovered all of them but some late (i.e. after more than 5 min); • *or* did not make any major error, but occasionally checked working, reviewed solution and monitored progress	• Made major errors but discovered all on time (i.e. within 5 min); • *or* did not make any major error, but often checked working, reviewed solution and monitored progress

Total: ______ / 12

There were 3 levels for each category in the scoring rubric: Levels 0 to 2. The total possible score for the scoring rubric was 12. The descriptors for each level included the student's processes and outcomes for the investigation of both types of tasks. However, some of the processes might not be observed, but the students might have engaged in these processes mentally without thinking them aloud. On one hand, the scoring rubric could only score the processes that were observed. On the other hand, the scoring rubric should not over-penalise students for failing to verbalise their thinking processes, especially when they were able to produce the outcomes.

For example, in the first category of understanding the task (U), a student might interpret the task correctly without showing any observable understanding process, but this does not mean that the student did not engage in these processes mentally. Thus the student should not be over-penalised, so he or she should obtain the highest score of 2 for this category. However, if the student had misinterpreted the task, the criterion would be whether he or she had recovered fast enough. If the student recovered on time (within 5 minutes), he or she would still be at Level 2 as the ability to recover on time was also an important one. But if the student recovered late (after 5 minutes), he or she would be at Level 1. If the student did not recover at all, the criterion would be whether the student had monitored his or her understanding. If the student had shown signs of monitoring understanding but still did not recover, the student would be at Level 1 for his or her effort in trying. But if the student did not show any sign of monitoring understanding and did not recover, he or she would be at Level 0. In this manner, the scoring rubric tried to balance between scoring observable processes, and not over-penalising students for producing outcomes but failing to verbalise their thinking processes.

In the second category of problem posing and extension (P/E), the descriptors distinguished between the two types of investigative tasks. For Type A tasks, a student who posed the general problem to search for any pattern, whether explicitly or otherwise, would be at Level 1. But a student could also pose specific problems to solve or extend at a later stage for Type A tasks. If the student posed at least one non-trivial problem with the intention to generalise, he or she would be at Level 2. But if the student could only pose trivial problems, or non-trivial problems without any intention to generalise, then he or she would be at Level 1. For Type B tasks, a student would need to pose specific problems to solve or extend. Similarly, whether the student was at Level 1 or 2 depended on whether the specific problems posed were trivial or non-trivial, and whether there was any intention to generalise. Level 0 was reserved for students who did not understand what to investigate and so did not search for any pattern or pose any specific problem to solve or extend. A problem was classified as trivial or non-trivial based on the task analysis described in Appendix E, and the classification had passed an inter-coder reliability test (see Section 5.4).

In the third category of specialising and using other heuristics (S/H), the descriptors also distinguished between the two types of investigative tasks. For Type A tasks, the process of specialising was divided into 3 levels: systematic (Level 2), purposeful (Level 1) and random (Level 0). The definitions of the three types of specialising were given in Section 7.2.3. For Type B tasks, the use of other heuristics was task dependent. This means that the scoring rubric could only divide the process broadly into 3 levels: use other heuristics effectively (Level 2), quite effectively (Level 1) and ineffectively (Level 0). The definitions of the three types of effectiveness for Posttest Task 2 (Sausage) were given in Section 7.3.3. Since Pretest Task 2 (Toast) is parallel to Posttest Task 2 in the structure of their solutions, i.e. the three toasting methods are parallel to the three cutting methods as explained in Table 3.5 in Section 3.6.4, then the three types of effectiveness for Pretest Task 2 are similarly defined: if the students discovered only Toasting Method A, they were reckoned to have used other heuristics 'ineffectively' (Level 0); if they struggled but eventually discovered Toasting Method B that gives the least toasting time, they were reckoned to have used other heuristics 'quite effectively' (Level 1); if they were able to discover Toasting Method B naturally without struggling, whether right from the beginning or after using Toasting Method A, they were reckoned to have used other heuristics 'effectively' (Level 2).

In the fourth category of conjecturing (C), a student would be at Level 2 if he or she was able to formulate at least one correct non-trivial conjecture. But if the conjecture was correct but trivial, the student would be at Level 1. Sometimes, a student might discover a correct non-trivial pattern, but he or she did not see it as the underlying pattern and so did not formulate any conjecture. In this case, the student would still be at Level 1. Level 0 was reserved for students who did not observe any pattern or formulate any conjecture, or observed incorrect patterns or formulated incorrect conjectures, or observed correct but trivial patterns.

In the fifth category of justifying and generalising (J/G), a student would be at Level 2 if he or she was able to prove at least one non-trivial conjecture correctly. It did not matter whether the proof would lead to any generalisation because the main criterion was that the conjecture must be non-trivial. But if the proven conjecture was trivial, then generalisation would play a part in determining whether a student was at Level 1 or 0. If a student correctly proved at least one trivial conjecture that will lead to generalisation, or refuted at least one incorrect non-trivial conjecture, or tried but failed to prove at least one correct non-trivial conjecture, then he or she would be at Level 1. Level 0 was reserved for students who did not formulate any conjecture to justify, or did not prove any conjecture due to lack of time, or tried but failed to justify any conjecture, or correctly proved trivial conjectures that will not lead to any generalisation, or refuted incorrect trivial conjectures, or wrongly accepted conjectures as true without testing or based on naïve testing.

In the last category of checking (R), the main criterion would be whether the checking and monitoring processes, such as checking working (CW), reviewing solution (MR) and monitoring progress (MP), had helped a student in discovering all the major errors on time (i.e. within 5 minutes). If this happened, the student would be at Level 2. But if the student only discovered some major errors, or discovered all the major errors but some of them late (i.e. after more than 5 minutes), he or she would be at Level 1. If the student did not discover any major error, he or she would be at Level 0. However, the above criterion applied if and only if a student had made at least one major error. But if the student did not make any major error, the criterion would be the total frequency of checking and monitoring processes: $T = CW + MR + MP$. The student would be at Level 2, 1 or 0 depending on whether he or she had engaged in all these processes often ($T \geq 9$), occasionally ($T = 5$ to 8), or seldom or never ($T \leq 4$) respectively.

8.2.2 Actual Scoring for Sample Students

This section will describe how the scoring rubric was used to score the pretest and the posttest in the present study by going through in detail the scoring for a student for each type of investigative tasks. The student selected for the Kaprekar Task (Type A) was S5 while the student chosen for the Sausage Task (Type B) was S9. These two students were selected because their pathway for the respective task was complete, so it was possible to illustrate how to score for each category of the scoring rubric. Table 8.2 shows the breakdown of the score for each category for the investigation of the Kaprekar Task by S5. The maximum possible score for each category is 2 and the maximum possible total score is 12.

Table 8.2 Scoring of S5 for Kaprekar Task

Category	**U**	**P/E**	**S/H**	**C**	**J/G**	**R**	**Total**
Level	2	2	1	2	2	1	**10**

To score each category, reference was made to the respective tables in the data analysis in Section 7.2 in the previous chapter:

- From Table 7.2 in Section 7.2.1, it was observed that S5 interpreted the task correctly, so she was at Level 2 for the category of understanding the task (U).
- From Table 7.3 in Section 7.2.2, it was observed that S5 posed two non-trivial specific problems with the intention to generalise, so she was at Level 2 for the category of problem posing and extension (P/E).
- From Table 7.4 in Section 7.2.3, it was observed that S5 chose five examples purposefully but she did not specialise systematically, so she was at Level 1 for the category of specialising and using other heuristics (S/H).
- From Table 7.8 in Section 7.2.4, it was observed that S5 formulated one correct non-trivial conjecture, so she was at Level 2 for the category of conjecturing (C).

- From Table 7.10 in Section 7.2.5, it was observed that S5 proved her non-trivial conjecture correctly, so she was at Level 2 for the category of justifying and generalising (J/G).
- From Table 7.11 in Section 7.2.6, it was observed that S5 only discovered one of her three major errors at about 8 minutes after making the error, so she was at Level 1 for the category of checking (R).

Therefore, based on the data analysis of her processes and outcomes in Section 7.2, S5 obtained a total score of 10 for her investigation of the Kaprekar Task. Table 8.3 shows the breakdown of the score for each category for the investigation of the Sausage Task by S9. The maximum possible score for each category is 2 and the maximum possible total score is 12.

Table 8.3 Scoring of S9 for Sausage Task

Category	**U**	**P/E**	**S/H**	**C**	**J/G**	**R**	**Total**
Level	2	2	2	2	1	2	**11**

To score each category, reference was made to the respective tables in the data analysis in Section 7.3:

- From Table 7.13 in Section 7.3.1, it was observed that S9 interpreted the task correctly, so he was at Level 2 for the category of understanding the task (U).
- From Table 7.14 in Section 7.3.2, it was observed that S9 posed two non-trivial specific problems for the original task, without any intention to generalise. It was further observed from Table 7.19 in Section 7.3.6 that S9 posed one non-trivial specific problem to extend the task, with the intention to

generalise. Thus he was at Level 2 for the category of problem posing and extension (P/E) for posing at least one non-trivial specific problem with the intention to generalise.

- From Table 7.16 in Section 7.3.3, it was observed that S9 discovered Cutting Method B that gives the least number of cuts. From his protocols discussed in the same section, it was found that he was able to reason effectively to discover Method B naturally without struggling. It was further observed from Table 7.22 in Section 7.3.7 that he used Method B during the extension of the task. Thus he was at Level 2 for using other heuristics (H) effectively.
- From Table 7.17 in Section 7.3.4, it was observed that S9 formulated a correct non-trivial conjecture for the original task. It was further observed from Table 7.23 in Section 7.3.8 that he also formulated a correct non-trivial conjecture during the extension of the task. Thus he was at Level 2 for the category of conjecturing (C) for formulating at least one correct non-trivial conjecture.
- From Table 7.18 in Section 7.3.5, it was observed that S9 wrongly accepted his conjecture for the original task as true without testing. It was further observed from Table 7.24 in Section 7.3.9 that he tried but failed to prove his correct non-trivial conjecture for the extension. Thus he was at Level 1 for the category of justifying and generalising (J/G) for trying to prove at least one correct non-trivial conjecture but failed.
- From Table 7.25 in Section 7.3.10, it was observed that S9 made only one major error but he discovered it on time (i.e. within 5 minutes), so he was at Level 2 for the category of Checking (R).

Therefore, based on the data analysis of his processes and outcomes in Section 7.3, S9 obtained a total score of 11 for his investigation of the Sausage Task. As can be seen from the above scoring system, most of the descriptors in the scoring rubric are objective, e.g. whether a student misinterpreted the task or whether he recovered within 5 minutes. The only descriptors that are subject to interpretation are whether the problem posed or conjectures formulated by the students are trivial or non-trivial. However, the classification of these outcomes as trivial or non-trivial had passed an inter-coder reliability test (see Section 5.4). This suggests that the scoring rubric is a reliable instrument to measure a student's proficiency in mathematical investigation.

8.2.3 Analysis of the Pretest and Posttest Scores

In this section, the pretest and posttest scores of the 10 students for the present study will be compared using descriptive statistics. Since the sample for the current research was a purposeful one (see Section 3.6.2) and not a random one, no inference will be made using statistical testing. Table 8.4 shows the detailed scores for the two pretest tasks and the two posttest tasks for the 10 students based on the scoring rubric described earlier in this chapter. The maximum score for each category is 2 and the maximum total score is 12. Table 8.5 shows the means and standard deviations of the test scores across tasks while Table 8.6 shows a summary of the test scores across students.

Table 8.4 Detailed Pretest and Posttest Scores

	Pretest Task 1 (Type A)							Posttest Task 1 (Type A)						
	U	P/E	S/H	C	J/G	R	Total	U	P/E	S/H	C	J/G	R	Total
S1	0	2	0	0	0	0	**2**	2	2	0	0	0	1	**5**
S2	2	1	0	1	0	1	**5**	2	1	1	0	1	0	**5**
S3	0	1	1	0	0	0	**2**	2	1	1	1	0	1	**6**
S4	2	1	1	1	0	0	**5**	2	1	1	0	0	0	**4**
S5	2	1	1	2	0	0	**6**	2	2	1	2	2	1	**10**
S6	0	1	1	0	0	0	**2**	2	1	1	0	0	0	**4**
S7	0	1	0	0	0	2	**3**	2	1	1	0	0	2	**6**
S8	0	1	0	1	0	0	**2**	2	1	0	1	0	0	**4**
S9	0	2	0	1	0	1	**4**	1	1	2	2	2	2	**10**
S10	1	1	1	1	0	0	**4**	2	1	0	2	0	0	**5**
Ave	**0.7**	**1.2**	**0.5**	**0.7**	**0**	**0.4**	**3.5**	**1.9**	**1.2**	**0.8**	**0.8**	**0.5**	**0.7**	**5.9**
S.d.	0.9	0.4	0.5	0.6	0	0.7	1.4	0.3	0.4	0.6	0.9	0.8	0.8	2.2

	Pretest Task 2 (Type B)							Posttest Task 2 (Type B)						
	U	P/E	S/H	C	J/G	R	Total	U	P/E	S/H	C	J/G	R	Total
S1	0	1	0	0	0	0	**1**	2	2	2	2	2	0	**10**
S2	2	1	0	0	0	0	**3**	2	1	0	1	0	1	**5**
S3	0	1	0	0	0	0	**1**	2	2	0	2	0	0	**6**
S4	0	2	2	2	1	1	**8**	2	1	2	1	0	0	**6**
S5	2	1	1	0	0	0	**4**	1	1	0	0	0	2	**4**
S6	2	1	0	0	0	0	**3**	0	1	0	0	0	0	**1**
S7	0	2	0	0	0	0	**2**	2	1	0	0	0	2	**5**
S8	2	1	1	0	0	0	**4**	2	2	1	2	1	1	**9**
S9	2	2	0	1	0	2	**7**	2	2	2	2	1	2	**11**
S10	1	1	1	0	0	0	**3**	2	2	1	2	2	0	**9**
Ave	**1.1**	**1.3**	**0.5**	**0.3**	**0.1**	**0.3**	**3.6**	**1.7**	**1.5**	**0.8**	**1.2**	**0.6**	**0.8**	**6.6**
S.d.	0.9	0.5	0.7	0.6	0.3	0.6	2.2	0.6	0.5	0.9	0.9	0.8	0.9	2.9

Table 8.5 Means and Standard Deviations of Test Scores Across Tasks

Mean	Type A Tasks							Type B Tasks						
	U	P/E	S/H	C	J/G	R	Total	U	P/E	S/H	C	J/G	R	Total
Pre	0.7	1.2	0.5	0.7	0	0.4	**3.5**	1.1	1.3	0.5	0.3	0.1	0.3	**3.6**
Post	1.9	1.2	0.8	0.8	0.5	0.7	**5.9**	1.7	1.5	0.8	1.2	0.6	0.8	**6.6**
Diff	**1.2**	**0**	**0.3**	**0.1**	**0.5**	**0.3**	**2.4**	**0.6**	**0.2**	**0.3**	**0.9**	**0.5**	**0.5**	**3.0**

S.d.	Type A Tasks							Type B Tasks						
	U	P/E	S/H	C	J/G	R	Total	U	P/E	S/H	C	J/G	R	Total
Pre	0.9	0.4	0.5	0.6	0	0.7	**1.4**	0.9	0.5	0.7	0.6	0.3	0.6	**2.2**
Post	0.3	0.4	0.6	0.9	0.8	0.8	**2.2**	0.6	0.5	0.9	0.9	0.8	0.9	**2.9**
Diff	**-0.6**	**0**	**0.1**	**0.3**	**0.8**	**0.1**	**0.8**	**-0.3**	**0**	**0.2**	**0.3**	**0.5**	**0.3**	**0.7**

Table 8.6 Summary of Test Scores Across Students

	Pretest 1	Posttest 1	Difference	Pretest 2	Posttest 2	Difference
S1	2	5	3	1	10	9
S2	5	5	0	3	5	2
S3	2	6	4	1	6	5
S4	5	4	-1	8	6	-2
S5	6	10	4	4	4	0
S6	2	4	2	3	1	-2
S7	3	6	3	2	5	3
S8	2	4	2	4	9	5
S9	4	10	6	7	11	4
S10	4	5	1	3	9	6
Ave	**3.5**	**5.9**	**2.4**	**3.6**	**6.6**	**3.0**
S.d.	1.4	2.2	2.0	2.2	2.9	3.4

(a) Comparison between Pretest Task 1 and Posttest Task 1 (Type A)

Table 8.4 shows that the students obtained an average total score of 3.5 and 5.9 (out of 12) for Pretest Task 1 and Posttest Task 1 respectively. If a student was at Level 1 for all the 6 categories, the total score would be 6. This means that, on average, the students were below Level 1 for the pretest task, but they had improved by 2.4 points to Level 1 for the posttest task. Table 8.5 shows that the students had improved in most categories for Posttest Task 1, except for problem posing[24] (P) and conjecturing (C) where there was not much difference. In particular, the students had improved the most for understanding the task (U) with a rise of 1.2 points, followed by justifying and generalising (J/G) with an increase of a moderate 0.5 point. The students had also improved slightly for specialising[25] (S) and checking (R) with an increase of 0.3 point each. The extreme scores were 0 for J/G for Pretest Task 1 and almost the maximum 2 for U for Posttest Task 1.

[24] The students did not extend (E) the two Type A tasks.
[25] The students did not use other heuristics (H) for the two Type A tasks.

In terms of spread, Table 8.5 shows that the standard deviation (s.d.) for Pretest Task 1 was highest for U at 0.9, and lowest for J/G at 0, but the s.d. for Posttest Task 1 was highest for C at 0.9, and second highest for J/G and R at 0.8, while it was lowest for U at 0.3. For the pretest task, the s.d. was the biggest for U at 0.9 since some students understood the task correctly, some misinterpreted the task but recovered, while the rest did not recover. But for the posttest task, most of them understood the task correctly, with the score very close to the maximum 2 at 1.9, and so the s.d. was the smallest for U at 0.3. On the other hand, the students all scored 0 for J/G for the pretest task, so the s.d. was the smallest at 0. But for the posttest task, some students had improved a lot, some had improved a little, while the rest had not improved at all, and so the s.d. for J/G was the second biggest at 0.8. Although the largest s.d. for the posttest task was for C at 0.9, the s.d. for C for the pretest task was already quite big at 0.6. This suggests that some students had improved for C while the others had not improved, thus making the s.d. even bigger. Overall, the s.d. for the Type A tasks had also become bigger from 1.4 to 2.2, which means that there was more variation in the students' posttest performance compared with the pretest.

Table 8.6 shows that the highest score for Pretest Task 1 was 6 for S5, followed by 5 for S2 and S4. But for Posttest Task 1, the highest score was 10 for S5 and S9, followed by 6 for S3. The top student (S5) for the pretest task ended up joint top for the posttest task, with an improvement of 4 points. But the 2 students (S2,S4) who scored the second highest for the pretest task performed only equally well or even slightly worse for the posttest task. One of the students (S4) was sick when he took the posttest, which might have affected his performance. However, one student (S9), who accomplished only a below average performance for the pretest task with a score of 4, improved the most by 6 points to be joint top for the posttest task. At the other end, 3 students (S3,S6,S7) obtained the lowest score of 2 for Pretest Task 1 while 2 students (S6,S8) obtained the lowest score of 4 for Posttest Task 1. This means that S6 was consistently the lowest scorer for both tasks, while S3 and S7 had improved by 4 points and 3 points respectively. S8 improved only by 2 points, thus ending up joint lowest for the posttest task.

(b) Comparison between Pretest Task 2 and Posttest Task 2 (Type B)

Table 8.4 shows that the students obtained an average total score of 3.6 and 6.6 (out of 12) for Pretest Task 2 and Posttest Task 2 respectively. This means that, on average, the students were below Level 1 for the pretest task, but they had improved by 3 points to Level 1 for the posttest task, just like for the two Type A tasks. Table 8.5 shows that the students had improved in all the categories for Posttest Task 2. In particular, the students had improved the most for conjecturing (C) with an increase of 0.9 point, followed by a moderate improvement of 0.5 to 0.6 point for understanding the task (U), justifying and generalising (J/G), and checking (R). The students had also improved slightly for problem posing and extension (P/E), and specialising and using other heuristics (S/H), with an increase of 0.2 to 0.3 point each. The extreme scores were 0.1 for J/G for Pretest Task 2 and 1.7 for U for Posttest Task 2, just like for the two Type A tasks.

In terms of spread, Table 8.5 shows that the s.d. for Pretest Task 2 was highest for U at 0.9, and lowest for J/G at 0.3, but the s.d. for Posttest Task 2 was highest for S/H, C and R at 0.9, and second highest for J/G at 0.8, while it was lowest for P/E at 0.5. Just like for Pretest Task 1, the s.d. for Pretest Task 2 was the biggest for U at 0.9. But for

Posttest Task 2, although most students understood the task correctly, there was still some variation and so the s.d. for U only dropped slightly to 0.6, which was unlike the biggest drop for Posttest Task 1. On the other hand, the biggest rise in the s.d. for Posttest Task 2 was an increase of 0.5 point for J/G from 0.3 to 0.8, which was similar to the biggest rise in the s.d. for Posttest Task 1. Overall, the s.d. for the Type B tasks had also increased from 2.2 to 2.9, which means that there was more variation in the students' posttest performance compared with the pretest, just like for the Type A tasks.

Table 8.6 shows that the highest score for Pretest Task 2 was 8 for S4, followed by 7 for S9. But for Posttest Task 2, the highest score was 11 for S9, followed by 10 for S1, and 9 for S8 and S10. The top student (S4) for the pretest task ended up performing worse for the posttest task, probably because he was sick when he took the posttest. However, the student (S9) with the second highest pretest score ended up the top student for the posttest task with a score of 11: he was the only one who found the complicated formula for the least number of cuts for Posttest Task 2 (see Section 6.3a). At the other end, 2 students (S1,S3) obtained the lowest score of 1 for Pretest Task 2 while one student (S6) obtained the lowest score of 1 for Posttest Task 2. The student with the biggest improvement was S1, who improved by 9 points from the joint lowest pretest score of 1 to the second highest posttest score of 10. This was because he did not understand the requirements for Pretest Task 2 and so did not know what to investigate. But he had learnt what to investigate for Posttest Task 2 and he managed to solve the problems that he had posed, thus leading to such a great improvement.

(c) Comparison Across the Two Types of Tasks

In general, Table 8.5 shows that the students performed equally badly for the two pretest tasks, with an average score of 3.5 and 3.6 (out of 12) respectively. But they had improved by quite a bit for the two posttest tasks: an increase of 2.4 and 3 points respectively. Nevertheless, they were still at about Level 1 on average for both the posttest tasks, with an average score of 5.9 and 6.6 respectively. In other words, the students performed better and improved more for Posttest Task 2 (Type B) than for Posttest Task 1 (Type A). On closer analysis of the two pretest tasks, the students performed the best in problem posing and extension (P/E) with a score of 1.2 and 1.3 respectively, and the worst in justifying and generalising (J/G) with a score of 0 and 0.1 respectively. But for the two posttest tasks, the students performed the best in understanding the task (U) with a score of 1.9 and 1.7 respectively, but the worst was still in J/G with a score of 0.5 and 0.6 respectively. Thus the main process that the students had improved a lot for both types of tasks was U: an increase of 1.2 and 0.6 point respectively. But for the Type B tasks, the improvement for U was only the second highest, while the greatest increase of 0.9 point was for conjecturing (C). On the other hand, the main process that showed the least improvement for both types of tasks was P/E: no difference for the Type A tasks and an increase of only 0.2 point for the Type B tasks.

In terms of spread of the total scores, Table 8.5 shows that the s.d. for the two Type B tasks were bigger than those for the two Type A tasks, and the s.d. for the posttest tasks were also bigger than those for the pretest tasks. In fact, the s.d. for Pretest Task 2 (Type B) was already equal to the s.d. for Posttest Task 1 (Type A). This suggests that there was more variation in the students' performance for Type B tasks compared with Type A tasks. On closer analysis, the s.d. for both pretest tasks were highest for U at 0.9, and lowest for J/G at 0 and 0.3 respectively. For both posttest tasks, the s.d. were among the highest for C, J/G and R from 0.8 to 0.9, and among the lowest for U and P/E from 0.3 to 0.6. There were two other similarities between the two types of tasks: the s.d. remained the same for P/E from the pretest to the posttest, and the increase in the s.d. from the pretest to the posttest was highest for J/G. Table 8.6 shows that 3 students (e.g. S3,S4,S7) had improved or performed worse by about the same amount from the pretest to the posttest across both types of tasks, while the improvement of the other students was task dependent, e.g. 3 students (S5,S6,S9) had improved more for Type A tasks, but the remaining 4 students (S1,S2,S8,S10) had shown greater improvement for Type B tasks. This suggests that the development of processes might depend on the types of tasks.

The above analysis examined the students' performance in mathematical investigation for the pretest and the posttest quantitatively. The next section will examine the students' performance for the tests qualitatively, and in particular, the development of the investigation processes.

8.3 DEVELOPMENT OF INVESTIGATION PROCESSES

In this section, the development of investigation processes for the 10 students in the present study will be examined by comparing the processes and outcomes for the pretest and the posttest across the two types of tasks (the reader should refer to Appendix D to be familiar with the four tasks). Unlike the previous chapter, the analysis in this section will not follow the order of the investigation stages, but it will follow the order of the categories in the scoring rubric in Section 8.2.1. For example, the second category combined the investigation stages of problem posing and extension because both stages were related to the main process of posing problems, the only difference being whether the given in the task statement was changed. Just like in the previous chapter where the students' protocols and test answer scripts for Posttest Task 1 (Kaprekar) and Posttest Task 2 (Sausage) were analysed and the findings displayed in the Summary Tables of Processes and Outcomes (TPO), the students' protocols and test answer scripts for Pretest Task 1 (Happy) and Pretest Task 2 (Toast) were similarly analysed and the findings displayed in corresponding TPO in Appendix M. But for ease of comparison between the pretest and the posttest in this section, the relevant information from all the TPO of the four tasks had to be extracted and presented in another type of summary tables as shown below.

8.3.1 Understanding the Task (Stage 1)

Table 8.7 shows a summary of the proficiency level of understanding the task (U) attained by the 10 students for the four tasks while Table 8.8 shows a summary of the frequencies of the understanding processes that might have helped the students to understand the task correctly. The information for the two pretest tasks was extracted from Tables M1.1 and M2.1 in Appendix M while the information for the two posttest tasks was obtained from Tables 7.2 and 7.13 in the previous chapter. Since some of the levels have more than one descriptor, Table 8.7 separates them in order to present a clearer picture as to how many students were at a particular level because of a certain descriptor. For example, for Pretest Task 1, the table shows that 3 students were at Level 2, but 2 of them interpreted the task correctly (Descriptor A) while the last student recovered from the misinterpretation within 5 minutes of the investigation (Descriptor B).

Table 8.7 Proficiency Level of Understanding the Task

Descriptors	Pretest 1 (Type A)	Posttest 1 (Type A)	Pretest 2 (Type B)	Posttest 2 (Type B)
Level 2 Descriptor A: Interpreted task correctly	2 (S4,S5)	7 (S1-S5, S7,S10)	3 (S2,S6,S8)	7 (S1,S2, S4,S7-S10)
Level 2 Descriptor B: Misinterpreted task but recovered on time (i.e. within 5 min)	1 (S2)	2 (S6,S8)	2 (S5,S9)	1 (S3)
Level 2: Level 2 Descriptors A and B	**3 (S2,S4,S5)**	**9 (S1-S8,S10)**	**5 (S2,S5, S6,S8,S9)**	**8 (S1-S4, S7-S10)**
Level 1 Descriptor A: Misinterpreted task but recovered late (i.e. after > 5 min)	1 (S10)	0	1 (S10)	1 (S5)
Level 1 Descriptor B: Misinterpreted task, and monitored understanding, but did not recover	0	1 (S9)	0	0
Level 1: Level 1 Descriptors A and B	**1 (S10)**	**1 (S9)**	**1 (S10)**	**1 (S5)**
Level 0: Misinterpreted task, but did not recover or monitor understanding	**6 (S1,S3, S6-S9)**	**0**	**4 (S1,S3, S4,S7)**	**1 (S6)**
Average Score for Level (out of 2)	**0.7**	**1.9**	**1.1**	**1.7**
Standard Deviation (s.d.)	0.9	0.3	0.9	0.6

Table 8.8 Frequencies of Processes for Understanding the Task

Processes	Pretest 1 (Type A)	Posttest 1 (Type A)	Pretest 2 (Type B)	Posttest 2 (Type B)
Re-read Task (RR)	36 (S1,S3-S8,S10)	47 (S1,S3-S10)	16 (S1-S3,S5-S7,S10)	11 (S3,S5-S7,S10)
Rephrased Task (RT)	7 (S3-S6)	18 (S4-S9)	6 (S1,S3,S6,S9)	13 (S3-S5,S7, S9,S10)
Highlighted Key Information (HI)	1 (S3)	16 (S3-S6,S9)	4 (S1,S10)	7 (S3-S6,S9)
Visualised Information (VI)	0	0	6 (S2,S5-S7,S9)	11 (S2,S3,S5-S7,S10)
Monitored Understanding (MU)	1 (S5)	9 (S1,S3,S4, S7,S9,S10)	11 (S2,S5,S6, S8-S10)	6 (S3,S7,S9,S10)
Total Frequency	**45 (S1,S3-S8,S10)**	**90 (S1,S3-S10)**	**43 (S1-S3,S5-S10)**	**48 (S2-S7,S9,S10)**

(a) Comparison between Pretest Task 1 and Posttest Task 1 (Type A)

Table 8.7 shows that only 2 students understood Pretest Task 1 correctly at the start while 7 students interpreted Posttest Task 1 correctly (Level 2). Although 2 other students recovered from misinterpreting Pretest Task 1, with one of them recovering on time (Level 2) but the other late (Level 1), 6 students still misinterpreted the task in the end (Level 0). But for Posttest Task 1, only one student did not recover from misinterpreting the task, but he did monitor his understanding (Level 1), so no one was at Level 0. Thus the students had performed better in understanding the task for Posttest Task 1 than for Pretest Task 1. Table 8.8 shows that the students had engaged in understanding processes, such as RR, RT, HI and MU, twice as often for Posttest Task 1 compared with Pretest Task 1. The timings on their protocols also show that they had spent more time understanding the posttest task than the pretest task. This suggests that engaging in these processes more often and for a longer period of time might have helped more students to correctly interpret Posttest Task 1 than Pretest Task 1. However, it was possible for some students to correctly understand the task without appearing to engage in these processes, e.g. S2 did not exhibit any understanding behaviour for Posttest Task 1 but he interpreted the task correctly.

(b) Comparison between Pretest Task 2 and Posttest Task 2 (Type B)

Table 8.7 shows that 3 students and 7 students correctly understood Pretest Task 2 and Posttest Task 2 respectively (Level 2). Although 3 other students recovered from misinterpreting Pretest Task 2, with 2 of them recovering on time (Level 2) but the other late (Level 1), 4 students still misinterpreted the task in the end (Level 0). But for Posttest Task 2, only one student did not recover from misinterpreting the task (Level 0). Thus the students had performed better in understanding the task for Posttest Task 2 than for Pretest Task 2, just like for the two Type A tasks.

Table 8.8 shows that the students had engaged in understanding processes, such as RR, RT, HI, VI and MU, slightly less often for Pretest Task 2 than for Posttest Task 2.

On closer analysis, it was observed that the students highlighted key information and visualised information less often for the pretest task than for the posttest task. They also re-read or rephrased Pretest Task 2 for 16 + 6 = 22 times, which was slightly lower than 11 + 13 = 24 times for Posttest Task 2. But the task statement for Pretest Task 2 contained a lot more information, such as the various timings for toasting one side of each slice of bread, putting in and taking out a slice, and turning a slice over. In other words, despite the fact that the task statement for Posttest Task 2 was much shorter, the students still re-read or rephrased the posttest task slightly more often than the pretest task. The students also monitored their understanding for Pretest Task 2 more often than for Posttest Task 2, probably because there were more given conditions in the task statement of Pretest Task 2 to clarify. But this metacognitive process was still not so helpful in preventing them from misinterpreting the pretest task. Overall, more students might have interpreted Posttest Task 2 correctly because they had engaged in these understanding processes a bit more often than for Pretest Task 2.

(c) Development of Understanding Processes

The above analysis shows that the students had engaged in the understanding processes more often for the two posttest tasks compared with the two pretest tasks. Before the pretest, the students only went through a two-hour familiarisation lesson to learn what to investigate when given an open investigative task, and to practise thinking aloud for the pretest (see outline of Lesson 1 in Appendix C), so the focus was not on the understanding processes. But the students had learnt during their normal school lessons to read textbook exercise questions carefully and to highlight key information before attempting to solve them. These strategies were very similar to the processes for understanding an investigative task. However, they did not engage in these processes often enough for the two pretest tasks, thus leading to most of them misinterpreting the pretest tasks.

During the first two-hour developing lesson of the teaching experiment after the pretest, part of the focus was on the processes for understanding an investigative task (see outline of Lesson 2 in Appendix C). The students were reminded of the usual strategies for understanding textbook exercise questions: they were guided me, who was the teacher, to articulate that they should read the task carefully, re-read or rephrase important parts of the task, and highlight key information. In addition, they were taught that they should try at least one example to understand a Type A task, and to visualise information by drawing a diagram if applicable. They were also taught that they should monitor their understanding to ensure that they had interpreted the task correctly before proceeding to the next stage of problem posing.

As a result, the students engaged in these processes more often for the two posttest tasks. In particular, they exhibited twice as many behaviours for understanding Posttest Task 1 as Pretest Task 1. Despite the fact that the task statement for Posttest Task 2 was much shorter and contained less information than that of Pretest Task 2, they still re-read or rephrased Posttest Task 2 slightly more often than Pretest Task 2. Thus more students interpreted the two posttest tasks correctly compared with the two pretest tasks. In other words, the students needed constant reminders during the teaching experiment to apply what they had learnt about the processes for understanding textbook exercise questions to understand investigative tasks.

8.3.2 Problem Posing and Extension (Stages 2 and 8)

Table 8.9 shows the proficiency level of problem posing and extension (P/E) attained by the 10 students for the four tasks, Table 8.10 shows the types of problems and extensions that the students had posed for the four tasks, Table 8.11 shows the number of students who had posed the intended problem or the intended extension for the two Type B tasks, while Table 8.12 shows the frequencies of the problem-posing and extension processes that might have helped the students to pose specific problems for the four tasks. The information for the two pretest tasks was extracted from Tables M1.2 and M2.2a to M2.2d in Appendix M while the information for the two posttest tasks was obtained from Tables 7.3, 7.14, 7.15, 7.18 and 7.19 in the previous chapter. It was possible for a student to fulfil more than one descriptor in the level that he or she was in, e.g. 7 students were at Level 1 for Pretest Task 2 as shown in Table 8.9, but 5 of them (S1-S3,S6,S8) satisfied more than one descriptor in that level. However, if a student was at a higher level but also satisfied some descriptors at a lower level, he or she would *not* be included at the lower level to avoid confusion, e.g. one other student (S7) also fulfilled Level 1 Descriptor C for Pretest Task 2, but she was not included at that level since she was already at Level 2.

Table 8.9 Proficiency Level of Problem Posing and Extension

Descriptors	Pretest 1 (Type A)	Posttest 1 (Type A)	Pretest 2 (Type B)	Posttest 2 (Type B)
Level 2 Descriptor A: Posed at least one non-trivial specific problem to solve, with the intention to generalise	2 (S1,S9)	2 (S1,S5)	0	0
Level 2 Descriptor B: Posed at least one non-trivial specific problem to extend, with the intention to generalise	0	0	3 (S4,S7,S9)	5 (S1,S3, S8-S10)
Level 2: Level 2 Descriptors A and B	**2 (S1,S9)**	**2 (S1,S5)**	**3 (S4,S7,S9)**	**5 (S1,S3, S8-S10)**
Level 1 Descriptor A: Posed general problem to search for any pattern explicitly	0	3 (S3,S7,S9)	0	0
Level 1 Descriptor B: Search for patterns without posing general problem explicitly	8 (S2-S8,S10)	5 (S2,S4, S6,S8,S10)	0	0
Level 1 Descriptor C: Posed at least one trivial specific problem to solve	1 (S8)	1 (S9)	5 (S1-S3, S6,S8)	5 (S2,S4, S5-S7)
Level 1 Descriptor D: Posed at least one trivial specific problem to extend	0	0	1 (S3)	5 (S2,S4, S5-S7)
Level 1 Descriptor E: Posed at least one non-trivial specific problem to solve, without any intention to generalise	0	0	7 (S1-S3,S5, S6,S8,S10)	3 (S2,S5,S6)
Level 1 Descriptor F: Posed at least one non-trivial specific problem to extend, without any intention to generalise	0	0	0	0
Level 1: Level 1 Descriptors A-F	**8 (S2-S8,S10)**	**8 (S2-S4, S6-S10)**	**7 (S1-S3,S5, S6,S8,S10)**	**5 (S2,S4, S5-S7)**
Level 0: Did not search for any pattern or pose any specific problem to solve or extend	**0**	**0**	**0**	**0**
Average Score for Level (out of 2)	**1.2**	**1.2**	**1.3**	**1.5**
Standard Deviation (s.d.)	0.4	0.4	0.5	0.5

Table 8.10 Types of Problems and Extensions

Problems and Extensions	Pretest 1 (Type A)	Posttest 1 (Type A)	Pretest 2 (Type B)	Posttest 2 (Type B)
No. of Trivial Problems	2 (S8)	4 (S5,S9)	8 (S1-S3,S6-S8)	12 (S1-S7,S9,S10)
No. of Non-Trivial Problems	2 (S1,S9)	3 (S1,S5)	9 (S1-S6,S8-S10)	10 (S1,S2,S5, S6,S8-S10)
Total No. of Problems	**4 (S1,S8,S9)**	**7 (S1,S5,S9)**	**17 (S1-S10)**	**22 (S1-S10)**
No. of Trivial Extensions	0	0	5 (S3,S7,S9)	14 (S1-S7,S10)
No. of Non-Trivial Extensions	0	0	3 (S4,S7,S9)	5 (S1,S3,S8-S10)
Total No. of Extensions	**0**	**0**	**8 (S3,S4,S7,S9)**	**19 (S1-S10)**
Total No. of Problems and Extensions	**4 (S1,S8,S9)**	**7 (S1,S5,S9)**	**25 (S1-S10)**	**41 (S1-S10)**

Table 8.11 Intended Problem and Intended Extension for Type B Tasks

Intended Problem or Extension	Pretest 2 (Type B)	Posttest 2 (Type B)
No. of students who posed intended problem	8 (S1-S6,S9,S10)	5 (S1,S2,S8-S10)
No. of students who posed intended extension	2 (S4,S9)	2 (S1,S9)

Table 8.12 Frequencies of Processes for Problem Posing and Extension

Processes	Pretest 1 (Type A)	Posttest 1 (Type A)	Pretest 2 (Type B)	Posttest 2 (Type B)
Referred to Task Statement	1 (S1)	1 (S5)	15 (S1-S10)	22 (S1-S10)
Referred to Current Working	2 (S8,S9)	1 (S1)	1 (S1)	4 (S1,S2,S9,S10)
Referred to Previous Result	0	4 (S5,S9)	9 (S1,S3,S4, S7-S9)	13 (S1,S2-S5, S8-S10)
Referred to Given Checklist	0	3 (S5)	0	10 (S2,S4,S6,S10)
Analysed Feasibility of Goal (MG)	0	0	0	1 (S10)
Total Frequency	**3 (S1,S8,S9)**	**9 (S1,S5,S9)**	**25 (S1-S10)**	**50 (S1-S10)**

(a) Comparison between Pretest Task 1 and Posttest Task 1 (Type A)

Table 8.9 shows that only 2 students were at Level 2 for both Pretest Task 1 and Posttest Task 1, and the remaining 8 students were at Level 1 for both tasks. No student was at Level 0. This means that the students had performed equally well in problem posing for both tasks. Although the students could search for any pattern (general problem) without posing specific problems, Table 8.10 shows that they had posed almost twice the number of specific problems for Posttest Task 1 compared with Pretest Task 1. Table 8.12 shows that the frequencies of referring to the task statement and the current working to think of specific problems to pose were about the same for both Pretest Task 1 and Posttest Task 1. But the students had also referred to the given checklist (see Appendix H) and the previous result to think of problems for Posttest Task 1, which did not happen in Pretest Task 1. The former was because the students were not given any checklist during the pretest since they would not have understood most of the investigation processes on the checklist. The latter suggests that a few students had learnt how to use previous results in Posttest Task 1 as a springboard to pose more specific problems to solve. None of the students analysed the feasibility of their goal (MG) for either Type A task.

(b) Comparison between Pretest Task 2 and Posttest Task 2 (Type B)

Table 8.9 shows that only 3 students posed at least one non-trivial specific problem to extend with the intention to generalise (Level 2) for Pretest Task 2, but 5 students were at that level for Posttest Task 2. This indicates that slightly more students were able to pose non-trivial extensions with the intention to generalise for the posttest task compared with the pretest task. All the remaining students were at Level 1 for both tasks. No student was at Level 0. Thus they had performed slightly better in problem posing for Posttest Task 2 than for Pretest Task 2. Table 8.10 shows that the students had posed 41 problems and extensions for Posttest Task 2, which were about 1.6 times the 25 problems and extensions that they had posed for Pretest Task 2. A closer analysis shows that they had posed 19 extensions for the posttest task compared with only 8 extensions for the pretest task. This suggests that they had learnt how to extend the task during the posttest.

It was observed from Table 8.12 that the students referred to the task statement, current working and previous results to think of problems to pose for Posttest Task 2 more often than for Pretest Task 2. In addition, they also referred to the checklist during the posttest, but they were not given the checklist for the pretest. Only one student analysed the feasibility of her goal (MG) for Posttest Task 2, but it was not effective (see Section 7.3.6b). No student exhibited this metacognitive behaviour for Pretest Task 2. In total, they engaged in these problem-posing processes twice as often for Posttest Task 2 compared with Pretest Task 2. This suggests that the frequent references to such processes might have helped them to pose more problems and extensions for the posttest task than for the pretest task. Comparing the frequencies of problem-posing processes across the two types of tasks in Table 8.12, it was observed that the students engaged in these processes far more often for the two Type B tasks than for the two Type A tasks. This was probably because they just needed to search for any pattern (general problem) for the Type A tasks and so they did not need to look at the task statement, current working or previous results to think of a specific problem to pose, unlike the Type B tasks where they had to refer to the task statement to think of the first specific problem to pose, and subsequently they also had to refer to their current working or previous results to think of other problems to pose or to extend.

Another issue to examine was whether the students were able to pose the intended problem or extension. Table 8.11 shows that 8 students had posed the intended problem of finding the least toasting time for Pretest Task 2, but only 2 students were able to pose the intended extension of generalising the least toasting time for toasting n slices in a grill that contains exactly 2 slices (no student posed the other intended extension where the grill contains exactly m slices). For Posttest Task 2, the same table shows that 5 students had posed the intended problem of finding the least number of cuts while only 2 students were able to pose the intended extension of generalising the least number of cuts for sharing n sausages equally among m people.

A closer analysis suggests that the task statement had played an important role in determining why more students had posed the intended problem for Pretest Task 2 compared with Posttest Task 2. For the pretest task, various timings were given for toasting one side of a slice of bread, putting a slice in, taking a slice out, and turning a slice over. Thus it was natural for the students to find the total toasting time, which might lead to finding the least toasting time later. In fact, 5 students (S2,S4,S5,S9, S10) went straight to find the least toasting time without posing the problem to find the total toasting time. But out of the 8 students who posed the intended problem, only 2 of them went on to generalise the least toasting time. However, for the posttest task, nothing was mentioned about the number of cuts in the task statement. It only stated: 'I need to cut 12 identical sausages …' Thus most of the students found it natural to just find how to cut the sausages, with only 5 students finding the number of cuts. In fact, all these 5 students also posed the intended problem of finding the least number of cuts, out of whom, only 2 of them went on to generalise the least number of cuts. Therefore, the ability to pose the intended problem or extension depends on the particular task and how the task statement is phrased.

(c) Development of Problem-Posing Processes

The above analysis shows that the students posed very few specific problems for the two Type A tasks compared with the two Type B tasks. This was expected because they had understood after the familiarisation lesson (see Lesson 1 in Appendix C), which was before the pretest, that the goal of a Type A task was to search for any pattern (general problem), so they just set out to search for patterns for both Type A tasks. In the developing lesson on problem posing (see Lesson 3), there was only enough time to focus on teaching them how to pose specific problems for Type B tasks. Although they were also taught briefly how to pose specific problems for Type A tasks such as "Is the sum of two happy numbers happy or sad?" for Pretest Task 1, the data for Posttest Task 1 suggested that they had not learnt how to do so. They were also taught in the familiarisation lesson that they could generalise by extending Type B tasks, but the habit to generalise whenever possible was only developed in the last lesson on extension (see Lesson 6 in Appendix C). Thus some of the students were not sure during the pretest that they could change the given in the task statement. For example, the following shows the protocols of one student (S7) during the pretest:

> "Last time I investigate total time taken to toast the three slices of bread. Then now I should ... change ... Can I change the question? Can I change to like 5 slices?" [S7; Pretest Task 2]

Another student (S3) changed the number of slices to be toasted without any intention to generalise at all. He first found the total toasting time for 6 slices and then for 12 slices. But he stopped there and struggled to think of another problem to investigate. This suggests that some of the students were either unsure during the pretest that they could change the given to generalise, or they did not fully understand that the purpose of changing the given was to generalise. However, Table 8.10 shows that all the students knew that they could change the given for Posttest Task 2 because all of them extended the task, but quite a number of them (S1-S6,S10) still changed the given without understanding that the purpose was to generalise. Therefore, there is a need for teachers to emphasise to their students that the purpose of changing the given in Type B tasks is to generalise.

Table 8.10 shows that metacognition was very much lacking in the problem-posing and extension stages. There was only one instance of analysing the feasibility of the goal (MG) during the extension of Posttest Task 2, but it was not effective. Despite teaching the students during Lesson 3 of the teaching experiment (see Appendix C) the need to analyse their problems or extensions to see whether they were worth pursuing or whether they were too trivial or too difficult to pursue, it seems that this process was not easy for them to pick up. However, this is an important metacognitive process. For example, one student (S1) pursued a challenging problem of finding a formula for the general term of a Kaprekar sequence for Posttest Task 1 without analysing whether the problem was feasible, and he failed to solve it; while 2 students (S2,S4) tried to find a formula for the amount of sausages each person will receive for Posttest Task 2, without realising that it was a very trivial problem as the result was clearly true. This suggests that teachers should focus on this metacognitive process to teach their students the need to analyse whether a problem is worth pursuing, or else the students might end up on a wild goose chase.

8.3.3 Specialising and Using Other Heuristics (Stage 3)

Table 8.13 shows the proficiency level of specialising and using other heuristics (S/H) attained by the 10 students for the four tasks, Table 8.14 shows the frequencies of metacognitive processes exhibited in this stage for the four tasks, while Table 8.15 shows the number of examples that the 10 students specialised systematically, purposefully and randomly for the two Type A tasks. The information for the two pretest tasks was extracted from Tables M1.3, M2.3a and M2.3b in Appendix M while the information for the two posttest tasks was obtained from Tables 7.4, 7.16 and 7.21 in the previous chapter.

Table 8.13 Proficiency Level of Specialising and Using Other Heuristics

Descriptors	Pretest 1 (Type A)	Posttest 1 (Type A)	Pretest 2 (Type B)	Posttest 2 (Type B)
Level 2 Descriptor A: Chose at least one set of systematic examples	0	1 (S9)	0	0
Level 2 Descriptor B: Used other heuristics effectively	0	0	1 (S4)	3 (S1,S4,S9)
Level 2: Level 2 Descriptors A and B	**0**	**1 (S9)**	**1 (S4)**	**3 (S1,S4,S9)**
Level 1 Descriptor A: Chose at least one example purposefully	5 (S3-S6,S10)	6 (S2-S7)	0	0
Level 1 Descriptor B: Used other heuristics quite effectively	0	0	3 (S5,S8,S10)	2 (S8,S10)
Level 1: Level 1 Descriptors A and B	**5 (S3-S6,S10)**	**6 (S2-S7)**	**3 (S5,S8,S10)**	**2 (S8,S10)**
Level 0 Descriptor A: Chose examples randomly	5 (S1,S2, S7-S9)	3 (S1,S8,S10)	0	0
Level 0 Descriptor B: Did not use other heuristics effectively	0	0	6 (S1-S3, S6,S7,S9)	5 (S2,S3, S5-S7)
Level 0: Level 1 Descriptors A and B	**5 (S1,S2, S7-S9)**	**3 (S1,S8,S10)**	**6 (S1-S3, S6,S7,S9)**	**5 (S2,S3, S5-S7)**
Average Score for Level (out of 2)	**0.5**	**0.8**	**0.5**	**0.8**
Standard Deviation (s.d.)	0.5	0.6	0.7	0.9

Table 8.14 Frequencies of Metacognitive Processes for Specialising and Using Other Heuristics

Processes	Pretest 1 (Type A)	Posttest 1 (Type A)	Pretest 2 (Type B)	Posttest 2 (Type B)
Analysed Feasibility of Plan (MF)	0	3 (S2,S5)	4 (S4,S5,S9)	3 (S1,S7)
Metacognitive Awareness (MA)	7 (S2,S5,S10)	2 (S2,S4)	2 (S4,S9)	1 (S7)
Total Frequency	**7 (S2,S5,S10)**	**5 (S2,S4,S5)**	**6 (S4,S5,S9)**	**4 (S1,S7)**

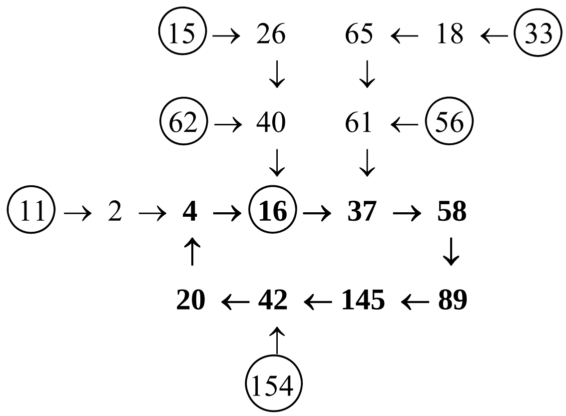
Table 8.15 Frequencies of Examples for Specialising

Processes	Pretest 1 (Type A)	Posttest 1 (Type A)
No. of Systematic Examples	0	2 [or 4%] (S9)
No. of Purposeful Examples	15 [or 19%] (S3-S6,S10)	16 [or 36%] (S2-S7,S9)
No. of Random Examples	66 [or 81%] (S1-S10)	27 [or 60%] (S1-S10)
Total No. of Examples	**81**	**45**

(a) Comparison between Pretest Task 1 and Posttest Task 1 (Type A)

Table 8.13 shows that the main process for the two Type A tasks at this stage was specialising. For Pretest Task 1, no student specialised systematically, 5 students specialised purposefully, and 5 students specialised randomly. For Posttest Task 1, one student specialised systematically, 6 students specialised purposefully, and 3 students specialised randomly. Thus they had performed slightly better in specialising for Posttest Task 1 than for Pretest Task 1. Table 8.15 shows that the students generated a lot more examples for Pretest Task 1 than for Posttest Task 1. This was partly because 7 students (S1-S3,S6,S8-S10) misinterpreted the pretest task by not repeating the process for the new number, so they tried separate examples with different starting numbers. But based on percentages, they generated more purposeful (36%) and systematic examples (4%) for the posttest task than for the pretest task.

It was observed from Table 8.14 that the students did not analyse the feasibility of their plan to specialise (MF) for Pretest Task 1, but 2 of them (S2,S5) engaged in this metacognitive process for a total of 3 times for Posttest Task 1. However, MF was not effective in helping the students to specialise more systematically (see data analysis in Section 7.2.3). On the other hand, Table 8.14 shows that there were more instances of metacognitive awareness (MA) being observed for Pretest Task 1 than for Posttest Task 1. This was probably because of the nature of the underlying patterns for the task. For the pretest task, the sequences merge very frequently. The following shows some sequences of sad numbers (the reader should refer to the task analysis in Appendix E to be familiar with these numbers). For example, a student who started with 11 for one sequence and 15 for another sequence will end up with the same number 16 at some point. In fact, there are many sequences that will also reach the number 16, e.g. a sequence with starting number 33, 56, 62 or 154, just to name a few.



But for Posttest Task 1, two Kaprekar sequences that start with two different self numbers will not merge when the numbers are less than 100 (see Appendix E). Since it was rarer for the sequences in Posttest Task 1 to merge than for the sequences in Pretest Task 1, there would be fewer opportunities for the students to be aware that a term in a particular sequence was equal to a term in another sequence for the posttest task, even if they might have possessed the ability. Metacognitive awareness is important because some students did not even realise when a sequence they were investigating had already merged with another sequence they had investigated earlier, so they wasted precious time by finding more terms for the sequence that they were investigating. For example, one student (S4) obtained the number 89 in the sequence in his Example 5 for Pretest Task 1, but he was not aware that 89 had also appeared in the sequence in his Example 2, so he wasted precious time by finding more terms for the sequence in his Example 5. However, for Posttest Task 1, the same student obtained the number 119 in the sequence in his Example 2, but he realised that 119 had also appeared in the sequence in his Example 1, so he did not waste time finding more terms for the sequence in his Example 2 (see Fig. 7.1 in Section 7.2.3).

(b) Comparison between Pretest Task 2 and Posttest Task 2 (Type B)

Table 8.13 shows that the main process for the two Type B tasks in this stage was using other heuristics. For Pretest Task 2, only one student used other heuristics ‘effectively’ to discover Toasting Method B that gives the least toasting time (see task analysis in Appendix E) while 3 students used other heuristics ‘quite effectively’, i.e. they struggled but finally discovered Method B. The remaining 6 students used other heuristics ‘ineffectively’ as they just used the usual Toasting Method A without discovering Method B. For Posttest Task 2, 3 students used other heuristics ‘effectively’ to discover Cutting Method B that gives the least number of cuts, while 2 students used other heuristics ‘quite effectively’, i.e. they struggled but finally found Method B (see data analysis in Section 7.3.3). The remaining 5 students used other heuristics ‘ineffectively’ as they just used the usual Cutting Method A without discovering Method B. Thus the students performed slightly better in using other heuristics for Posttest Task 2 than for Pretest Task 2. Table 8.14 shows that 3 students (S4,S5,S9) analysed the feasibility of their plan to use other heuristics (MF) on 4 occasions for Pretest Task 2, while 2 students (S1,S7) engaged in this metacognitive process on 3 occasions for Posttest Task 2. For the pretest task, the 3 students analysed the feasibility of pursuing different possibilities after toasting one side of the first two slices of bread. The following shows the protocols of one of the students.

> “After these 30 seconds … I will … what should I do ah? Should I take out both? Or just turn one and take out another one?” [S4; Pretest Task 2]

In the end, the student (S4) decided to take out one slice but turn over the other slice. In this manner, he discovered Toasting Method B that gives the least toasting time. But the second student (S5), despite analysing the feasibility of pursuing different possibilities using the same reasoning, did not discover Method B until towards the end of the test. The third student (S9) analysed different toasting methods on two occasions, such as toasting one slice first before toasting the other two slices, and toasting one slice at a time, but his metacognitive behaviours were not effective in helping him discover Method B. For Posttest Task 2, the first student (S1) analysed the feasibility of using Cutting Method A for his extension, which helped him to understand the idea behind the cutting method so that he could apply it successfully to his extension. But analysing the feasibility of using multiples or factors to think of a cutting method on two occasions was not effective for the other student (S7) since the cutting method does not depend on multiples or factors per se[26] (see data analysis in Sections 7.3.3 and 7.3.7b). It was further observed from Table 8.14 that 2 students (S4,S9) engaged in metacognitive awareness (MA) on 2 occasions for Pretest Task 2, while one student (S7) exhibited MA on one occasion for Posttest Task 2. For the pretest task, the first student (S4) realised that the total time for toasting the 3 slices of bread using Toasting Method A was too big as shown in his protocols:

> "And this number is surely very big [double underline: 159 s] and there must be a way which is even faster. So I will now think of a way …" [S4; Pretest Task 2]

It is puzzling why the student would think that 159 seconds was too long a timing, but this metacognitive awareness had resulted in his search for a shorter method (i.e. Toasting Method B), which he found soon afterwards. For the second student (S9), he was extending the task to toast 4 slices of bread. He was aware immediately that he could use the total toasting time for toasting 2 slices of bread in Toasting Method A

[26] Although students can use the LCM to think of Cutting Method A, and the general formula for the least number of cuts depends on the HCF, the student (S7) did not find the LCM or the HCF: she just used multiples and factors.

(for which he finished toasting the first 2 slices before toasting the third slice), and multiply it by 2 to give the total toasting time for the 4 slices. This might seem obvious to the reader, but there were 3 students (S3,S4,S7) who actually wasted precious time by going through the steps of toasting the slices starting from 2 slices all over again. Thus metacognitive awareness had resulted in S9 saving precious time. For Posttest Task 2, the student (S7) was aware that she had tried a cutting method before, thus saving her precious time from trying the same method again (see data analysis in Section 7.3.3).

(c) Development of the Processes of Specialising and Using Other Heuristics

The above analysis shows that the students had improved slightly in the process of specialising for Posttest Task 1 compared with Pretest Task 1. However, systematic specialising was still lacking in the posttest task. This is indeed surprising because the students were taught to specialise systematically to search for patterns during Lesson 4 of the teaching experiment (see Appendix C) and they usually specialised systematically during the other lessons as well. So it appears that the students were confused by the instruction in the task statement of Posttest Task 1: "Choose any number." They seem to take this literally as choosing a number randomly. Thus most of the students did not specialise systematically for the posttest task despite being taught to do so during the teaching experiment. As such, care must be taken to teach students to specialise systematically to search for patterns even in such situations.

For Type B tasks, the students had improved slightly in using other heuristics, such as reasoning, more effectively to solve the problems posed. These heuristics were usually task specific, and they were developed by letting students try various Type B tasks during the teaching experiment, rather than taught specifically, although effective reasoning was part of the focus for Lesson 4. Both the pretest and the posttest data show that there is a lot of scope for developing in the students effective reasoning skills because the average score was rather low.

We will now examine some examples of ineffective reasoning for both Type B tasks. For Pretest Task 2, one student (S3) used a toasting method similar to Toasting Method A to toast the 3 slices of bread. Then he wanted to find out whether the total time taken to toast one slice was equal to the total toasting time for the 3 slices divided by 3. Although the answer was clearly negative (since Method A involves toasting the first 2 slices *together* before toasting the third slice), he actually went through the steps of toasting one slice on both sides in order to find the total toasting time. When he realised that the answer was negative, he wanted to know what had caused the difference. So he went through the steps of toasting the first 2 slices all over again, before he was able to account for the difference. Thus he lacked the ability to reason effectively because he took so long to understand such a simple fact.

Another student (S9) wanted to find the fastest way to toast the 3 slices of bread for Pretest Task 2. He started with Toasting Method A. Then he decided to toast one slice at a time (Toasting Method C). Although it was evident that Method C would take a longer time than Method A (since Method A involves toasting the first 2 slices *together*), he did not realise that. Instead, he went through the steps of toasting one slice at a time. When he finally calculated the total toasting time, he was shocked:

> "What?! So we toast one bread by one bread is a bad idea." [S9; Pretest Task 2]

Then he tried what he called an 'ingenious way of cooking [*sic*] the bread':

> "You know we can turn the bread outside, right, so that it doesn't affect us when we are turning the bread inside … like, cost time. So this is the creatively, make, making, so I'm creatively making use of time." [S9; Pretest Task 2]

In other words, after toasting the first two slices on one side, he decided to take out the two slices and then put them back in the grill because he believed that this action would not involve turning the slices, so he would save the 3 seconds to turn a slice in the grill. But this method will be much longer since it takes 5 seconds to take out a slice and another 5 seconds to put the slice back into the grill. However, the student did not realise this, so he went through the steps of toasting the 3 slices using his 'ingenious way'. When he finally calculated the total toasting time, he was shocked:

> "Oh my god! It takes more time!" [S9; Pretest Task 2]

But this was the same student who later discovered the complicated general formula for the least number of cuts for Posttest Task 2 as described in Section 6.3(a). It is indeed puzzling that he was unable to observe this simple fact in the pretest but he was able to reason so effectively in the posttest.

For Posttest Task 2, most of the students were also unable to reason effectively, although there was a slight improvement. Although 7 students could reason that each person will get 2/3 of a sausage for Posttest Task 2, only 3 of them were able to see straightaway that each sausage could be cut at the 2/3-mark (Cutting Method B) instead of dividing each sausage into 3 equal parts (Cutting Method A), while another 2 students had to struggle to discover Method B (see data analysis in Section 7.3.3).

What was surprising was that 3 students (S2,S5,S7) even attempted to cut each sausage into 18 equal parts (Cutting Method C) in order to share the 12 sausages equally among the 18 people, just like 3 students (S1,S6,S9) who attempted Toasting Method C (toast one slice at a time) for Pretest Task 2 when two slices could be toasted together. Thus there is a lot of room for improvement in teaching the students how to reason effectively.

One possible reason that the students toasted the bread or cut the sausages any old how was that they did not pause to analyse the feasibility of their plan (MF). The above analysis shows that very few students actually engaged in this metacognitive process, which was helpful only in very few cases. During Lesson 4 of the teaching experiment (see Appendix C), the students were taught the need to analyse their plan to see whether it was worth pursuing. But just like the analysis of the feasibility of the goal (MG) in the problem-posing stage and the extension stage, the analysis of the feasibility of the plan in the stage of specialising and using other heuristics was also not easy for the students to pick up. Therefore, there is a need for teachers to focus more on developing this important metacognitive process.

Metacognitive awareness is a newly discovered process during the course of the present research. Although it was already exhibited in the pretest, the limited time window between the pretest and the developing lessons did not allow me to transcribe and code the 20 pretest transcripts in time to discover this process. As such, I did not teach or develop this important process during the teaching experiment, as explained earlier in this section. Moving forward, there is a need for further research to study how this process could be developed in the students.

8.3.4 Conjecturing (Stage 4)

Table 8.16 shows the proficiency level of conjecturing (C) attained by the 10 students for the four tasks, Table 8.17 shows the frequencies of conjecturing outcomes, while Table 8.18 shows the different processes that the students used to search for patterns. The information for the two pretest tasks was extracted from Tables M1.4a, M1.4b, M2.4a and M2.4b in Appendix M while the information for the two posttest tasks was obtained from Tables 7.8, 7.9, 7.17 and 7.23 in the previous chapter.

Table 8.16 Proficiency Level of Conjecturing

Descriptors	Pretest 1 (Type A)	Posttest 1 (Type A)	Pretest 2 (Type B)	Posttest 2 (Type B)
Level 2: Formulated at least one correct non-trivial conjecture	**1 (S5)**	**3 (S5,S9,S10)**	**1 (S4)**	**5 (S1,S3, S8-S10)**
Level 1 Descriptor A: Observed at least one correct non-trivial pattern	3 (S2,S4,S10)	2 (S3,S8)	0	0
Level 1 Descriptor B: Formulated at least one correct trivial conjecture	2 (S8,S9)	0	1 (S9)	2 (S2,S4)
Level 1: Level 1 Descriptors A and B	**5 (S2,S4, S8-S10)**	**2 (S3,S8)**	**1 (S9)**	**2 (S2,S4)**
Level 0 Descriptor A: Did not observe any pattern	0	0	0	0
Level 0 Descriptor B: Did not formulate any conjecture	3 (S3,S6,S7)	1 (S7)	7 (S2,S5-S8,S10)	3 (S5-S7)
Level 0 Descriptor C: Observed incorrect patterns	4 (S1,S3, S6,S7)	5 (S1,S2, S4,S6,S7)	0	0
Level 0 Descriptor D: Formulated incorrect conjectures	1 (S1)	3 (S1,S2,S4)	2 (S1,S3)	0
Level 0 Descriptor E: Observed correct trivial patterns	1 (S3)	0	0	0
Level 0: Level 1 Descriptors A-E	**4 (S1,S3, S6,S7)**	**5 (S1,S2, S4,S6,S7)**	**8 (S1-S3, S5-S8,S10)**	**3 (S5-S7)**
Average Score for Level (out of 2)	**0.7**	**0.8**	**0.3**	**1.2**
Standard Deviation (s.d.)	0.6	0.9	0.6	0.9

* It was possible for a student to fulfil more than one descriptor in the level that he or she was in, e.g. 4 students were at Level 0 for Pretest Task 1, but 3 of them (S3,S6,S7) satisfied more than one descriptor in that level. However, if a student was at a higher level but also fulfilled some descriptors at a lower level, he or she would *not* be included at the lower level to avoid confusion, e.g. one other student (S5) also satisfied Level 1 Descriptor A for Pretest Task 1, but she was not included at that level since she was already at Level 2.

Table 8.17 Frequencies of Outcomes for Conjecturing

Outcomes	Pretest 1 (Type A)	Posttest 1 (Type A)	Pretest 2 (Type B)	Posttest 2 (Type B)
No. of Non-Trivial Conjectures	2* (S2,S5)	5 (S2,S5,S9,S10)	9 (S1,S3,S4,S9)	9* (S1-S3,S8-S10)
No. of Trivial Conjectures	7 (S1,S8,S9)	7* (S1,S3-S6,S9)	2* (S4,S9)	3* (S2,S4,S8)
No. of Correct Conjectures	4 (S5,S8,S9)	4 (S5,S9,S10)	4 (S4,S9)	9 (S1-S4,S8-S10)
No. of Incorrect Conjectures	5* (S1,S2,S9)	8* (S1-S6,S9,S10)	7* (S1,S3,S9)	3* (S2,S8,S9)
Total No. of Conjectures	**9 (S1,S2,S5,S8,S9)**	**12 (S1-S6,S9,S10)**	**11 (S1,S3,S4,S9)**	**12 (S1-S4,S8-S10)**

* These figures are the actual ones for the outcomes. They are different from those in Table 8.16 as the figures in Table 8.16 are only for the students in that level. For example, S2 had formulated an incorrect non-trivial conjecture for Pretest Task 1, so it was included in this table, but this Level 0 outcome was excluded from Table 8.16 to avoid confusion since the student was already in Level 1.

Table 8.18 Processes for Searching for Patterns

Processes	Pretest 1 (Type A)	Posttest 1 (Type A)
No. of students who searched for patterns in the terms of the sequence	5 (S2,S4,S5,S7,S10)	3 (S2,S5,S8)
No. of students who searched for patterns in the differences between consecutive terms of the sequence	5 (S2,S3,S6,S9,S10)	10 (S1-S10)
No. of students who searched for other patterns within the sequence	3 (S1,S8,S10)	4 (S5,S7,S8,S10)
No. of students who searched for patterns across sequences	0	0
No. of students who exhibited metacognitive processes (MF or MA)	0	0

(a) Comparison between Pretest Task 1 and Posttest Task 1 (Type A)

Table 8.16 shows that 3 students were at Level 2 for formulating at least one correct non-trivial conjecture for Posttest Task 1, but only one student was at the same level for Pretest Task 1. However, 5 students were at Level 1 for the pretest task compared with only 2 students for the posttest task. Therefore, on average, there was negligible difference in their performance between the pretest (score = 0.7) and the posttest (score = 0.8). On the other hand, Table 8.17 shows that more students were able to formulate conjectures for the posttest task than for the pretest task: 8 students had formulated a total of 12 conjectures for Posttest Task 1, out of which 5 were non-trivial, while only 5 students had formulated a total of 9 conjectures for Pretest Task 1, out of which only 2 were non-trivial. Although the students had formulated more non-trivial conjectures for Posttest Task 1 than Pretest Task 1, the number of correct conjectures was the same for both tasks, which accounted for the negligible difference between the scores for the two tests.

For Pretest Task 1, sequences of sad numbers will end in a loop while sequences of happy numbers will terminate at the number 1 (the reader should refer to Appendix E to be familiar with the different patterns for the two tasks). Thus 5 students were on the right track when they searched for patterns in the terms of the sequence, as shown in Table 8.18. However, one of them (S7) misinterpreted the task and did not recover. Another one (S2) observed only sad numbers as he did not specialise systematically, and he spent the rest of his time trying to find out why it worked this way but failed. Two other students (S4,S10) treated the happy numbers as exceptions instead of another pattern in its own right: it seems that they were unable to accept that a sequence can terminate, or that there is more than one type of patterns. Only one student (S5) was able to formulate the non-trivial conjecture that there are only these two types of sequences. For Posttest Task 1, 3 students observed the Type 1a 'multiples' pattern, but only one of them (S5) formulated it as a non-trivial conjecture. The other 2 students (S3,S8) failed to accept this as the underlying pattern because this was not one of the usual patterns that they had learnt in school (see data analysis in Section 7.2.4). Although all the 10 students tried to search for patterns in the differences between consecutive terms, they were unable to observe the much more complicated 'digital roots' pattern, except for one student (S10).

(b) Comparison between Pretest Task 2 and Posttest Task 2 (Type B)

Table 8.16 shows that 5 students were at Level 2 for formulating at least one correct non-trivial conjecture for Posttest Task 2, but only one student was at the same level for Pretest Task 2. Moreover, 2 students were at Level 1 for Posttest Task 2 compared with only one student for Pretest Task 2. Thus only 3 students were at Level 0 for the posttest task while 8 students were at the same level for the pretest task. Therefore, they had performed much better in conjecturing for Posttest Task 2 than for Pretest Task 2. But Table 8.17 shows that the number of conjectures formulated for both tasks was about the same, although more students were able to formulate conjectures for the posttest task than for the pretest task: 8 students had formulated a total of 12 conjectures for Posttest Task 1 while only 4 students had formulated a total of 11 conjectures for Pretest Task 1. The number of non-trivial conjectures formulated was the same for both tasks, but there were far more correct conjectures for Posttest Task 1 than Pretest Task 1, which accounted for the much higher posttest score.

For Pretest Task 2, most of the students (except S7 and S8) posed the problem of finding the least toasting time for the original task (see Table M2.2a in Appendix M), but only 4 of them discovered Toasting Method B that gives the least toasting time (see Table M2.3a). This means that the other 6 students, who used another toasting method, would be unable to formulate a correct conjecture about the least toasting time. However, out of the 4 students who used Method B, one student (S8) did not pose the problem of finding the least toasting time, and 2 students (S5,S10) did not finish solving the problem. Thus only the last student (S4) obtained a correct non-trivial conjecture about the least toasting time, so he was at Level 2. For the extension, 4 students extended the task (see Table M2.2c), but only one of them (the same student S4) obtained a correct non-trivial conjecture using Method B to generalise the least toasting time (see Table M2.4b). One other student (S9) posed a correct trivial conjecture, so he was at Level 1. Therefore, so few students (only 2 of them) were at Level 1 or 2 because (i) most students were unable to discover Toasting Method B to formulate a correct conjecture, and (ii) most students did not extend the task, so there were fewer opportunities to formulate other conjectures as there was essentially only one conjecture for the original task.

For Posttest Task 2, only 5 students discovered Cutting Method B that gives the least number of cuts, but one of them (S4) did not pose the problem of finding the least number of cuts. In the end, only 4 students (S1,S8-S10) obtained a correct non-trivial conjecture about the least number of cuts (see data analysis in Sections 7.3.2 to 7.3.4), so they were at Level 2. For the extension, all the students extended the task in different ways, not necessarily to generalise the least number of cuts. Thus there were more opportunities for one more student (S3) to formulate a correct non-trivial conjecture (Level 2) and 2 more students (S2,S4) to formulate a correct trivial conjecture each (Level 1) for the extension (see data analysis in Section 7.3.8). Therefore, the majority of the students (i.e. 7 of them) were at Level 1 or 2 because (i) slightly more students were able to discover Method B to formulate a correct conjecture for the posttest task than the pretest task, and (ii) all the students extended the task, thus there were more opportunities to formulate other conjectures.

(c) Development of Conjecturing Processes

The above analysis shows that there was not much difference in the conjecturing process among the students between the pretest and the posttest for the two Type A tasks, but there was a big improvement for the two Type B tasks. During Lesson 4 of the teaching experiment (see Appendix C), the students were taught to specialise systematically for Type A tasks to search for patterns. However, the students did not specialise systematically for Posttest Task 1, which had affected their ability to observe the underlying patterns (conjecturing). Moreover, the tasks used in that lesson were not on sequences although one of the two tasks still involved number patterns: palindromic numbers (see Appendix D). During Lesson 2 when the students were given more time to attempt Pretest Task 1 after some discussion on what they could have investigated during the test, the students were guided to search for patterns within the sequence and across sequences, but the focus during that lesson was on understanding the task. In fact, all the other tasks used in the teaching experiment did not involve sequences. Therefore, there is room for improvement in teaching students to search for patterns within and across sequences, especially in identifying unfamiliar patterns which they have not learnt before.

For the two Type B tasks, the factors that affect whether the students will formulate a correct conjecture include (i) the use of effective reasoning to find the optimal toasting or cutting method, and (ii) the habit to extend the task to generalise, which will provide the students further opportunities to formulate other conjectures. As discussed in Section 8.3.3, the students had learnt to be slightly more effective in using reasoning for the posttest task than the pretest task, and as discussed in Section 8.3.2, all the students had developed the habit to extend the posttest task compared with very few students who extended the pretest task. Therefore, the students performed a lot better in the conjecturing stage because of these processes in other stages.

Despite teaching the students in Lesson 4 of the teaching experiment the need to analyse the feasibility of their plan to conjecture (MF) for both types of tasks, the students did not exhibit this metacognitive behaviour during the posttest (nor did they exhibit this process on their own during the pretest). If they had, it might have helped them, for example, to realise that they could search for patterns across sequences for Posttest Task 1, which might lead them to discover the self numbers (see task analysis in Appendix E); or to reason more effectively so as to discover the optimal cutting method in order to formulate a correct conjecture about the least number of cuts for Posttest Task 2. Therefore, there is a need for teachers to focus on developing this metacognitive process when teaching their students to do investigation.

8.3.5 Justifying and Generalising (Stages 5 and 6)

Table 8.19 shows the proficiency level of justifying and generalising (J/G) attained by the 10 students for the four tasks, Table 8.20 shows the frequencies of justifying and generalising outcomes, while Table 8.21 shows the frequencies of justifying processes that might have helped the students to prove or refute their conjectures. The information for the two pretest tasks was extracted from Tables M1.5, M2.5a and M2.5b in Appendix M while the information for the two posttest tasks was obtained from Tables 7.10, 7.18 and 7.24 in the previous chapter.

Table 8.19 Proficiency Level of Justifying and Generalising

Descriptors	Pretest 1 (Type A)	Posttest 1 (Type A)	Pretest 2 (Type B)	Posttest 2 (Type B)
Level 2: Correctly proved at least one non-trivial conjecture	**0**	**2 (S5,S9)**	**0**	**2 (S1,S10)**
Level 1 Descriptor A: Correctly proved at least one trivial conjecture that will lead to generalisation	0	0	1 (S4)	0
Level 1 Descriptor B: Refuted at least one incorrect non-trivial conjecture	0	1 (S2)	0	0
Level 1 Descriptor C: Tried but failed to prove at least one correct non-trivial conjecture	0	0	0	2 (S8,S9)
Level 1: Level 1 Descriptors A-D	**0**	**1 (S2)**	**1 (S4)**	**2 (S8,S9)**
Level 0 Descriptor A: Did not formulate any conjecture to justify	5 (S3,S4, S6,S7,S10)	2 (S7,S8)	8 (S1-S3, S5-S8,S10)	3 (S5-S7)
Level 0 Descriptor B: Did not prove any conjecture due to lack of time	0	3 (S3,S4,S10)	0	0
Level 0 Descriptor C: Tried but failed to justify any conjecture	2 (S1,S2)	0	0	0
Level 0 Descriptor D: Correctly proved trivial conjectures that will not lead to any generalisation	0	0	0	0
Level 0 Descriptor E: Refuted incorrect trivial conjectures	1 (S9)	0	0	0
Level 0 Descriptor F: Wrongly accepted all conjectures as true without testing	1 (S8)	1 (S1,S6)	3 (S1,S3,S9)	1 (S2)
Level 0 Descriptor G: Wrongly accepted all conjectures as true based on naïve testing	4 (S1,S5, S8,S9)	0	0	3 (S2-S4)
Level 0: Level 0 Descriptors A-G	**10 (S1-S10)**	**7 (S1,S3,S4, S6-S8,S10)**	**9 (S1-S3, S5-S10)**	**6 (S2-S7)**
Average Score for Level (out of 2)	**0**	**0.5**	**0.1**	**0.6**
Standard Deviation (s.d.)	0	0.8	0.3	0.8

* It was possible for a student to fulfil more than one descriptor in the level that he or she was in. However, if a student was at a higher level but also fulfilled some descriptors at a lower level, he or she would *not* be included at the lower level to avoid confusion.

Table 8.20 Frequencies of Outcomes for Justifying and Generalising

Outcomes	Pretest 1 (Type A)	Posttest 1 (Type A)	Pretest 2 (Type B)	Posttest 2 (Type B)
Test ended after formulating conjecture	0	3 (S3,S4,S10)	0	0
Wrongly accepted conjecture as true without testing	1 (S8)	2 (S1,S6)	10* (S1,S3,S4,S9)	4* (S2,S8,S9)
Test ended during naïve testing	1 (S1)	0	0	0
Test ended during justifying using non-proof argument	0	1* (S9)	0	0
Wrongly accepted conjecture as true based on naïve testing	4 (S1,S5,S8,S9)	0	0	3 (S2-S4)
Refuted incorrect conjecture based on naïve testing	1 (S9)	3* (S2,S5,S9)	0	0
Tried non-proof argument but failed to justify conjecture	1 (S2)	0	0	3* (S8,S9)
Tried formal proof but failed to justify conjecture	1 (S1)	0	0	0
Proven conjecture using non-proof argument, which led to generalisation	0	3 (S5,S9)	1 (S4)	0
Proven conjecture using non-proof argument, which did not lead to any generalisation	0	0	0	2 (S1,S10)
Proven conjecture using formal proof	0	0	0	0
Total No. of Conjectures	9 (S1,S2, S5,S8,S9)	12 (S1-S6,S9,S10)	11 (S1,S3,S4,S9)	12 (S1-S4, S8-S10)
Total No. of Proven Conjectures	**0**	**3 (S5,S9)**	**1 (S4)**	**2 (S1,S10)**
Total No. of Generalisation	**0**	**3 (S5,S9)**	**1 (S4)**	**0**

* These figures are the actual ones for the outcomes. They are different from those in Table 8.18 as the figures in Table 8.18 are only for the students in that level. For example, S5 and S9 also refuted incorrect conjectures for Posttest Task 1, so they were included in this table, but they were excluded from Table 8.18 for this Level 1 outcome to avoid confusion since they were already in Level 2.

Table 8.21 Frequencies of Processes for Justifying

Processes	Pretest 1 (Type A)	Posttest 1 (Type A)	Pretest 2 (Type B)	Posttest 2 (Type B)
Naïve Testing	6 (S1,S5,S8,S9)	4 (S2,S5,S9)	0	5 (S2-S4,S8,S9)
Non-proof Argument	1 (S2)	4 (S5,S9)	1 (S4)	5 (S1,S8-S10)
Formal Proof	1 (S1)	1 (S5)	0	0
Analysed Feasibility of Plan (MF)	0	0	0	1 (S9)
Metacognitive Awareness (MA)	0	0	0	0

(a) Comparison between Pretest Task 1 and Posttest Task 1 (Type A)

Table 8.19 shows that all the 10 students were at Level 0 for Pretest Task 1 but 7 students were at the same level for Posttest Task 1. One of the remaining 3 students was at Level 1 while the other 2 students were at Level 2 for the posttest task. Thus they had performed better in justifying for Posttest Task 1 than for Pretest Task 1. Table 8.20 shows that more students had formulated more conjectures to justify for the posttest task than for the pretest task: 8 students had formulated a total of 12 conjectures for Posttest Task 1 compared with only 5 students who had formulated a total of 9 conjectures for Pretest Task 1. But this cannot be the main reason to account for the difference in performance in justifying since the number of correct conjectures formulated for each of the two tasks to be proven was the same at 4 (see Table 8.17). In other words, it depends on what the students did with their correct conjectures. On closer analysis of Pretest Task 1 in Table M1.5 in Appendix M, it was observed that all the 3 students (S5,S8,S9) wrongly accepted all their 4 correct conjectures as true without testing or based on naïve testing. But for Posttest Task 1, only one student (S10) did not justify her correct conjecture as the test ended just after she formulated it, while the other 2 students (S5,S9) proved their 3 conjectures correctly using a non-proof argument (see data analysis in Section 7.2.5).

Table 8.20 shows that 4 students (S1,S5,S8,S9) wrongly accepted 5 conjectures (4 correct and 1 incorrect) as true without testing or based on naïve testing for Pretest Task 1. But for Posttest Task 1, only 2 students (S1,S6) accepted 2 (incorrect) conjectures as true without testing, but no student accepted any conjecture as true based on naïve testing. Table 8.21 also shows that the students used naïve testing slightly more often for Pretest Task 1 than for Posttest Task 1, but they used a non-proof argument slightly more often for the posttest task[27]. This suggests that more students had understood during the posttest that they should not accept a conjecture as true based on naïve testing, but they must justify their conjecture using a non-proof argument or a formal proof. But very few students attempted a formal proof using algebra, and the 2 students (S1,S5) who tried were unable to justify any conjecture. The students also did not analyse the feasibility of their plan to justify (MF), and they did not exhibit any metacognitive awareness (MA) at this stage for either task.

(b) Comparison between Pretest Task 2 and Posttest Task 2 (Type B)

Table 8.19 shows that only one student was at Level 1 while all the other 9 students were at Level 0 for Pretest Task 2, compared with 2 students at Level 2 and 2 students at Level 1 for Posttest Task 2. Thus they had performed better in justifying for Posttest Task 2 than for Pretest Task 2. Unlike the analysis of the two Type A tasks earlier, the students formulated about the same number of conjectures for the two Type B tasks, but there were far more correct conjectures formulated for Posttest Task 2 than for Pretest Task 2: Table 8.17 shows that there were 9 correct conjectures for the posttest task compared with only 4 correct conjectures for the pretest task. With more correct conjectures to justify for Posttest Task 2, it was possible that they had a higher chance to prove these conjectures for the posttest task than for the pretest task. On closer analysis of Pretest Task 2 in Tables M2.5a and M2.5b in Appendix M, it was observed that 2 students (S4,S9) accepted a total of 3 correct conjectures as true without testing, and only one student (S4) proved a trivial conjecture correctly using a non-proof argument. But for Posttest Task 2, 5 students (S2-S4,S8,S9) wrongly

[27] Although S2 tried to use a non-proof argument in Pretest Task 1 to explain the pattern of sad numbers, he was unsuccessful.

accepted a total of 5 correct conjectures as true without testing or based on naïve testing, while 4 students (S1,S8-S10) tried to justify the other 4 correct conjectures, with only 2 of them (S1,S10) succeeding in proving their 2 conjectures correctly using a non-proof argument (see data analysis in Sections 7.3.5 and 7.3.9).

Table 8.20 shows that for Pretest Task 2, 4 students accepted a total of 10 conjectures (3 correct and 7 incorrect) as true without testing, and no student accepted any conjecture as true based on naïve testing. But for Posttest Task 2, 5 students accepted a total of 7 conjectures (5 correct and 2 incorrect) as true without testing or based on naïve testing. Table 8.21 shows that the students engaged in the justifying processes a lot more often for Posttest Task 2 than for Pretest Task 2. Half the number of students engaged in naïve testing for the posttest task because they had extended the task to generalise, so their conjectures would involve different numbers of sausages and people, thus allowing the possibility to test their conjectures using naïve testing. But for the pretest task, 6 of the 10 students did not extend the task to generalise, so there were fewer opportunities for them to test their conjectures using naïve testing. In fact, only 3 students (S3,S4,S9) managed to formulate a total of 7 conjectures to generalise for Pretest Task 2, but 6 of these conjectures were accepted as true without testing. Only one student (S4) succeeded in proving the non-trivial conjecture about a general formula for the least time needed to toast n slices of bread by using a non-proof argument (see Table M2.5b in Appendix M). However, for the posttest task, Table 8.21 shows that 4 students had learnt to use a non-proof argument to try to justify their 5 conjectures. No student attempted a formal proof for both tasks.

The students did not exhibit any metacognitive awareness (MA) in this stage for either task, but one of them (S9) analysed the feasibility of his plan to justify his conjecture (MF). The following shows his protocols after he had found the general formula for the least number of cuts:

> "But I don't know how to prove it. I think it is a little off my limits. How do I prove this conjecture? The conjecture is uh, a little bit pretty much complicated. Eh, no, the conjecture is not complicated, but is very hard to prove." [S9; Posttest Task 2]

Nevertheless, the student decided to try to prove the conjecture by using a non-proof argument to explain why the general formula works because there was still a bit of time left for the test and there was nothing else for him to do. But he failed to justify because the proof was not easy, which he had realised earlier.

(c) Development of Justifying Processes

The above analysis shows that the students had improved in the process of justifying for both the posttest tasks compared with the two pretest tasks due to three factors: (i) the students had formulated more correct conjectures to justify for Posttest Task 2 than for Pretest Task 2, thus increasing the chances of proving the conjectures, (ii) more students had extended Posttest Task 2 to generalise, thus providing more opportunities to formulate and justify other conjectures, and (iii) more students had realised in the posttest that they should prove their conjectures, instead of accepting their conjectures as true without testing or based on naïve testing, which happened more frequently during the pretest.

In Lesson 5 of the teaching experiment (see Appendix C), the students were taught that an observed pattern might not be true by using Task 9 (Chords and Regions). The pattern for the maximum number of regions in a circle formed by the chords, where *n* is the number of points on the circumference of the circle, is 1, 2, 4, 8, 16, … (see Appendix D for the task). Most people would guess that the formula for the general term is 2^{n-1}, but the formula based on the observed pattern is actually wrong. This example was also used to show that naïve testing is not fool proof since the formula works for $n = 1$ to 5, but it fails when $n = 6$. Thus the students were convinced of the need to treat an observed pattern as a conjecture: it could be refuted by a counter example from naïve testing, but it must be proven by a non-proof argument or a formal proof. As the actual formula[28] for Task 9 was beyond the students, Task 10 (Tiles) was used for the students to learn how to prove a formula by using a non-proof argument and a formal proof. This task was selected partly for its complicated formula for the number of cracked square tiles, $m + n - \text{LCM}(m, n)$, where m and n are the dimensions of the rectangle, which in a way is similar to the complicated formula for the least number of cuts for Posttest Task 2, $m - \text{HCF}(m, n)$, where m is the number of people and n is the number of sausages.

Hence, some of the students realised the need to prove their conjectures in the posttest while others still did not do so. For example, 2 students (S1,S10) proved that their cutting method will give the least number of cuts for Posttest Task 2, but none of the students realised that they had to prove that their toasting method will give the least toasting time for Pretest Task 2. Similarly, none of the students accepted their conjectures as true based on naïve testing for Posttest Task 1 compared with 4 students who did so for Pretest Task 1.

---

[28] The correct formula for the maximum number of regions in the circle is $\binom{n}{4} + \binom{n-1}{2} + \binom{n}{1}$.

Despite teaching the students in Lesson 5 of the teaching experiment the need to analyse the feasibility of their plan to justify (MF) for both types of tasks, they did not exhibit this metacognitive process during the posttest (nor did they exhibit this behaviour on their own during the pretest). If they had, it might have helped them to realise that they must not accept their conjectures as true without testing or based on naïve testing. Thus there is a need for teachers to focus on developing this metacognitive process when teaching their students to do investigation.

8.3.6 Checking (Stage 7)

Table 8.22 shows the proficiency level of checking (R) attained by the 10 students for the four tasks, Table 8.23 shows the frequencies of checking outcomes, while Table 8.24 shows the frequencies of checking and monitoring processes that might have helped the students to discover their mistakes. The information for the two pretest tasks was extracted from Tables M1.6 and M2.6 in Appendix M while the information for the two posttest tasks was obtained from Tables 7.11 and 7.25 in the previous chapter. The errors in these tables do not include errors due to misinterpretation and errors due to accepting conjectures as true without testing or based on naïve testing, as these had been dealt with in the understanding stage and justifying stage respectively.

Table 8.22 Proficiency Level of Checking

Descriptors	Pretest 1 (Type A)	Posttest 1 (Type A)	Pretest 2 (Type B)	Posttest 2 (Type B)
Level 2 Descriptor A: Made major errors but discovered all on time (i.e. within 5 min)	0	1 (S9)	0	1 (S9)
Level 2 Descriptor B: Did not make any major error, but often checked working, reviewed solution and monitored progress	1 (S7)	1 (S7)	1 (S9)	2 (S5,S7)
Level 2: Level 2 Descriptors A and B	**1 (S7)**	**2 (S7,S9)**	**1 (S9)**	**3 (S5,S7,S9)**
Level 1 Descriptor A: Made major errors but discovered only some of them	1 (S9)	3 (S3-S5)	0	1 (S8)
Level 1 Descriptor B: Made major errors and discovered all of them but some late (i.e. after more than 5 min)	1 (S2)	0	0	0
Level 1 Descriptor C: Did not make any major error, but occasionally checked working, reviewed solution and monitored progress	0	0	1 (S4)	1 (S2)
Level 1: Level 1 Descriptors A-C	**2 (S2,S9)**	**3 (S3-S5)**	**1 (S4)**	**2 (S2,S8)**
Level 0 Descriptor A: Made major errors but did not discover	4 (S3-S6)	4 (S2,S6, S8,S10)	1 (S8)	2 (S4,S6)
Level 0 Descriptor B: Did not make any major error, but never or seldom checked working, reviewed solution and monitored progress	3 (S1,S8,S10)	1 (S1)	7 (S1-S3, S5-S7,S10)	3 (S1,S3,S10)
Level 0: Level 0 Descriptors A and B	**7 (S1,S3-S6,S8,S10)**	**5 (S1,S2, S6,S8,S10)**	**8 (S1-S3,S5-S7,S8,S10)**	**5 (S1,S3, S4,S6,S10)**
Average Score for Level (out of 2)	**0.4**	**0.7**	**0.3**	**0.8**
Standard Deviation (s.d.)	0.7	0.8	0.6	0.9

Table 8.23 Frequencies of Outcomes for Checking

Outcomes	Pretest 1 (Type A)	Posttest 1 (Type A)	Pretest 2 (Type B)	Posttest 2 (Type B)
No. of Major Errors Made	10 (S2-S6,S9)	16 (S2-S6, S8-S10)	5 (S8)	6 (S4,S6,S8,S9)
No. of Minor Errors Made	28 (S1-S7, S9,S10)	23 (S1-S7, S9,S10)	14 (S1-S9)	19 (S1,S2, S4,S6-S9)
Total No. of Errors Made	**38 (S1-S7, S9,S10)**	**39 (S1-S10)**	**19 (S1-S9)**	**25 (S1,S2, S4,S6-S9)**
No. of Major Errors Discovered	3 (S2,S9)	6 (S3-S5,S9)	0	2 (S8,S9)
No. of Minor Errors Discovered	20 (S1-S7,S9)	14 (S1-S7, S9,S10)	10 (S1,S3-S7,S9)	10 (S4,S6-S9)
Total No. of Errors Discovered	**23 (S1-S7,S9)**	**20 (S1-S7, S9,S10)**	**10 (S1,S3-S7,S9)**	**12 (S4,S6-S9)**

Table 8.24 Frequencies of Processes for Checking

Outcomes	Pretest 1 (Type A)	Posttest 1 (Type A)	Pretest 2 (Type B)	Posttest 2 (Type B)
Checked Most Parts of Working	3 (S1,S4)	7 (S1,S3,S4,S10)	4 (S2,S3)	1 (S4)
Checked Some Parts of Working	8 (S3,S4, S9,S10)	7 (S1,S6,S7,S10)	9 (S1,S3-S5, S7,S9,S10)	5 (S7,S9,S10)
Just Glanced Through Working to Check	1 (S9)	7 (S3,S5,S8-S10)	1 (S9)	6 (S2,S5,S10)
Other Ways of Checking Working, e.g. Work Backwards	0	0	1 (S4)	6 (S1,S2, S7,S8,S10)
Total for Checking Working (CW)	**12 (S1,S3, S4,S9,S10)**	**21 (S1,S3-S10)**	**15 (S1-S5, S7,S9,S10)**	**18 (S1,S2,S4, S5,S7-S10)**
Monitored Progress (MP)	31 (S1,S3, S4,S6-S10)	39 (S2,S3, S5,S7-S9)	4 (S4,S6,S9)	10 (S1,S7)
Reviewed Solution (MR)	5 (S1,S4,S5,S9)	4 (S5,S9)	15 (S1,S4, S6,S8,S9)	12 (S1-S5, S9,S10)
Total for CW + MP + MR	**48 (S1,S3-S10)**	**64 (S1-S10)**	**34 (S1-S10)**	**40 (S1-S5, S7-S10)**
Metacognitive Awareness (MA)	4 (S1,S9)	12 (S1,S2,S5-S7,S9,S10)	3 (S5,S7,S10)	2 (S8,S9)

(a) Comparison between Pretest Task 1 and Posttest Task 1 (Type A)

Table 8.22 shows that for Pretest Task 1, one student was at Level 2, 2 students were at Level 1, and 7 students were at Level 0. But for Posttest Task 1, 2 students were at Level 2, 3 students were at Level 1, and 5 students were at Level 0. Thus they had performed slightly better in checking for Posttest Task 1 than for Pretest Task 1. Table 8.23 shows that the total number of errors made by the students for the two tasks was about the same, but the students made more major errors for Posttest Task 1. However, the students had also discovered more major errors for the posttest task, although the total number of errors discovered was slightly more for the pretest task. This explained why they had scored better for Posttest Task 1 as they had discovered more major errors for the posttest task.

Table 8.24 shows that the students had checked working (CW) and monitored progress (MP) more often for Posttest Task 1 than for Pretest Task 1, but the frequency for reviewing the solution to see if it had met the goal of the task (MR) was about the same for both tasks. Thus the total frequency for the processes, CW + MP + MR, was a lot higher for the posttest task. Although a large portion of this frequency belonged to MP, this metacognitive process was not very effective: the students monitored their progress for Posttest Task 1 more often when they were stuck, but this process did not help most of them make any progress (see data analysis in Section 7.2.6). A detailed analysis of the students' protocols and answer scripts for Pretest Task 1 had revealed the same finding. For example, S7 monitored progress the most often at 11 times for Pretest Task 1 and 11 times for Posttest Task 1, but she was still stuck. Table 8.24 also shows that a lot more students exhibited metacognitive awareness (MA) for Posttest Task 1 than for Pretest Task 1. For Posttest Task 1, MA had played a more critical role than all the checking processes in helping the students discover their errors by sensing something amiss (see data analysis in Section 7.2.6). A detailed analysis of the students' protocols and answer scripts for Pretest Task 1 also revealed that MA had helped S1 and S9 to discover their errors, but checking working was not so effective in helping them discover their errors.

(b) Comparison between Pretest Task 2 and Posttest Task 2 (Type B)

Table 8.22 shows that for Pretest Task 2, one student was at Level 2, one student was at Level 1, and 8 students were at Level 0. But for Posttest Task 2, 3 students were at Level 2, 2 students were at Level 1, and 5 students were at Level 0. Thus they had performed better in checking for the posttest task than for the pretest task. Table 8.23 shows that the students had made more errors for Posttest Task 2 than for Pretest Task

2, but the number of major errors made was about the same for both tasks. However, the students had also discovered more major errors for the posttest task, although the total number of minor errors discovered was the same for both tasks. This explained why they had scored better for Posttest Task 2 as they had discovered more major errors for the posttest task.

Table 8.24 shows that the students had checked working (CW) slightly more often for Posttest Task 2 than for Pretest Task 2, but they had monitored progress (MP) more often for the posttest task. However, they reviewed the solution to see if it had met the goal of the task (MR) slightly less often for the posttest task. Overall, the total frequency for the processes, CW + MP + MR, was slightly higher for Posttest Task 2 than for Pretest Task 2. But the total frequencies for the two Type B tasks were low when compared with those for the two Type A tasks. On closer analysis, they seldom monitored their progress for the Type B tasks, unlike the very high frequency of MP for the Type A tasks. This was due to more students being unable to discover any pattern in the Type A tasks, which caused most of them (6 to 8 students) to monitor their progress a lot more frequently, unlike the Type B tasks where most students thought that they could solve the problems that they had posed, which resulted in only a few students (2 to 3 students) monitoring their progress. Just like for the Type A tasks, MP was generally not effective for the Type B tasks because they did not know what else to do. However, they reviewed their solution (MR) more often for the Type B tasks than for the Type A tasks. This was probably due to more students being unable to observe any pattern in the Type A tasks, and so there was no complete solution to review to see if it had met the goal of the task, unlike the Type B tasks where more students had solved their problems.

As for metacognitive awareness (MA), there was not much difference in the frequencies for both tasks. For Posttest Task 2, MA had played a critical role in helping some students sense something amiss, which led them to discover their errors (see data analysis in Section 7.3.10). A detailed analysis of the students' protocols and answer scripts for Pretest Task 2 also revealed that MA had helped S5 to discover an error, and S7 to check her working but the test ended before she could discover her error. Although more students had checked their working more often for Pretest Task 2 than for Posttest Task 2, this process was not critical in helping them discover their errors. For example, out of the 8 students who checked most or some parts of their working for Pretest Task 2 for a total of 4 + 9 = 13 times, only 2 of the checking occasions led to the discovery of 2 minor errors for 2 students (S3,S5).

(c) Development of Checking Processes

The above analysis shows that the students had improved in the process of checking for both the posttest tasks compared with the pretest tasks, although the improvement for the Type A tasks was only a slight one. The main factor that contributed to the discovery of errors was the ability to sense something amiss (MA), which resulted in the students checking only the necessary parts of their working.

In Lesson 6 of the teaching experiment (see Appendix C), the students were reminded that they should check their working occasionally, not just at the end after solving a problem. Since it was too time consuming to check every single step of their working all the time, they were also taught to check their answer by using other means, such as working backwards and examining whether their answer was reasonable or logical, if possible. They were told that they should always review their solution after solving a problem to see if it had met the goal of the task, to evaluate the efficacy of their method of solution, and to look for alternative methods. In Lesson 4 of the teaching experiment, they were taught to monitor their own progress every 5 minutes or so. In lesson 5, I tried to develop in the students the habit to regulate their investigation by monitoring their own progress. As a result, they had engaged in the checking and monitoring processes more often for the two posttest tasks than for the two pretest tasks. In particular, they had engaged in other means of checking their working, such as working backwards for Posttest Task 2. But what was lacking was their ability to sense something amiss (MA) when there was an error. As explained at the end of Section 8.3.3, the teaching experiment did not teach them metacognitive awareness as this process was only discovered after the data collection.

8.4 SUMMARY OF ANSWER TO RESEARCH QUESTION 3

This section will summarise the main findings from analysing the data collected for the present study in order to answer Research Question 3. In general, most students had improved for both posttest tasks, with some students showing better improvement for one type of tasks over the other. Overall, the students had improved slightly more for the Type B tasks than for the Type A tasks, although on average, the students were still at Level 1 for both posttest tasks. The greatest improvement was in the category of understanding the task (U) for the Type A tasks, and in the category of conjecturing (C) for the Type B tasks. At the other end, the least improvement was in the category of problem posing and extension (P/E) for both types of tasks. There was also negligible difference in performance in the category of conjecturing (C) for the Type A tasks.

(a) Understanding the Task

One of the main problems that students faced for the two pretest tasks was the misinterpretation of the tasks due to the lack of engagement in the understanding processes. Although they had learnt during their normal lessons to read textbook exercise questions carefully and to highlight key information before solving, they did not transfer these skills when attempting the two pretest tasks. But after the teaching experiment, they engaged in these processes more often for the two posttest tasks, and in particular, a lot more frequently for Posttest Task 1, thus leading to the greatest improvement in this category of understanding the task (U) for the Type A task.

(b) Problem Posing and Extension

There was not much improvement for problem posing and extension (P/E) for both types of tasks. For Type A tasks, it was understandable since the students just posed the general problem of searching for any pattern. Although they could pose specific problems to solve, there was not enough time in the teaching experiment to develop these processes since the focus was to teach them how to pose specific problems for Type B tasks to solve and to extend to generalise. As a result, all the students actually extended Posttest Task 2, but most of them still did not fully realise that the purpose of extending Type B tasks is to generalise. Thus there was only a slight improvement for the category of P/E for Type B tasks.

(c) Specialising and Using Other Heuristics

There was only a slight improvement for specialising and using other heuristics (S/H) for both types of tasks. Although the students were taught to specialise systematically to search for patterns for Type A tasks during the teaching experiment, they still failed to do so for Posttest Task 1. It seems that they literally took the instruction in the task statement 'choose any number' to mean choosing a number randomly. Thus care must be taken to teach them to specialise systematically even in such situations. For Type B tasks, it was not easy to teach them how to use other heuristics effectively since this was task dependent. But the metacognitive process of analysing the feasibility of the plan to use other heuristics (MF) had helped some students to examine their reasoning more effectively. This suggests that there is a need to develop this important metacognitive process as it was clearly lacking in all the four tasks.

(d) Conjecturing

For the Type B tasks, the greatest improvement was in the category of conjecturing (C), but for the Type A tasks, there was negligible difference in this category. There were three factors to account for the negligible difference for the Type A tasks: (i) the students did not specialise systematically to search for patterns, so many of them were unable to observe the underlying patterns; (ii) most of the tasks used in the teaching experiment were not sequences like the two Type A tasks, so they had not fully learnt how to search for patterns within and across sequences; and (iii) they had difficulty identifying unfamiliar patterns which they had not learnt before, as they failed to recognise that the patterns they had observed were actually the underlying patterns. Thus there is a need to improve in these areas when teaching students the process of conjecturing. On the other hand, they had improved in formulating conjectures for the Type B tasks because of other processes: (i) the ability to reason more effectively, which resulted in finding the optimal cutting method and thus a correct conjecture formulated; and (ii) the habit to extend the task to generalise, which provided more opportunities to formulate other conjectures. This shows how the main processes are linked together and how they could affect one another.

(e) Justifying and Generalising

There was an improvement for justifying and generalising (J/G) for both types of tasks because of three factors: (i) the students had formulated more conjectures for Posttest Task 2 and thus providing more opportunities to justify; (ii) more students had extended Posttest Task 2 to generalise and thus leading to more conjectures that will generalise; and (iii) they had learnt from the teaching experiment that they should prove their conjectures, instead of accepting the conjectures as true without testing or based on naïve testing, which happened very often during the pretest. They had also learnt how to use a non-proof argument to justify their conjectures, but they seldom used a formal proof involving algebra, which was too difficult for them since they were only in Secondary 2.

(f) Checking

There was an improvement for checking (R) for both types of tasks, although the improvement for the Type A tasks was only a slight one. The students had learnt from the teaching experiment to engage in the checking and monitoring processes more often during the posttest, and in particular, some students were able to check their working by using other means, such as working backwards, for Posttest Task 2. But it was found that the main factor that enabled some students to discover their errors was metacognitive awareness (MA): the ability to sense something amiss when they made a mistake. But MA is a newly discovered process in the course of the present study and so there is a need for further research on how to develop this process.

(g) Metacognitive Processes

In general, the students seldom engaged in metacognitive processes for all the four tasks, especially in analysing the feasibility of their goal (MG) and analysing the feasibility of their plan (MF). Although these processes were taught in the teaching experiment, it seems that it was not easy for the students to pick them up. On the other hand, the main bulk of metacognitive processes came from monitoring progress (MP), and the main contributors were 2 students (S7,S8) who monitored their progress from 9 to 11 times for the two Type A tasks and for Posttest Task 2. However, most of the MP were not effective as they did not know what else to do. Therefore, there is a need to research further into how to develop in students the habit to perform MG and MF more frequently, and how to monitor their progress more effectively.

8.5 CONCLUDING REMARKS FOR THIS CHAPTER

Chapter 8 has answered Research Question 3. The students generally performed better for the posttest than for the pretest for both types of tasks. The analysis has identified which processes the students had developed well, and which processes the students needed more time to develop. Although the teaching experiment was effective in developing the investigation processes to a certain extent, there were some areas that could be improved on. Chapter 9 will present the main contributions for the present study and conclude with some implications for teaching and research.

CHAPTER 9: CONCLUSION AND IMPLICATIONS

This chapter will present the main contributions of the present research, delineate the limitations of the current study, and draw some implications for teaching and further research.

9.1 MAIN CONTRIBUTIONS OF PRESENT RESEARCH

The main contributions of the present study are based mostly on the findings of the three research questions in Chapters 6-8. A summary of the answer to each research question has been given at the end of each of the previous three chapters.

9.1.1 Relationship between Investigation and Problem Solving

Although many educators (e.g. Evans, 1987; Pirie, 1987) agree that there is an overlap between investigation and problem solving, these same educators still end up separating them into two distinct processes: investigation must involve (open) investigative tasks while problem solving must involve (closed) mathematical problems. Another conflicting view in current literature is that some researchers (e.g. Orton & Frobisher, 1996) believed that mathematical problems include investigative tasks while others (e.g. Cai & Cifarelli, 2005) believed that investigation includes problem solving. To resolve these issues, I realised that the problem lies in the different usages of the term ‘investigation’, and so I decided to separate investigation as a task, a process and an activity (see Sections 2.1.2 and 2.1.3).

With this alternative viewpoint, the relationship between investigation and problem solving becomes clearer. Investigation, as an activity, involves the use of an (open) investigative task, and it includes both problem posing and problem solving. But there are generally two approaches to problem solving (Pólya, 1957; Lakatos, 1976): the inductive approach that involves specialising, conjecturing, justifying and generalising (which is the investigation process), and the deductive approach that uses other heuristics such as reasoning. The clarification of the relationship between investigation and problem solving is not just academic. Since it has been identified which processes in investigation are similar to those in problem solving, students can apply the same processes in both investigation and problem solving. Research in these problem-solving processes can also be applied to research in investigation, especially when there is a lack of research in investigation processes.

9.1.2 Mathematical Investigation Models, Coding Scheme and Scoring Rubric

Based on the literature review in Chapter 2 and some modifications on my part, two theoretical mathematical investigation models have been developed for the current research to display the investigation pathways or interactions among the processes: one model for cognitive processes and the other model for metacognitive processes (see Section 3.2). These models were then validated and refined based on the empirical data obtained from the posttest of the present study (see Section 7.4.2).

An important finding during the validation is that the process of conjecturing is actually much more complicated than posited in the theoretical model for cognitive processes, where students are supposed to treat their observed patterns as conjectures to be proven or refuted. Although this was confirmed by some students who did just that, there were some other students who would try further examples to be more certain of their observed pattern before treating it as a conjecture. This means that testing using examples could occur at two levels: (i) between observing a pattern and formulating it as a conjecture, and (ii) after formulating it as a conjecture before proving it (see Section 7.2.4a). This is consistent with Frobisher's (1994) model where he differentiated between 'conjecturing' and 'hypothesising' with two levels of testing using empirical data (see Section 2.2.2g on page 59), although most educators (e.g. Lampert, 1990; Mason et al., 1985) do not distinguish between them. In other words, 'observing a pattern' and 'formulating a conjecture' in the refined model for the present study would correspond to 'conjecturing' and 'hypothesising' in Frobisher's model respectively. This distinction is not just an academic issue because it would affect how teachers teach their students, e.g. teachers should teach their students to be more certain of an observed pattern by testing using further examples, instead of hastily accepting it as a conjecture and then try to prove it, when the pattern could have easily been refuted by counter examples.

Another important finding is metacognitive awareness, or knowledge of cognition, which was usually portrayed in most literature as something that was more passive than active (e.g. Desoete & Veenman, 2006; Schraw & Moshman, 1995). That was why most research studies on knowledge of cognition (e.g. Mevarech & Fridkin, 2006; Wong, 1989) used a paper-and-pencil test to try to understand the knowledge in the minds of the subjects. However, it was found that some students in the present research exhibited a more active awareness of their cognition, rather than a passive knowledge of cognition. This metacognitive awareness had proven to be useful to help some students discover their mistakes when they sensed something amiss, or to save time by being aware that they had generated a sequence or tried a cutting method before, and so they should not redo the investigation. The refined investigation model on metacognitive processes shows that this metacognitive process had manifested itself during specialising, using other heuristics and conjecturing during the investigation of the two posttest tasks.

The Final Coding Scheme was devised in Chapter 4 to code the students' cognitive and metacognitive behaviours in their thinking-aloud protocols in order to inform the two investigation models, and the Investigation Scoring Rubric was formulated in Chapter 8 to evaluate the students' proficiency in investigation based on the quality of their processes and outcomes. Since there are few empirical studies on investigation processes, the two Refined Investigation Models, the Final Coding Scheme and the Investigation Scoring Rubric developed in the present research could be useful to help other researchers and educators to study the nature and development of investigation processes in other setting.

9.1.3 Effect of Processes on Outcomes in Mathematical Investigation

The study has revealed that re-reading or rephrasing the *relevant* parts of the task statement has helped some students to understand the task correctly or to recover from any misinterpretation. But referring to the task statement does not seem to help some students to pose specific problems for Type B tasks. In fact, the students still had great difficulty posing the intended problem and the intended extension for Posttest Task 2 (Sausage) despite going through the teaching experiment. The data analysis has suggested that the ability to pose the intended problem or extension depends on the particular task and how the task statement is phrased (see Section 8.3.2b). Since a

Type B investigative task is obtained from a mathematical problem by removing the original intended problem and replacing it with the word ‘investigate’ (Frobisher, 1994), the implication of this finding is that the teachers ought to be aware that there is a possibility of ‘losing’ the intended problem or extension that they might want their students to solve. Therefore, there is a need for teachers to think about how to guide their students to pose the intended problem or extension for a Type B task, and yet not close up the task by restricting the students’ freedom and creativity to pose other types of problems and extensions to solve.

It was found that most of the students specialised randomly for the two Type A tasks instead of systematically as advocated by Mason et al. (1985). It seems that the students literally took the instruction in both task statements ‘choose any number’ to mean choosing a number randomly. Thus care must be taken to teach students to specialise systematically even in such situations. The present study also found that most of the students wrongly accepted their conjectures as true without testing or based on naïve testing (Lakatos, 1976) during the pretest, but this had improved for the posttest where more students tried to prove their conjectures. However, most of the students used a non-proof argument (Stylianides, 2008), instead of a formal proof, to justify their conjectures. Very few students attempted a formal proof involving algebra, which also did not work out. Although some educators (e.g. Holding, 1991; Tall, 1991) preferred the rigour found in formal proofs, it seems that formal proofs might be beyond the abilities of these Secondary 2 students.

Another finding is that the students were unable to observe patterns that were not the usual types of patterns that they had learnt before. For example, consecutive terms of a Kaprekar sequence (Posttest Task 1) are not in a fixed increasing order, i.e. the next term is not obtained from the previous term by adding a constant, and so there is no relationship between the term T_n and its position n (see Section 7.2.4c on page 301). This means that it is not possible to find a formula for the n-th term in terms of its position n, unlike the sequences that the students had learnt in their normal school lessons where they could usually find a formula for the general term, e.g. $T_n = 3n$ for the multiples of 3 sequence: 3, 6, 9, … For the latter sequence, the teacher would not accept patterns like 'the sequence is increasing' and 'all the terms are divisible by 3' because these patterns are clearly true. As a result, the students failed to realise that 'the sequence is increasing' and 'all the terms are divisible by 3 if the starting term is divisible by 3' could be treated as patterns for Kaprekar sequences because they might not be true. However, two students were able to actively apply their knowledge, or what Schoenfeld (1985) called 'resources', to discover the Type 1a 'multiples' pattern or the complicated Type 2 'digital roots' pattern. Another student was also able to discover the complex general formula for the least number of cuts for Posttest Task 2 (Sausage) just by searching for patterns from two random examples.

The study also found that most of the students lacked metacognitive skills. Those who could progress in their investigation seldom stopped to monitor their progress, while those, who monitored their progress often, did so because they were stuck. However, this metacognitive behaviour was not so effective since the students did not know what else to do, other than to continue in the same approach. The students could have analysed the feasibility of their plan to solve the problem before blindly continuing in the same ineffective approach, but most of them did not do so. Only one student analysed the feasibility of her goal, but it was also not effective. This probably explains why some of the students ended up pursuing very trivial or very difficult problems because they did not pause to analyse whether these problems were feasible.

9.1.4 Development of Processes in Mathematical Investigation

In general, the cognitive processes that had developed rather well during the teaching experiment were the main processes of understanding the task and justifying. During the pretest, many students did not spend much time trying to understand the task properly, so quite a number of them misinterpreted the task. Since they had learnt the usual processes for understanding textbook exercise questions during their normal school lessons, it was naturally very easy for them to pick these up for understanding investigative tasks. But the point is that teachers should not assume that students would automatically apply these processes for understanding textbook exercise questions to understanding other types of tasks, unless they are reminded to do so. As for justifying, some students had learnt during the teaching experiment that they should not accept a conjecture as true without testing or based on naïve testing (Lakatos, 1976), which partly accounted for the improvement in the justifying process. The other reason was that the students had formulated more conjectures during the posttest, thus providing more opportunities to justify.

The cognitive processes that did not develop so well during the teaching experiment were the main processes of problem posing and extension. Although all the students had learnt how to extend the Type B task during the posttest, quite a number of them still did not fully understand that the purpose of such extension was to generalise, so they just changed the given variables *per se* without the intention to generalise. Therefore, there was a need to improve in this aspect of developing in students the habit to generalise whenever possible because "generalisations are the life-blood of mathematics" (Mason et al., 1985, p. 8). There was also a need to teach students how to pose non-trivial specific problems for Type A tasks, especially the sub-process of analogy (Kilpatrick, 1987) in using a previous result as a springboard to pose more problems to solve or extend. Although this sub-process was taught in the teaching experiment, albeit briefly due to the lack of time, the study has found that only very few students were able to pose problems using analogy.

However, the development of the main process of conjecturing was mixed. There was negligible difference for conjecturing between the two Type A tasks in the pretest and the posttest, but there was a great improvement for the Type B task in the posttest. The students did not specialise systematically to search for patterns within and across sequences for the Type A posttest task, and they had difficulty identifying unfamiliar patterns which they had not learnt before. But for the Type B posttest task, the ability to formulate more correct non-trivial conjectures was partly due to the ability to use reasoning more effectively in the main process of using other heuristics, and partly due to the habit to extend the task to generalise which provided more opportunities to formulate other conjectures. This suggests how the main processes are intricately linked together and how they could affect one another.

Despite teaching students various metacognitive processes, it was found that most of these processes were not easily picked up by the students, except for monitoring of progress although it was not so effective. Thus there was a need to look into how to develop in students the habit to engage in these metacognitive processes, especially in analysing the feasibility of their goal and the feasibility of their plan, since these two processes were seldom engaged by the students in both the pretest and the posttest.

9.2 LIMITATIONS OF PRESENT STUDY

The following are some limitations of the current study.

- The data analysis in this study was mainly qualitative. Although steps have been taken to improve the validity and reliability of the present research, such as inter-coder reliability tests for the coding scheme and the classification of the quality of certain investigation outcomes as trivial or non-trivial, interpreting behaviours is still very much a human endeavour which could never be 100% accurate in such studies.

- Although the thinking-aloud method is found most suitable to track students' actual thinking processes and the students in this study had been trained and given some practice in thinking aloud to alleviate some of the shortfalls, some students might still be uncomfortable in verbalising their thoughts, especially when they were videotaped doing it. The time available for the students to practise thinking aloud was also limited by what the school could spare for this research. As a result, some students kept silent during the pretest and posttest occasionally, so it could not be inferred what they were thinking during this time. Therefore, valuable insight into the students' minds at critical periods might be lost. On the other hand, thinking aloud might also interfere with the students' investigation and thus affect the outcomes.

- Due to the time that the school could spare for the present study, the duration of the teaching experiment was rather short, consisting of only six lessons lasting two hours each, and the duration of the pretest and the posttest was only one hour each, with 30 minutes for each task. As a result, there was not enough time to develop each process more fully. Moreover, the short duration to investigate each task during the test means that the students might not have time to pose more problems to solve or to observe more patterns.

- An in-depth study with a small sample size and a small number of tasks for the pretest and the posttest mean that the findings for the present study have limited generalisability. A bigger sample size with more investigative tasks of other varieties for the tests would provide more insights into the nature of the students' investigation processes.

9.3 IMPLICATIONS AND RECOMMENDATIONS FOR TEACHING AND RESEARCH

The purpose of the present study is to inform teachers and researchers on the nature and development of processes in mathematical investigation. Based on the findings and limitations of the present study, some recommendations for teaching and research are outlined below.

9.3.1 Implications and Recommendations for Teaching

An implication of the findings in the present study for teaching is that teachers could teach their students to use the investigation process in solving mathematical problems since these processes (namely, specialising, conjecturing, justifying and generalising) are similar in both investigation and problem solving. Using the Handshake Problem (Task 4) in Section 2.1.2(d) as an example, if students do not know how to solve it by using other heuristics such as reasoning, they could be taught to investigate by trying smaller numbers of participants (specialising) in order to search for patterns (conjecturing) in order to generalise. But the students need to learn that the observed pattern is just a conjecture that needs to be justified.

Another implication is how teachers could develop in their students the processes in mathematical investigation, based on the understanding of the nature of investigation processes as depicted in the two investigation models and the development of such processes in the teaching experiment. The following are some suggestions for teachers to consider when teaching students how to do mathematical investigation using open investigative tasks.

- Trying some examples to make sense of Type A investigative tasks (Mason et al, 1985) is a new skill that should be taught because most students would not have learnt it during normal school lessons. But there is also a need to remind students to apply the usual processes for understanding textbook exercise questions which they have learnt, such as reading the question carefully, to understand an investigative task properly. This is important because if students misinterpret the task, they would end up on a wild goose chase (Schoenfeld, 1987), which was exactly what happened to most students in the present study who did not spend much time understanding the task during the pretest.

- Students need to be taught what to investigate for an investigative task that contains no stated problem. For Type A tasks, after teaching and some exposure, the students generally know that they were supposed to search for any pattern. But for Type B tasks, it is not easy to teach students how to pose the intended problem as suggested by Frobisher (1994). One way is to discuss in class after each student has spent some time thinking about it, so that collectively, there is a higher chance that someone might pose the intended problem. But it does not really matter if no one poses the intended problem if some of them are able to pose other 'unexpected' problems that are interesting (Brown & Walter, 2005). On one hand, students should be encouraged to pose difficult problems that they might not be able to solve (Mason et al, 1985). On the other hand, teachers should emphasise that it does not mean that they have to solve them (because some students in the present study wasted the entire investigation pursuing such unachievable goals). Therefore, the students need to be taught how to analyse which problems are feasible or worth pursuing, a skill that was clearly lacking even in the posttest of the present study.

- Teachers should teach their students to specialise systematically to search for patterns (Mason et al, 1985), but there is a need to let the students attempt investigative tasks that tell them to 'choose any number', so as to emphasise to them that this does not mean that they should choose numbers randomly to specialise. Students should be exposed to more unusual patterns that they have not come across before because most of the students in the present study had difficulty recognising these unfamiliar patterns as 'the patterns' even when they had observed them. They should be taught to be more metacognitively aware of what they are investigating, and not specialise or search for patterns blindly. For example, they should be aware of the previous examples that they have tried before so that they will not redo the same examples, or they should use some reasoning to explain why the differences between consecutive numbers in a Kaprekar sequence will never repeat so that they will not waste time searching for a pattern that will never exist. This also includes teaching students to analyse the feasibility of their plan of attack (Schoenfeld, 1987).

- There is a need to use a concrete example to demonstrate to students that an observed pattern is actually not the underlying pattern (see Task 9 on Chords and Regions in Appendix D) in order to convince them that they should always test their conjectures. Although most of the students had learnt to do so in the posttest, some of them still accepted their conjectures as true without testing or based on naïve testing. This means that teachers should spend more time developing in their students the habit to justify conjectures. In particular, there is a need to teach students that some results obtained from using other heuristics (i.e. not an observed pattern) are also conjectures, e.g. just because a method gives a shorter toasting time or a fewer number of cuts than all the other methods does not mean that this method will give the least toasting time or the least number of cuts (this was a mistake that some students still made during the posttest as it was not emphasised during the teaching experiment).

- On the other hand, teachers should teach their students not to hastily accept an observed pattern as a conjecture and then try to justify it. Instead, the students should learn to verify that the observed pattern could at least withstand the weight of a few more examples because there are occasions in the present study when the students could have easily rejected the observed patterns based on some counter examples but they spent so much time trying to think of a proof. At Secondary 2 level, teachers should probably focus on teaching students to use a non-proof argument to justify their conjectures (Mason et al., 1985; Stylianides, 2008) because the students in the present study, who are from a high-performing Singapore school, are not able to cope with formal proofs. At higher level, teachers could then consider developing in students the ability to use algebra as a formal proof (Holding, 1991; Tall, 1991).

- Since it is not feasible to check working at every step, teachers should teach their students to check essential working occasionally because some students in the present study could go through the entire investigation without checking any working when they had actually made some mistakes. The students should also be taught to review their solution to see if it had met the goal of the investigative task (Jacobs & Paris, 1987; Schraw, 2001), e.g. if the students are to ask themselves whether they have solved the problem of finding a cutting method that will give the least number of cuts, it might make them realise that they need to prove that the number of cuts is actually the least.

- Although students need to be taught to extend Type B tasks by changing the given, the teachers should emphasise that the purpose of such extension is to generalise (Mason et al., 1985) since quite a number of students still changed the given in the posttest without trying to generalise. For Type A tasks, the students should be taught to pose more specific problems by using analogy (Kilpatrick, 1987), especially by using previous results as a springboard to pose more problems to solve, with or without extending the task.

9.3.2 Implications and Recommendations for Research

The following are some suggestions for further research on investigation using open investigative tasks.

- More research needs to be done to study the development of some of the investigation processes because they were clearly lacking in the students even after the teaching experiment, e.g. how to teach students to pose the intended problem or the intended extension, how to teach students to monitor progress effectively, and how to develop metacognitive awareness among the students, especially the ability to sense something amiss when a mistake occurs.

- On the other hand, some of the processes might not have developed because they might take a longer time. Therefore, a suggestion for further research is to conduct a much longer teaching experiment to study whether the students are able to develop more of these processes, e.g. analysing the feasibility of the goal or the plan, and the habit to generalise whenever possible.

- Average and low-achieving students might approach mathematical investigation differently from the high-achieving students in the present study. Similarly, students at different age groups might engage in the investigation processes differently from the Secondary 2 students in this research. Thus a suggestion for further research is to study the similarities and differences in the investigation processes across students of different age groups and with academic achievements.

- Since it was found that formal proofs involving algebra were beyond the high-achieving Secondary 2 students in this study, a suggestion for further research is to study the nature and development of formal proofs in mathematical investigation by involving high-achieving Upper Secondary students.

- Tanner (1989) observed that group work and discussion during investigation had helped the students to generate and test ideas, and to practise the communication of ideas which would force them to clarify and redefine their ideas if necessary. Thus a suggestion for further research is to study the effect of group work or pair work on the development of investigation processes.

- Research has suggested that affect, which includes beliefs and attitudes, plays an important role in a student's learning. Many studies have found that a positive correlation between attitude and achievement in mathematics (McLeod, 1992). Schoenfeld (1985) has also conducted some studies to suggest that students' beliefs could also influence their problem-solving behaviour. Thus a suggestion for further research is to examine the effect of students' affective variables on the investigation processes and outcomes.

9.4 EPILOGUE

Although mathematical investigation is important in many school curricula and there are a lot of resources on investigation, there are very few empirical studies on the thinking processes that students engage in when they attempt open investigative tasks. Therefore, the present study has contributed to current research on investigation by opening a window into the 10 students' minds to glimpse the nature of their cognitive and metacognitive processes during mathematical investigation, and by suggesting how some of these processes could be developed in the classroom.

REFERENCES

Ainley, J. (1988). Playing games and real mathematics. In D. Pimm (Ed.), *Mathematics, teachers and children* (pp. 239-248). London: Hodder and Stoughton.

Ainley, J. (1990). Playing games and learning mathematics. In L. P. Steffe & T. Wood (Eds.), *Transforming children's mathematics education: International perspectives* (pp. 84-91). Hillsdale, NJ: Erlbaum.

Artzt, A. F., & Armour-Thomas, E. (1992). Development of a cognitive-metacognitive framework for protocol analysis of mathematical problem solving in small groups. *Cognition and Instruction, 9*, 137-175.

Askey, R. (1999). The third mathematics education revolution. In E. A. Gavosto, S. G. Krantz, & W. McCallum (Eds.), *Contemporary issues in mathematics education* (pp. 95-107). Australia: Cambridge University Press.

Australian Education Council. (1991). *A national statement on mathematics for Australian schools.* Melbourne: Curriculum Corporation.

Backhouse, J., Haggarty, L., Pirie, S., & Stratton, J. (1992). *Improving the learning of mathematics.* London: Cassell.

Bailey, J. (2007). Mathematical investigations: A primary teacher educator's narrative journey of professional awareness. In J. Watson & K. Beswick (Eds.), *Proceedings of the 30th annual conference of the Mathematics Education Research Group of Australasia (MERGA): Mathematics: Essential research, essential practice* (Vol. 1, pp. 103-112). Adelaide, South Australia: MERGA.

Bastow, B., Hughes, J., Kissane, B., & Mortlock, R. (1991). *40 mathematical investigations* (2nd ed.). Western Australia: Mathematical Association of Western Australia.

Becker, J. P., & Shimada, S. (1997). *The open-ended approach: A new proposal for teaching mathematics.* Reston, VA: National Council of Teachers of Mathematics.

Begle, E. G. (1979). *Critical variables in mathematics education*. Washington, DC: Mathematical Association of America and National Council of Teachers of Mathematics.

Biddle, A., Savage, S., Smith, T., & Vowles, L. (1988). *Mathematics for investigative coursework.* West Sussex, England: The Mathematics Centre, West Sussex Institute of Higher Education.

Bird, M. (1980). Using noughts and crosses as a starting point. *Mathematics Teaching, 90*, 12-15.

Bishop, A. (1982). Teachers and research: The ATM connection. *Mathematics Teaching, 100*, 62-66.

Bloom, B. S., Engelhart, M. D., Furst, E. J., Hill, W. H., & Krathwohl, D. R. (1956). *Taxonomy of educational objectives: The classification of educational goals. Handbook I. Cognitive domain.* New York: David McKay.

Boaler, J. (1998). Open and closed mathematics: Student experiences and understandings. *Journal for Research in Mathematics Education, 29*, 41-62.

Brenner, M. E., & Moschkovich, J. N. (Eds.). (2002). *Everyday and academic mathematics in the classroom.* Reston, VA: National Council of Teachers of Mathematics.

Brown, A. (1987). Metacognition, executive control, self-regulation and other more mysterious mechanisms. In F. E. Weinert & R. H. Kluwe (Eds.), *Metacognition, motivation and understanding* (pp. 65-116). Hillsdale, NJ: Erlbaum.

Brown, A. L., Bransford, J., Ferrara, R., & Campione, J. (1983). Learning, remembering and understanding. In P. H. Mussen (Ed.), *Handbook of child psychology* (Vol. 3). New York: Wiley.

Brown, S. I. (1996). Posing mathematically: A novelette. In T. J. Cooney, S. I. Brown, J. A. Dossey, G. Schrage, & E. Ch. Wittmann (Eds.), *Mathematics, pedagogy, and secondary teacher education* (pp. 281-370). Portsmouth, USA: Heinemann.

Brown, S. I., & Walter, M. I. (2005). *The art of problem posing* (3rd ed.). Mahwah, NJ: Erlbaum.

Bruner, J. S. (1961). The act of discovery. *Harvard Educational Review, 31*, 21-32.

Burns, R. B. (2000). *Introduction to research methods* (4th ed.). NSW, Australia: Longman.

Burton, L. (1984). *Thinking things through: Problem solving in mathematics.* Oxford, UK: Basil Blackwell.

Cai, J., & Cifarelli, V. (2005). Exploring mathematical exploration: How two college students formulated and solve their own mathematical problems. *Focus on Learning Problems in Mathematics, 27*(3), 43-72.

Cai, J., Mamona-Downs, J., & Weber, K. (2005). Mathematical problem solving: What we know and where we are going. *Mathematical Behavior, 24*, 217-220.

Cambridge University Press (2012). *Cambridge Dictionaries Online.* Retrieved February 29, 2012, from http://dictionary.cambridge.org

Carraher, D. W., & Schliemann, A. D. (2002). Is everyday mathematics truly relevant to mathematics education? In E. Yackel (Series Ed.) & M. E. Brenner & J. N. Moschkovich (Monograph Eds.), *Everyday and academic mathematics in the classroom* (pp. 131-153). Reston, VA: National Council of Teachers of Mathematics.

Cederberg, J. N. (2001). *A course in modern geometries* (2nd ed.). New York: Springer.

Chong, T. H, Khoo, P. S., Foong, P. Y., Kaur, B., & Lim-Teo, S. K. (1991). *A state-of-the-art review on mathematics education in Singapore.* Singapore: Institute of Education.

Chow, I. Y. P. (2004). *Impact of open-ended problem solving as an alternative assessment on secondary one mathematics students.* Unpublished master's thesis, National Institute of Education, Nanyang Technological University, Singapore.

Christiansen, B., & Walther, G. (1986). Task and activity. In B. Christiansen, A. G. Howson, & M. Otte (Eds.), *Perspectives on mathematics education: Papers submitted by members of the Bacomet Group* (pp. 243-307). Dordrecht, The Netherlands: Reidel.

Cifarelli, V., & Cai, J. (2004, July). *A framework for examining the mathematical exploration of problem solvers.* Research paper presented to the Special Interest Group on Problem Solving of the International Congress on Mathematics Education, Copenhagen, Denmark.

Civil, M. (2002). Everyday mathematics, mathematicians' mathematics, and school mathematics: Can we bring them together? In E. Yackel (Series Ed.) & M. E. Brenner & J. N. Moschkovich (Monograph Eds.), *Everyday and academic mathematics in the classroom* (pp. 40-62). Reston, VA: National Council of Teachers of Mathematics.

Clement, J. (2000). Analysis of clinical interviews: Foundations and model viability. In A. E. Kelly & R. A. Lesh (Eds.), *Handbook of research design in mathematics and science education* (pp. 547-589). Mahwah, NJ: Erlbaum.

Cobb, P. (1991). Reconstructing elementary school mathematics. *Focus on Learning Problems in Mathematics, 13*(2), 3-32.

Cockcroft, W. H. (1982). *Mathematics counts: Report of the committee of inquiry into the teaching of mathematics in schools under the chairmanship of Dr W H Cockcroft.* London: Her Majesty's Stationery Office (HMSO).

Collins, A. (1988). Different goals of inquiry teaching. *Questioning Exchange, 2*(1), 39-45.

Davies, B. (1980). Using calculators with juniors. *Mathematics Teaching, 93*, 12-15.

Davis, P. J., & Hersh, R. (1981). *The mathematical experience.* Brighton, Sussex: Harvester Press.

Delaney, K. (1996). Exploring difficulties in teaching mathematics through investigations in the primary classroom. *For the Learning of Mathematics, 16*(1), 27-33.

DeMarois, P., McGowen, M., & Whitkanack, D. (1996). *Mathematical investigations: An introduction to algebraic thinking.* New York: HarperCollins College.

Desoete, A., & Veenman, M. (2006). Metacognition in mathematics: Critical issues on nature, theory, assessment and treatment. In A. Desoete & M. Veenman (Eds.), *Metacognition in mathematics education* (pp. 1-10). New York: Nova Science.

Dindyal, J., Tay, E. G., Toh, T. L., Leong, Y. H., & Quek, K. S. (2012). Mathematical problem solving for everyone: A new beginning. *The Mathematics Educator, 13*(2), 1-20.

Doyle, W. (1983). Academic work. *Review of Educational Research, 53*, 159-199.

English, L. (2007). Interdisciplinary Modelling in the Primary Mathematics Curriculum. In J. Watson & K. Beswick (Eds.), *Proceedings of the 30th annual conference of the Mathematics Education Research Group of Australasia (MERGA): Mathematics: Essential research, essential practice* (Vol. 1, pp. 275-284). Adelaide, South Australia: MERGA.

English, L. D., Jones, G. A., Lesh, R. A., Tirosh, D., & Bussi, M. B. (2002). Future issues and directions in international mathematics education research. In L. D. English (Ed.), *Handbook of international research in mathematics education* (pp. 787-806). Mahwah, NJ: Erlbaum.

Ericsson, K. A., & Simon, H. A. (1980). Verbal reports as data. *Psychological Review, 87*(3), 215-251.

Ericsson, K. A., & Simon, H. A. (1993). *Protocol analysis: Verbal reports as data* (Rev. ed.). Cambridge, MA: MIT Press.

Ernest, P. (1991). *The philosophy of mathematics education.* London: Falmer Press.

Euclid (1956). *The thirteen books of Euclid's Elements.* New York: Dover.

Evans, J. (1987). Investigations: The state of the art. *Mathematics in School, 16*(1), 27-30.

Fabian, J. (1990). *Creative thinking and problem solving.* Chelsea, MI: Lewis.

Fan, L., & Zhu, Y. (2000). Problem solving in Singaporean secondary mathematics textbooks. *The Mathematics Educator, 5*(2), 117-141.

Flavell, J. (1976). Metacognitive aspects of problem solving. In L. Resnick (Ed.), *The nature of intelligence* (pp. 231-236). Hillsdale: Erlbaum.

Flavell, J. H. (1987). Speculations about the nature and development of metacognition. In F. E. Weinert & R. H. Kluwe (Eds.), *Metacognition, motivation and understanding* (pp. 1-29). Hillsdale, NJ: Erlbaum.

Focant, J., Grégoire, J., & Desoete, A. (2006). Goal-setting, planning and control strategies and arithmetical problem solving at Grade 5. In A. Desoete & M. Veenman (Eds.), *Metacognition in mathematics education* (pp. 51-71). New York: Nova Science.

Foong, P. Y. (1990). *A metacognitive-heuristic approach to mathematical problem solving.* Unpublished doctoral dissertation, Monash University, Victoria, Australia.

Foong, P. Y. (2007). Teacher as researcher: A review of mathematics education research of Singapore teachers. *The Mathematics Educator, 10*(1), 3-20.

Foong, P. Y. (2009). Review of research on mathematical problem solving in Singapore. In Niss, M., P. Y., Lee & Kilpatrick, J. (Series Eds.), & K. Y. Wong, P. Y. Lee, B. Kaur, P. Y. Foong & S. F. Ng (Volume Eds.), *Mathematics Education: The Singapore Journey* (pp. 263-300). Singapore: World Scientific.

Frobisher, L. (1994). Problems, investigations and an investigative approach. In A. Orton & G. Wain (Eds.), *Issues in teaching mathematics* (pp. 150-173). London: Cassell.

Garofalo, J., & Lester, F. K., Jr. (1985). Metacognition, cognitive monitoring, and mathematical performance. *Journal for Research in Mathematics Education, 16*, 163-176.

Goldin, G. A. (2000). A scientific perspective on structured, task-based interviews in mathematics education research. In A. E. Kelly & R. A. Lesh (Eds.), *Handbook of research design in mathematics and science education* (pp. 517-545). Mahwah, NJ: Erlbaum.

Goldin, G. A. (2002). Representation in mathematical learning and problem solving. In L. D. English (Ed.), *Handbook of international research in mathematics education* (pp. 197-218). Mahwah, NJ: Erlbaum.

Hatch, G. (2002). Maximising energy in the learning of mathematics. In L. Haggarty (Ed.), *Teaching mathematics in secondary schools* (pp. 115-128). London: RoutledgeFalmer.

Hawera, N. (2006). Maori preservice primary teachers' responses to mathematical investigations. In P. Grootenboer, R. Zevenbergen, & M. Chinnappan (Eds.), *Proceedings of the 29th annual conference of the Mathematics Education Research Group of Australasia: Identities, cultures and learning spaces* (Vol. 1, pp. 286-292).

Height, T. P. (1989). *Mathematical investigations in the classroom.* Australia: Longman Cheshire Pty.

Henderson, K. B., & Pingry, R. E. (1953). Problem solving in mathematics. In H. F. Fehr (Ed.), *The learning of mathematics: Its theory and practice* (pp. 228-270). Washington, DC: National Council of Teachers of Mathematics.

Henningsen, M., & Stein, M. K. (1997). Mathematical tasks and student cognition: Classroom-based factors that support and inhibit high-level mathematical thinking and reasoning. *Journal for Research in Mathematics Education, 28*, 524-549.

Hiebert, J., & Wearne, D. (1993). Instructional tasks, classroom discourse, and students' learning in second-grade arithmetic. *American Educational Research Journal, 30*, 393-425.

Ho, F. H., & Soon, S. (1998). *Patterns and structures.* Singapore: Greenfield.

Holding, J. (1991). *The investigations book: A resource book for teachers of mathematics.* Cambridge, UK: Cambridge University Press.

Howson, A. G. (1983). *A review of research in mathematics education: Part C. Curricular development and curricular research. A historical and comparative view.* Windsor, Berks: NFER-Nelson.

Howson, G., Keitel, C., & Kilpatrick, J. (1981). *Curriculum development in mathematics.* Cambridge: Cambridge University Press.

Jacobs, J. E., & Paris, S. G. (1987). Children's metacognition and reading: Issues in definition, measurement and instruction. *Educational Psychologist, 22*, 255-278.

Jaworski, B. (1994). *Investigating mathematics teaching: A constructivist enquiry.* London: Falmer Press.

Kaiser, G., & Sriraman, B. (2006). A global survey of international perspectives on modelling in mathematics education. *Zentralblatt für Didaktik der Mathematik, 38*, 302-310.

Kaur, B. (1995). *An investigation of children's knowledge and strategies in mathematical problem solving.* Unpublished doctoral dissertation, Faculty of Education, Monash University, Australia.

Keiichi, S. (2000). Metacognition in mathematics education. In O. Toshio (Ed.), *Mathematics education in Japan* (pp. 65-68). Japan: Japan Society of Mathematics Education.

Kelly, A. E., & Lesh, R. A. (Eds.). (2000a). *Handbook of research design in mathematics and science education.* Mahwah, NJ: Erlbaum.

Kelly, A. E., & Lesh, R. A. (2000b). Trends and shifts in research methods. In A. E. Kelly & R. A. Lesh (Eds.), *Handbook of research design in mathematics and science education* (pp. 35-44). Mahwah, NJ: Erlbaum.

Kilpatrick, J. (1968). Analyzing the solution of word problems in mathematics: An exploratory study (Doctoral dissertation, Stanford University, 1967). *Dissertation Abstracts International, 28*, 4380A.

Kilpatrick, J. (1987). Problem formulating: Where do good problems come from? In A. H. Schoenfeld (Ed.), *Cognitive science and mathematics education* (pp. 123-147). Hillsdale, NJ: Erlbaum.

Krulik, S. (Ed.). (1980). *Problem solving in school mathematics: 1980 Yearbook* (R. E. Reys, General Yearbook Ed.). Reston, VA: NCTM.

Krulik, S., & Rudnick, J. A. (1993). *Reasoning and problem solving: A handbook for elementary school teachers.* Boston: Allyn and Bacon.

Krutetskii, V. A. (1976). The psychology of mathematical abilities in school children. In J. Kilpatrick & I. Wirszup (Series Eds.), *Soviet studies in the psychology of learning and teaching mathematics.* Chicago: University of Chicago Press.

Lakatos, I. (1976). *Proofs and refutations: The logic of mathematical discovery.* New York: Cambridge University Press.

Lampert, M. (1990). When the problem is not the question and the solution is not the answer: Mathematical knowing and teaching. *American Educational Research Journal, 27*, 29-63.

Lampert, M., & Ball, D. L. (1998). *Teaching, multimedia, and mathematics: Investigations of real practice.* New York: Teachers College Press.

Lee, M., & Miller, M. (1997). *Real-life math investigations: 30 activities that help students apply mathematical thinking to real-life situations.* New York: Scholastic Professional Books.

Lee, P. Y., & Fan, L. (2004). The development of Singapore mathematics curriculum: Understanding the changes in syllabus, textbook and approaches. In P. Y. Lee, D. Zhang, & N. Song (Eds.), *Proceedings of the International Congress of Mathematicians (ICM 2002) Satellite Conference on the Reform of Mathematics Curriculum and its Education in the 21st Century* (pp. 31-36). Chongqing, China: Chongqing Publishing House.

Lenchner, G. (1983). *Creative problem solving in school mathematics.* New York: Glenwood.

Lerman, S. (1989). Investigations: Where to now? In P. Ernest (Ed.), *Mathematics teaching: The state of the art* (pp. 73-80). London: Falmer Press.

Lesh, R. A., Lovitts, B., & Kelly, A. E. (2000). Purposes and assumptions of this book. In A. E. Kelly & R. A. Lesh (Eds.), *Handbook of research design in mathematics and science education* (pp. 17-33). Mahwah, NJ: Erlbaum.

Lester, F. K., Jr. (1980). Problem solving: Is it a problem? In M. M. Lindquist (Ed.), *Selected issues in mathematics education* (pp. 29-45). Berkeley, CA: McCutchan.

Lingefjärd, T., & Meier, S. (2010). Teachers as managers of the modelling process. *Mathematics Education Research Journal, 22*, 92-107.

Liu, H. K. (1982). *Heuristic strategies in geometrical problem-solving used by a group of form five students.* Unpublished master's thesis, School of Education, Chinese University of Hong Kong, Hong Kong.

Love, E. (1988). Evaluating mathematical activity. In D. Pimm (Ed.), *Mathematics, teachers and children* (pp. 249-262). London: Hodder and Stoughton.

Maher, C. A. (2005). How students structure their investigations and learn mathematics: insights form a long-term study. *Mathematical Behavior, 24*, 1-14.

Mason, J. (1978). On investigations. *Mathematics Teaching, 84*, 43-47.

Mason, J. (1999). *Learning and doing mathematics* (2nd ed.). York, England: QED.

Mason, J., Burton, L., & Stacey, K. (1985). *Thinking mathematically* (Rev. ed.). Wokingham, England: Addison-Wesley.

Mason, J., & Johnston-Wilder, S. (2006). *Designing and using mathematical tasks.* St Albans, England: Tarquin Publications.

McLeod, D. B. (1992). Research on affect in mathematics education: A reconceptualization. In D. A. Grouws (Ed.), *Handbook of research on mathematics teaching and learning* (pp. 575-596). New York: National Council of Teachers of Mathematics & MacMillan.

Meijer, J., Veenman, M. V. J., & van Hout-Wolters, H. A. M. (2006). Metacognitive activities in test-studying and problem-solving: Development of a taxonomy. *Educational Research and Evaluation, 12*(3), 209-237.

Mevarech, Z., & Fridkin, S. (2006). The effects of IMPROVE on mathematical knowledge, mathematical reasoning and meta-cognition. *Metacognition and Learning, 1*, 85-97.

Miles, M. B., & Huberman, M. A. (1994). *Qualitative data analysis: An expanded sourcebook* (2nd ed.). Beverley Hills: Sage.

Ministry of Education of Singapore. (1990). *Mathematics syllabus: Primary.* Singapore: Curriculum Planning Division.

Ministry of Education of Singapore. (1997). *Shaping our future: Thinking schools, learning nation.* Retrieved February 29, 2012, from http://www.moe.gov.sg/speeches/1997/020697.htm

Ministry of Education of Singapore. (2000). *Mathematics syllabus: Lower secondary.* Singapore: Curriculum Planning and Development Division.

Ministry of Education of Singapore. (2003). *Teach less, learn more.* Retrieved February 29, 2012, from http://www.moe.gov.sg/bluesky/print_tllm.htm

Ministry of Education of Singapore. (2012). *O-level mathematics teaching and learning syllabus.* Singapore: Curriculum Planning and Development Division.

Morgan, C. (1998). *Writing mathematically: The discourse of investigation.* London: Falmer Press.

Moschkovich, J. N. (2002a). An introduction to examining everyday and academic mathematical practices. In E. Yackel (Series Ed.) & M. E. Brenner & J. N. Moschkovich (Monograph Eds.), *Everyday and academic mathematics in the classroom* (pp. 1-11). Reston, VA: National Council of Teachers of Mathematics.

Moschkovich, J. N. (2002b). Bringing together workplace and academic mathematical practices during classroom assessments. In E. Yackel (Series Ed.) & M. E. Brenner & J. N. Moschkovich (Monograph Eds.), *Everyday and academic mathematics in the classroom* (pp. 93-110). Reston, VA: National Council of Teachers of Mathematics.

Mottershead, L. (1985). *Investigations in mathematics.* Oxford, UK: Basil Blackwell.

National Council of Teachers of Mathematics. (1980). *An agenda for action: Recommendations for school mathematics of the 1980s.* Reston, VA: National Council of Teachers of Mathematics.

National Council of Teachers of Mathematics. (1989). *Curriculum and evaluation standards for school mathematics.* Reston, VA: National Council of Teachers of Mathematics.

National Council of Teachers of Mathematics. (1991). *Professional standards for teaching mathematics.* Reston, VA: National Council of Teachers of Mathematics.

National Council of Teachers of Mathematics. (2000). *Principles and standards for school mathematics.* Reston, VA: National Council of Teachers of Mathematics.

Newell, A., & Simon, H. A. (1972). *Human problem solving.* Englewood Cliffs, NJ: Prentice-Hall.

Newman, A. (1983). *The Newman language of mathematics kit.* Australia: Harcourt Brace Jovanovich Group.

Ng, H. C. (2003). *Benefits of using investigative tasks in the primary classroom.* Unpublished master's thesis, National Institute of Education, Nanyang Technological University, Singapore.

Ng, L. E., Teo, L. E. L., & Leow, H. F. (2005, June). *Conducting investigative tasks or mini-projects in the secondary classrooms.* Paper presented in the Mathematics Teachers' Conference 2005, Singapore.

Nielsen, L., Patronis, T., & Skovsmose, O. (1999). *Connecting corners of Europe: A Greek Danish project in mathematics education.* Århus: Systime.

Nisbett, R., & Wilson, T. (1977). Telling more than we can know: Verbal reports on mental processes. *Psychological Review, 84*, 231-259.

Noddings, N. (1990). Constructivism in mathematical education. In R. B. Davis, C. A. Maher, & N. Noddings (Eds.), *Constructivist views on the teaching and learning of mathematics: Journal for Research in Mathematics Education Monograph Number 4* (pp. 7-18). Reston, VA: National Council of Teachers of Mathematics.

Orton, A., & Frobisher, L. (1996). *Insights into teaching mathematics.* London: Cassell.

Pimm, D. (1987). *Speaking mathematically: Communication in mathematics classrooms.* London: Routledge & Kegan Paul.

Pirie, S. (1987). *Mathematical investigations in your classroom: A guide for teachers.* Basingstoke, UK: Macmillan.

Pólya, G. (1957). *How to solve it: A new aspect of mathematical method* (2nd ed.). Princeton, NJ: Princeton University Press.

Pope, R. (2005). *Creativity: Theory, history, practice.* London: Routledge.

Reys, R. E., Lindquist, M. M., Lambdin, D. V., Smith, N. L., & Suydam, M. N. (2012). *Helping children learn mathematics* (10th ed.). Hoboken, NJ: John Wiley & Sons.

Riffe, D., Lacy, S., & Fico, F. (1998). *Analyzing media messages: Quantitative content analysis.* New York: Erlbaum.

Schoenfeld, A. H. (1983). Episodes and executive decisions in mathematical problem solving. In R. Lesh & M. Landau (Eds.), *Acquisition of mathematics concepts and processes.* (pp. 345-395). New York: Academic Press.

Schoenfeld, A. H. (1984). Beyond the purely cognitive: Belief systems, social cognitions and metacognitions as driving forces in intellectual performance. *Cognitive Science, 7*, 329-363.

Schoenfeld, A. H. (1985). *Mathematical problem solving.* Orlando, FL: Academic Press.

Schoenfeld, A. H. (1987). What's all the fuss about metacognition? In A. H. Schoenfeld (Ed.), *Cognitive science and mathematics education* (pp. 189-215). Hillsdale, NJ: Erlbaum.

Schoenfeld, A. H. (1992). Learning to think mathematically: Problem solving, metacognition, and sense making in mathematics. In D. A. Grouws (Ed.), *Handbook of research on mathematics teaching and learning* (pp. 334-370). Reston, VA: National Council of Teachers of Mathematics & Macmillan.

Schoenfeld, A. H. (2002). Research methods in (mathematics) education. In L. D. English (Ed.), *Handbook of international research in mathematics education* (pp. 435-487). Mahwah, NJ: Erlbaum.

Schraw, G. (2001). Promoting general metacognitive awareness. In H. J. Hartman (Ed.), *Metacognition in learning and instruction: Theory, research and practice* (pp. 3-16). The Netherlands: Kluwer Academic.

Schraw, G., & Dennison, R. S. (1994). Assessing metacognitive awareness. *Contemporary Educational Psychology, 19*, 460-475.

Schraw, G., & Moshman, D. (1995). Metacognitive theories. *Educational Psychology Review 7*, 351-371.

Shufelt, G. (Ed.). (1983). *The agenda in action. 1983 Yearbook* (J. R. Smart, General Yearbook Ed.). Reston, VA: National Council of Teachers of Mathematics.

Silver, E. A. (1994). On mathematical problem posing. *For the Learning of Mathematics, 14*(1), 19-28.

Silverman, D. (2005). *Doing qualitative research* (2nd ed.). London: SAGE.

Singapore Government. (2004). *National Day Rally 2004 Speech.* Retrieved February 29, 2012, from http://www.gov.sg/nd/ND04.htm

Skovsmose, O. (2002). Landscapes of investigation. In L. Haggarty (Ed.), *Teaching mathematics in secondary schools* (pp. 115-128). London: RoutledgeFalmer.

Steffe, L. P. (1983). *The teaching experiment methodology in a constructivist research programme.* Paper presented at the Fourth International Congress on Mathematical Education, Boston.

Steffe, L. P., & Thompson, P. W. (2000). Teaching experiment methodology: Underlying principles and essential elements. In A. K. Kelly & R. A. Lesh (Eds.), *Handbook of research design in mathematics and science education* (pp. 267-306). Mahwah, NJ: Erlbaum.

Stein, M, & Burchartz, B. (2006). The invisible wall project: Reasoning and problem solving processes of primary and lower secondary students. *Mathematical Thinking and Learning, 8*(1), 65-90.

Stein, M. K., Grover, B. W., & Henningsen, M. (1996). Building student capacity for mathematical thinking and reasoning: An analysis of mathematical tasks used in reform classrooms. *American Educational Research Journal, 33*, 455-488.

Sternberg, R. J. (1982). A componential approach to intellectual development. In R. J. Sternberg (Ed.), *Advances in the psychology of human intelligence* (Vol. 1, pp. 413-463). Hillsdale, NJ: Erlbaum.

Stylianides, G. J. (2008). An analytical framework of reasoning-and-proving. *For the Learning of Mathematics, 28*(1), 9-16.

Tall, D. (1991). *Advanced mathematical thinking.* Dordrecht, The Netherlands: Kluwer Academic Press.

Tanner, H. (1989). Introducing investigations. *Mathematics Teaching, 127*, 20-23.

Teong, S. K. (2002). An investigative approach to mathematics teaching and learning. *The Mathematics Educator, 6*(2), 32-46.

The MacTutor History of Mathematics Archive (1999). *Archimedes of Syracuse.* Retrieved February 29, 2012, from http://turnbull.mcs.st-and.ac.uk/history/Biographies/Archimedes.html

van Oers, B. (1996). Learning mathematics as a meaningful activity. In L. Steffe, P. Nesher, P. Cobb, G. A. Goldin, & B. Greer (Eds.), *Theories of mathematical learning* (pp. 91-113). Mahwah, NJ: Erlbaum.

van Someren, M. W., Barnard, Y. F., & Sandberg, J. A. C. (1994). *The thinking aloud method: A practical guide to modeling cognitive processes.* London: Academic Press.

Wheeler, D. (1982). Mathematization matters. *For the Learning of Mathematics, 3*(1), 45-47.

Wheeler, D. (1983). Mathematization: Its Nature and Use in Education. In M. Zweng, T. Green, J. Kilpatrick, H. Pollak, & M. Suydam (Eds.), *Proceedings of the Fourth International Congress on Mathematical Education* (pp. 290-291). Boston, MA: Birkhäuser.

Wikipedia (2012). *Self number*. Retrieved February 29, 2012, from http://en.wikipedia.org/wiki/Self_number

Wolf, A. (1990). Testing investigations. In P. Dowling & R. Noss (Eds.), *Mathematics versus the national curriculum* (pp. 137-153). London: Falmer Press.

Wong, K. Y. (1991). Curriculum development in Singapore. In C. Marsh & P. Morris (Eds.), *Curriculum development in East Asia* (pp. 129-160). London: Falmer Press.

Wong, P. S. K. (1989, December). *The effects of academic settings on students' metacognition in mathematical problem solving.* Paper presented at the Third Educational Research Association Annual Conference, Adelaide, Australia.

Yeo, J. B. W. (2007). Mathematically-rich games. In *Electronic Proceedings of International Conference on Education: Redesigning Pedagogy: Culture, Knowledge and Understanding*. Singapore: Centre for Research in Pedagogy & Practice (CRPP), NIE, NTU.

Yeo, J. B. W., Hon, S. F., & Cheng, T. G. H. (2006). Engaged learning in mathematics. In *Electronic Proceedings of Educational Research Association of Singapore Conference: Diversity for Excellence: Engaged Pedagogies*. Singapore: ERAS.

Yip, J. S. K., & Sim, W. K. (1990). *Evolution of educational excellence: 25 years of education in the Republic of Singapore.* Singapore: Longman.

APPENDIX A: INITIAL SMALL STUDY TEST INSTRUMENT

This appendix shows the two tasks for a small study involving a class of pre-service teachers as described in Section 3.3. The purpose was to fine-tune the test instrument for the initial exploratory study.

Task 1

Natural numbers are positive whole numbers: 1, 2, 3, 4, …

Polite numbers are natural numbers that can be expressed as the sum of consecutive natural numbers. For example,

$$9 = 2 + 3 + 4$$
$$11 = 5 + 6$$
$$18 = 3 + 4 + 5 + 6.$$

Investigate.

Questions: Write down your honest thoughts and feelings when you first see this question. Have you seen this question before? Do you know what to investigate? If you don't know what to investigate, what do you do? If you know what to investigate, start investigating now.

Note: It is ok if you don't know what to do. You will *not* get a poorer grade just because you write down you don't know what to do. Just be honest.

Task 2

Natural numbers are positive whole numbers: 1, 2, 3, 4, …

Polite numbers are natural numbers that can be expressed as the sum of consecutive natural numbers. For example,

$$9 = 2 + 3 + 4$$
$$11 = 5 + 6$$
$$18 = 3 + 4 + 5 + 6.$$

Which numbers are polite?

Questions: Do you know how to start if the task is phrased this way? How do you go about finding the answer? Write down what you are **thinking** when you try to solve this problem. We are interested to find out how you try to solve the problem and why you try to solve it in this way. For example, you can write, "I suddenly think of trying out some examples to see if I can see any pattern…" and then you show your working to try out.

Note: We are also interested in your **draft working** as well. It is ok if you write down some working to solve the problem but then you get stuck halfway. You will *not* get a poorer grade just because you write this down. The question will then be, "What are you going to do next?"

APPENDIX B: EXPLORATORY STUDY TEST INSTRUMENT

This appendix shows the three tasks for the initial exploratory study described in Section 3.3. Since this test instrument was not used for the main study, only the tasks (but not the instructions for students and invigilators, the questionnaire for the survey, and the retrospective interview schedule) are shown here due to space constraint.

Mathematical Investigative Task 1: Powers of a Number

9^5 means 9 multiplied by itself 5 times, i.e. $9^5 = 9 \times 9 \times 9 \times 9 \times 9 = 59\,049$.
Powers of 9 are 9^1, 9^2, 9^3, 9^4, 9^5, 9^6, … etc.
Investigate the powers of 9.

For example, you can investigate the following or you can *pose your own questions*:
(a) Find as many patterns as you can about the powers of 9.
(b) Explain why these patterns occur.
(c) Do these patterns occur for powers of other numbers?

Mathematical Investigative Task 2: Polite Numbers

Natural numbers are positive whole numbers: 1, 2, 3, 4, …
Polite numbers are natural numbers that can be expressed as the sum of two or more consecutive natural numbers. For example,

$$9 = 2 + 3 + 4 = 4 + 5$$
$$11 = 5 + 6$$
$$18 = 3 + 4 + 5 + 6.$$

Investigate.

Mathematical Investigative Task 3: Basketball Tournament

In a basketball tournament, each team must play against every other team once.
(a) How many matches will there be if there are 20 teams in the tournament?
(b) What else do you want to investigate? When you pose a question to investigate, write it down first.

APPENDIX C: OUTLINES OF FAMILIARISATION LESSON AND DEVELOPING LESSONS

Table C1 shows the outline and purpose of each lesson for the teaching experiment (see Section 3.6.3a for more details). Lesson 1 was the familiarisation lesson, and Lessons 2 to 6 were developing lessons. Each lesson lasted two hours. The detailed investigative tasks are given in Appendix D and the instructional strategies are given in Section 3.6.3(b).

Table C1 Outlines of Familiarisation Lesson and Developing Lessons

Lesson	Tasks	Purpose	Outline of Lesson
1	• Task 1: Powers of a Number • Task 2: Handshakes	To familiarise students with what to do during mathematical investigation; to make students aware that they can generalise; and to teach students how to think aloud for the pretest	Students were taught what to do when given a task that ended with the word 'Investigate': they were taught to search for any pattern for Type A tasks and to pose specific problems to solve for Type B tasks; in particular, they were taught that they could change the given in Type B tasks to generalise, but it was not emphasised that they should generalise whenever possible; the students also practised thinking aloud during investigation
Pretest	• Pretest Task 1: Happy Numbers • Pretest Task 2: Toast	The first task was to find out whether the students knew how to search for any pattern; the second task was to find out whether the students knew how to pose specific problems to solve and to generalise whenever possible	Each student was videotaped separately while they thought aloud during the pretest
2	• Task 3: Happy Numbers (same as Pretest Task 1) • Task 4: Toast (same as Pretest Task 2)	Focus was on the processes for understanding the task, and what students could have investigated for their pretest tasks: what were some patterns for Pretest Task 1 and how to generalise for Pretest Task 2	Students were reminded of the strategies for understanding textbook questions, e.g. read the task carefully, re-read or rephrase task, highlight key information, and visualise given information; they were taught a new process: try example to understand Type A tasks
3	• Task 5: Magic Trick • Task 6: Amazing Race	Focus was on developing students' problem-posing processes and to convince them that it was alright to pose a difficult problem that they may not be able to solve; and on analysing the feasibility of their goal or problem	Students were taught to pose specific problems for both types of tasks, but focus was on posing problems for Type B tasks, since students were expected to search for any pattern for Type A tasks; they were also taught to analyse whether the problem was too easy and so not worth pursuing, or whether it was too difficult to pursue

Lesson	Tasks	Purpose	Outline of Lesson
4	• Task 7: Squares • Task 8: Palindromic Numbers	Focus was on developing students' problem-solving heuristics, especially, specialising to search for patterns (conjecturing) and using reasoning to solve problems which may result in the formulation of conjectures; and on how to regulate their own investigation using some metacognitive processes	Students were taught to specialise systematically to search for patterns, and to use other heuristics such as reasoning to solve problems or to formulate conjectures; they were also taught to analyse the feasibility of their plan of attack (specialising, using other heuristics, conjecturing) to see whether it was worth pursuing, and to monitor their own progress every 5 minutes or so
5	• Task 9: Chords and Regions • Task 10: Tiles	Focus was on developing students' justifying processes; (Task 9 was a good example to show that an obvious observed pattern was not the underlying pattern); developing the habit to regulate their own investigation; and to let students practise thinking aloud for the posttest	Students were taught that certain results were actually conjectures to be proven or refuted; how to test conjectures using naïve testing; how to justify conjectures using a non-proof argument or a formal proof; how to analyse the feasibility of their justifying plan to see whether it was worth pursuing; and to develop the habit of monitoring their own progress; students also practised thinking aloud during investigation
6	• Task 11: GST and Discount • Task 12: Polite Numbers	Focus was on developing in students the habit to generalise whenever possible and to extend the task; what to do when they were stuck, and in particular, the need to incubate; to develop the habit of checking their working and reviewing their solution; and to let students practise thinking aloud for the posttest	Students were taught they should always extend Type B tasks to generalise whenever possible; to analyse their plan of attack when they were stuck and to incubate if necessary; to occasionally check their working step by step, or by working backwards, or by examining whether the answer was reasonable or logical; to always review their solution after solving a problem to see if it had met the goal of the task, to evaluate the efficacy of their method of solution, and to look for alternative methods; students also practised thinking aloud during investigation
Posttest	• Posttest Task 1: Kaprekar Sequences • Posttest Task 2: Sausage	The first task was to find out whether the students knew how to search for any pattern; the second task was to find out whether the students knew how to pose specific problems to solve, including the habit to generalise	Each student was videotaped separately while they thought aloud during the posttest

APPENDIX D: INVESTIGATIVE TASKS FOR LESSONS, PRETEST AND POSTTEST

This appendix shows the tasks used in the lessons and the two tests. Further elaboration of the tasks and the rationales for choosing them are given in Section 3.6.4.

Lesson 1 (Familiarisation Lesson):

- ***Investigative Task 1: Powers of a Number***

  3^5 means 3 multiplied by itself 5 times, i.e. $3^5 = 3 \times 3 \times 3 \times 3 \times 3 = 243$.
  Powers of 3 are 3^1, 3^2, 3^3, 3^4, 3^5, 3^6, … etc. Investigate.

- ***Investigative Task 2: Handshakes***

  At a workshop, each of the 20 participants shakes hand once with each of the other participants. Investigate.

Pretest

- ***Pretest Investigative Task 1: Square Each Digit and Add***

  Choose any number. Square each digit of the number and add to obtain a new number. Repeat this process for the new number until you have a good reason to stop. Investigate.

- ***Pretest Investigative Task 2: Toast***

  Three slices of bread are to be toasted under a grill. The grill can hold exactly two slices. Only one side of each slice is toasted at a time. It takes 30 seconds to toast one side of a slice of bread, 5 seconds to put a slice in or to take a slice out, and 3 seconds to turn a slice over. Investigate.

Lesson 2 (Developing Lesson):

- ***Investigative Task 3: Square Each Digit and Add***

  [*Same as Pretest Investigative Task 1*]

- ***Investigative Task 4: Toast***

  [*Same as Pretest Investigative Task 2*]

Lesson 3 (Developing Lesson):

- ***Investigative Task 5: Magic Trick***

  Choose any three-digit number. Reverse the digits to get another number. Subtract the smaller number from the larger number to obtain another number. Add this number to its reverse to obtain a final number. Investigate.

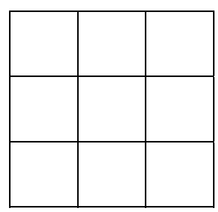
- ***Investigative Task 6: Amazing Race***

  In a 100-metre race, Ali beats Bernard by 10 metres. The two boys plan to have another race (where both boys will still run at the same rate as before). Bernard suggests that he should be given a head start: he wants to start 10 metres in front of the start line while Ali still starts at the starting line. However, Ali disagrees. He suggests that he (Ali) will start 10 metres behind the start line while Bernard still starts at the starting line. Investigate.

Lesson 4 (Developing Lesson):

- ***Investigative Task 7: Squares***

  The diagram below shows a 3 × 3 square grid. There are many squares inside this grid. Investigate.



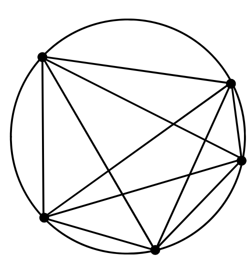
- ***Investigative Task 8: Palindromic Numbers***

  Palindromic numbers read the same forwards and backwards, e.g. 33, 828, 1441 and 71617. Someone claims that all 4-digit palindromic numbers are divisible by a certain number. Investigate.

Lesson 5 (Developing Lesson):

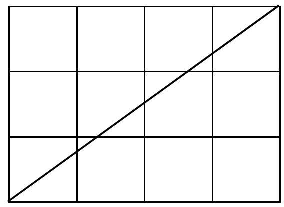
- ***Investigative Task 9: Chords and Regions***

  The figure below shows a circle with 5 points on its circumference. Each point is joined to every other point by a line (called a chord). Notice that no three chords intersect at the same point inside the circle. Investigate.



- ***Investigative Task 10: Tiles***

  The diagram below shows a 4-by-3 rectangular table covered with square tiles and a crack along its diagonal. The crack affects 6 squares which have to be replaced. Investigate.



Lesson 6 (Developing Lesson):

- ***Investigative Task 11: GST and Discount***

  There is a 10% discount on an item that costs $100 but there is also a 7% GST. The salesperson wants to charge your friend the GST first before giving him or her the discount. However, your friend disagrees. He or she insists that the salesperson gives the discount first before charging the GST. Investigate.

- ***Investigative Task 12: Polite Numbers***

  Natural numbers are positive whole numbers: 1, 2, 3, 4, …
  Polite numbers are natural numbers that can be expressed as the sum of two or more consecutive natural numbers. For example,

$$
\begin{aligned}
9 &= 2 + 3 + 4 = 4 + 5 \\
11 &= 5 + 6 \\
18 &= 3 + 4 + 5 + 6.
\end{aligned}
$$

  Investigate.

Posttest

- ***Posttest Investigative Task 1: Add Sum of Digits to Number***

  Choose any number. Add the sum of its digits to the number itself to obtain a new number. Repeat this process for the new number and so forth. Investigate.

- ***Posttest Investigative Task 2: Sausages***

  I need to cut 12 identical sausages so that I can share them equally among 18 people. Investigate.

APPENDIX E: TASK ANALYSES FOR THE PRETEST AND POSTTEST ITEMS

This appendix contains the task analyses for the two investigative tasks used in the pretest and the two investigative tasks used in the posttest. Secondary 2 students are not expected to discover the more complicated patterns or to justify conjectures using complicated formal proofs. But the task analyses should be more comprehensive, i.e. they will include the more complicated patterns and formal proofs if applicable.

Task Analysis for Pretest Task 1 (Happy Numbers[29])

> **Pretest Investigative Task 1: Square Each Digit and Add**
>
> Choose any number. Square each digit of the number and add to obtain a new number. Repeat this process for the new number until you have a good reason to stop. Investigate.

Stage 1: Understanding the Task (U)

- Students should try an example to make sense of the task. For example:

$$\begin{aligned} & 28 \\ \rightarrow \quad 2^2 + 8^2 &= 68 \\ \rightarrow \quad 6^2 + 8^2 &= 100 \\ \rightarrow \quad 1^2 + 0^2 + 0^2 &= 1 \end{aligned}$$

- Possible mistakes or problems in understanding the task:
  - misinterpret the 'new number' in the task statement to mean 'choosing a completely new number', and thus does not repeat the process for the new number;
  - does not know how to add the squares of the digits for a one-digit number.

Stage 2: Problem Posing (P)

- Students should pose the general problem of searching for any pattern.
- Students could also pose specific problems to investigate. The following shows a list of trivial and non-trivial problems. The classification of a problem as trivial or non-trivial has been subjected to an inter-coder reliability test in Section 5.4.
- Some trivial problems:
  - Is there any pattern in consecutive new numbers (i.e. consecutive terms of the sequence)?
  - Is there any pattern in the differences between consecutive new numbers?
  - Is there any pattern in the last digit of consecutive new numbers?
- Some non-trivial problems:
  - Is there a general formula to obtain the next term of the sequence?
  - Is there a general formula for a happy number or a sad number?
  - Are there more happy numbers than sad numbers?
  - Are there infinitely many happy numbers and sad numbers?
  - Is the sum of two happy numbers happy or sad?
  - Is the product of two happy numbers happy or sad?

[29] The term 'Happy Numbers' will be used in this thesis for ease of discussion, but the students in the present study were not expected to know the term. In fact, the heading for this task in the pretest was 'Square Each Digit and Add'.

Stage 3: Specialising (S)

- Students should try examples systematically to search for patterns, instead of using other heuristics.
- Students should discover four shortcuts when specialising:
  – if a number appears in a zapping[30] sequence, there is no need to use that number as a starting number to investigate, e.g. if the student has found the zapping sequence 28, 68, 100, 1, ... , there is no need to investigate what will happen if the starting number is 28, 68, 100 or 1, because all these numbers will belong to the same sequence[31];
  – if a number appears in a zapping sequence, there is no need to investigate when will happen for another number obtained by changing the order of the digits of the former number, e.g. the numbers 123, 132, 213, 231, 312 and 321 will all give the same number in the next zapping;
  – if a number appears in a zapping sequence, there is no need to investigate when will happen for another number obtained by adding zeros to the former number, e.g. the numbers 12, 102, 120 and 1002 will all give the same number in the next zapping;
  – if a number in a zapping sequence appears in another zapping sequence, then the two sequences can be merged at that number to investigate together.

Stage 4: Conjecturing (C)

- For ease of discussion in this thesis, the following shows two types of patterns.

Pattern 1: Some zapping sequences will terminate at the number 1, e.g.

$$\begin{array}{lr} & 28 \\ \rightarrow & 2^2 + 8^2 = 68 \\ \rightarrow & 6^2 + 8^2 = 100 \\ \rightarrow & 1^2 + 0^2 + 0^2 = 1 \\ \rightarrow & 1^2 = 1 \end{array}$$

All the numbers in the zapping sequence, 28, 68, 100 and 1, are called *Happy Numbers*. The following shows a list of happy numbers less than or equal to 100 for easy reference: 1, 7, 10, 13, 19, 23, 28, 31, 32, 44, 49, 68, 70, 79, 82, 86, 91, 94, 97, 100.

Pattern 2: Some zapping sequences will end in the same loop. The following shows some numbers in such sequences. These numbers are called *Sad Numbers*.

$$\begin{array}{ccccccccccccc} & & & & 15 & \rightarrow & 26 & & 65 & \leftarrow & 18 & \leftarrow & 33 \\ & & & & & & \downarrow & & \downarrow & & & & \\ & & & & 62 & \rightarrow & 40 & & 61 & \leftarrow & 56 & & \\ & & & & & & \downarrow & & \downarrow & & & & \\ 11 & \rightarrow & 2 & \rightarrow & \mathbf{4} & \rightarrow & \mathbf{16} & \rightarrow & \mathbf{37} & \rightarrow & \mathbf{58} & & \\ & & & & \uparrow & & & & & & \downarrow & & \\ & & & & \mathbf{20} & \leftarrow & \mathbf{42} & \leftarrow & \mathbf{145} & \leftarrow & \mathbf{89} & & \\ & & & & & & \uparrow & & & & & & \\ & & & & & & 154 & & & & & & \end{array}$$

30 The process of adding the square of each digit of a number is called ‘zapping’, and the sequence is called a ‘zapping sequence’. The students in the present study were not expected to know these terms or other technical terms such as ‘happy number’ and ‘sad number’.

31 The reader may think that this is obvious, but it was not obvious to some students in this study.

- The following shows a list of trivial and non-trivial conjectures. The classification of a pattern or conjecture as trivial or non-trivial has been subjected to an inter-coder reliability test in Section 5.4. Conjectures 1-3 are trivial while Conjectures 4-9 are non-trivial. Conjectures 4, 8 and 9 are false, while Conjectures 1-3, 5 and 7 are true. I do not know whether Conjecture 6 is true or false. This is a good example to illustrate that it is not possible for the teacher to know everything in mathematical investigation.

  Conjecture 1: If a number in a zapping sequence is happy or sad, then all the numbers in the same sequence are also happy or sad respectively, e.g. 28 → 68 → 100 → 1, so all the numbers, 28, 68, 100 and 1, are happy numbers. [Trivial; true]

  Conjecture 2: The rearrangement of the digits of a number does not matter in determining whether the number is happy or sad, e.g. 28 and 82 are both happy. [Trivial; true]

  Conjecture 3: The insertion or removal of any number of zeros anywhere in a number does not affect whether the number is happy or sad, e.g. 47, 407 and 7040 are all sad. [Trivial; true]

  Conjecture 4: When two zapping sequences first merge at the same number, the preceding terms before this number in the two sequences will be different numbers but with the *same unique combination*, i.e. the difference in the preceding terms will just be rearrangements of the digits and/or insertion of zeros anywhere in the terms. For example, if two zapping sequences merge at the same number 100, then the preceding terms will be one of these numbers: 68, 86, 608, 680, 806, 860, 6008, etc. [Non-trivial; false]

  Conjecture 5: A positive integer is either a happy or a sad number. [Non-trivial; true]

  Conjecture 6: There are more sad numbers than happy numbers. [Non-trivial]

  Conjecture 7: There are infinitely many happy numbers. Similarly, there are infinitely many sad numbers. [Non-trivial; true]

  Conjecture 8: The sum of two happy numbers is always happy. Similarly, the sum of two sad numbers is always sad. [Non-trivial; false]

  Conjecture 9: The product of two happy numbers is always happy. Similarly, the product of two sad numbers is always sad. [Non-trivial; false]

Stage 5: Justifying (J)

- The following shows how to prove or refute the above conjectures. Secondary 2 students are not expected to produce the more difficult proofs, such as the one for Conjecture 5.

  Conjecture 1: This is clearly true but we can still use a simple argument. Suppose a number in a zapping sequence is happy. Then the sequence will terminate at the number 1. Thus all the numbers in this sequence will be happy. A similar argument applies for sad numbers.

  Conjecture 2: The following shows two different kinds of justification.

Non-Proof Argument:

The rearrangement of the digits of a number does not matter in determining whether the number is happy or sad because the sum of the squares of digits does not depend on the order of the digits.

Formal Proof:

Let the digits of a number x be a_1, a_2, a_3, … a_n, where a_i is the i-th digit of the term. Then the next term in the zapping sequence will be:

$$a_1^2 + a_2^2 + a_3^2 + \ldots + a_n^2.$$

Consider another number y with the same digits as x but arranged in a different order. Then the next term in the sequence will still be equal to:

$$a_1^2 + a_2^2 + a_3^2 + \ldots + a_n^2.$$

Thus if the next term $a_1^2 + a_2^2 + a_3^2 + \ldots + a_n^2$ is happy or sad, then Conjecture 1 (which is a proven result) will imply that both x and y will be happy or sad respectively.
Therefore, the rearrangement of the digits of a number does not matter in determining whether the number is happy or sad.

Conjecture 3: The insertion or removal of any number of zeros anywhere in a number does not affect whether the number is happy or sad because the squares of zeros, which will still be zeros, will not affect the sum of the squares of the digits. Students can use a formal proof to justify this conjecture, which will be similar to the formal proof for Conjecture 2.

Conjecture 4: This conjecture is *false*. A counter example is 68 and 5555. Both numbers will give the same number 100 in the next zap.

Conjecture 5: If a number n has m digits, then the next term in the zapping sequence will be at most $9^2 \times m$, i.e. $81m$.
If $m \geq 4$, then $n \geq 10^{m-1} > 81m$ (to prove that $10^{m-1} > 81m$ for $m \geq 4$, we can show that the function $y = 10^{x-1} - 81x$ is strictly increasing for $x \geq 4$ by showing that $\frac{dy}{dx} > 0$ for $x \geq 4$).
This means that if a term of a zapping sequence has at least 4 digits, the next term will be strictly less than this term, i.e. the sequence is strictly decreasing until it reaches a term that has less than 4 digits.

Now consider a 3-digit number. The largest sum of the squares of the digits is $81 \times 3 = 243$, which occurs for the number 999. This means that if n is a 3-digit number, the next term will be less than or equal to 243.
Now consider a 3-digit number less than or equal to 243. The largest sum of the squares of the digits is 163, which occurs for the number 199. This means that if n is a 3-digit number less than or equal to 243, the next term will be less than or equal to 163.
Now consider a 3-digit number less than or equal to 163. The largest sum of the squares of the digits is 107, which occurs for the number 159. This means that if n is a 3-digit number less than or equal to 163, the next term will be less than or equal to 107.

Now consider a 3-digit number less than or equal to 107. The largest sum of the squares of the digits is 50, which occurs for the number 107. This means that if *n* is a 3-digit number less than or equal to 107, the next term will be less than or equal to 50.

The above shows that if a term of a zapping sequence has 3 digits, the next term will be less than this term, i.e. the sequence is strictly decreasing until it reaches a term that has less than 3 digits. In other words, no matter what the starting number is, the zapping sequence is strictly decreasing until it drops below 100.

Although a two-digit number may produce a three-digit number in the next term, from the above argument, the three-digit number will start decreasing to a two-digit number again.

Now, using exhaustive listing, a number that is less than 100 will either end up with the number 1, or go into the same loop described in Stage 4. Hence, a positive integer is either a happy or a sad number.

Conjecture 6: Although there are more sad numbers than happy numbers in the first 100 positive integers (by exhaustive listing), I do not know how to prove or refute this conjecture. This is a good example to illustrate that it is not possible for the teacher to know everything in mathematical investigation.

Conjecture 7: There are infinitely many happy numbers because zeros can be inserted anywhere in any happy number, and the resulting numbers will still be happy. A similar argument applies for sad numbers.

Conjecture 8: This conjecture is *false*. Counter examples: The sum of the two happy numbers, 7 and 10, is 17, which is sad[32]; while the sum of the two sad numbers, 3 and 4, is 7, which is happy. (On the other hand, the sum of two happy numbers is not always sad, e.g. 10 + 13 = 23, which are all happy numbers; while the sum of two sad numbers is not always happy, e.g. 2 + 3 = 5, which are all sad numbers.)

Conjecture 9: This conjecture is *false*. Counter examples: The product of the two happy numbers, 7 and 23, is 161, which is sad; while the product of the two sad numbers, 2 and 5, is 10, which is happy. (On the other hand, the product of two happy numbers is not always sad, e.g. $7 \times 10 = 70$, which are all happy numbers; while the product of two sad numbers is not always happy, e.g. $2 \times 3 = 6$, which are all sad numbers.)

Stage 6: Generalising (G)

- Conjectures 1-3,5-7 will lead to a generalisation if they are proven. Conjectures 4, 8 and 9 are not general results because they are false.

Stage 7: Checking (R)

- Students need to check:
  - all calculations for repeating the process are correct;
  - the arguments used in the proving of conjectures are sound.

[32] For ease of reference, happy numbers less than or equal to 100 are: 1, 7, 10, 13, 19, 23, 28, 31, 32, 44, 49, 68, 70, 79, 82, 86, 91, 94, 97, 100.

- Students have three possibilities to choose from at this juncture:
  - they can search for more patterns without extending the task, e.g. if they have found and justified Conjecture 1, they can search for more patterns; or they could pose the specific problems listed in Stage 2 (go to Stage 2);
  - they can extend the task by changing the given (go to Stage 8);
  - they can stop, i.e. finish the investigation.

Stage 8: Extension (E)

- Students are not expected to extend the task within the 30 minutes given for the test because there are many underlying patterns to observe, although some students may extend the task. Moreover, extension of a Type A task usually ends up with a completely new task with entirely different patterns.
- The following shows a list of possible extensions for longer investigation:
  - Find the difference (instead of the sum) of the squares of the digits of a number.
  - Find the product of the squares of the digits of a number.
  - Find the sum of the cubes of the digits of a number.
  - Find the sum of the *n*-th powers of the digits of a number.

Task Analysis for Pretest Task 2 (Toast)

> **Pretest Investigative Task 2: Toast**
>
> Three slices of bread are to be toasted under a grill. The grill can hold exactly two slices. Only one side of each slice is toasted at a time. It takes 30 seconds to toast one side of a slice of bread, 5 seconds to put a slice in or to take a slice out, and 3 seconds to turn a slice over. Investigate.

Stage 1: Understanding the Task (U)

- Students should highlight key words such as the number of slices to be toasted, the number of slices the grill can hold, and all the given timings.
- Possible mistakes or problems in understanding the task:
  - need to subscribe to the assumption that the actions, such as putting in, turning over, and taking out, can be performed on two slices at the same time since a person has two hands (this is called the *First or Usual Interpretation*)
  - if the students think that only one action can be performed at a time, they need to take into account that the first slice will have been toasted for 5 seconds by the time the second slice is put inside, etc. (this is called the *Second or Alternative Interpretation*)
  - unless the students mention that the grill will be turned on only after the second slice is put inside the grill, and the grill will be turned off after toasting the two slices before taking out the first slice, etc., with the assumption that the time taken to turn the grill on or off is negligible (this is called the *Third or Possible Interpretation*)
  - a common mistake for performing one action at a time is to ignore the fact that the first slice will have been toasted for 5 seconds by the time the second slice is put inside the grill (this is called the *Fourth or Wrong Interpretation*), unless there is mention of switching the grill on or off as in the *Third Interpretation*.

Stage 2: Problem Posing (P)

- Students need to pose specific problems to solve. The following shows a list of possible specific problems, which have been classified as trivial or non-trivial (the classification has passed the inter-coder reliability test as discussed in Section 5.4). P4 is the intended problem (see Section 2.2.3b for a brief explanation of intended problems).

  - *Problem 1 (P1):* Find how to toast the three slices of bread. [Trivial]

  - *Problem 2 (P2):* Find the time taken to toast the three slices of bread. [Trivial]

  - *Problem 3 (P3):* Find a few methods to toast the three slices of bread. [Non-Trivial]

  - *Problem 4 (P4):* Find the shortest time to toast the three slices of bread. [Non-Trivial; Intended Problem]

Stage 3: Specialising and Using Other Heuristics (S/H)

- Students should use other heuristics, such as reasoning, to solve the problems posed in Stage 2. They should consider various methods to toast the bread, which will give a different total toasting time depending on which interpretation that is used (see Stage 1), although the steps for the various methods are essentially the same. The timings shown below for the various methods are for the *First or Usual Interpretation*.

– *Toasting Method A (Usual Method):* Put two slices in, turn them over and then take them out before putting in the last slice.

Action	Time Taken
Put in the first two slices	5 s
Toast one side of the first two slices	30 s
Turn over the first two slices	3 s
Toast the other side of the first two slices	30 s
Take out the first two slices	5 s
Put in the third slice	5 s
Toast one side of the third slice	30 s
Turn over the third slice	3 s
Toast the other side of the third slice	30 s
Take out the third slice	5 s
Total Toasting Time	**146 s**

– *Toasting Method B (Shortest Method):* Put two slices in; turn one slice over but take out the other slice to put the third slice in[33]; then take out the first slice that is toasted on both sides and put in the second slice that is toasted on one side, and turn over the third slice. This method will give the shortest toasting time: 113 seconds.

Concurrent Action	Time Taken	Concurrent Action	Time Taken
Put in Slice 1	5 s	Put in Slice 2	5 s
Toast Slice 1 Side X	30 s	Toast Slice 2 Side X	30 s
Turn over Slice 1	3 s	Take out Slice 2	5 s
Toast Slice 1 Side Y	30 s	Put in Slice 3	5 s
Take out Slice 1	5 s	Toast Slice 3 Side X	30 s
Put in Slice 2	5 s	Turn over Slice 3	3 s
Toast Slice 2 Side Y	30 s	Toast Slice 3 Side Y	30 s
Take out Slice 2	5 s	Take out Slice 3	5 s
Total Toasting Time	**113 s**	**Total Toasting Time**	**113 s**

– *Toasting Method C (Longest Method)[34]:* Toast one slice at a time.

Action	Time Taken
Put in the first slice	5 s
Toast one side of the first slice	30 s
Turn over the first slice	3 s
Toast the other side of the first slice	30 s
Take out the first slice	5 s
Put in the second slice	5 s
Toast one side of the second slice	30 s
Turn over the second slice	3 s
Toast the other side of the second slice	30 s
Take out the second slice	5 s
Put in the third slice	5 s
Toast one side of the third slice	30 s
Turn over the third slice	3 s
Toast the other side of the third slice	30 s
Take out the third slice	5 s
Total Toasting Time	**219 s**

[33] It is assumed that it is possible to turn a slice (which takes only 3 s) and start toasting, when the other slice is being taken out and the third slice put in (which takes a total of 10 s). Even if this assumption is not valid, meaning that the third slice has to be put in before the turned over slice can start toasting, Toasting Method B will still give the least total toasting time at 120 s.

[34] This is the longest method if no time is wasted doing nothing at all. Some students in the present study actually used Toasting Method C.

Stage 4: Conjecturing (C)

- Students can solve Problems 1-3 using either Toasting Method A, B or C without forming any conjecture.
- Students need to form a conjecture for Problem 4 that Toasting Method B will give the shortest toasting time. This conjecture is non-trivial.

Stage 5: Justifying (J)

- For Problem 4, students need to justify the conjecture that Toasting Method B will give the shortest toasting time. They can use a non-proof argument:

  Proof of Conjecture that Toasting Method B will give the shortest toasting time

  At the start, there is a need to put in both slices so that no space in the grill will be wasted toasting nothing. In other words, Toasting Method C will waste the toasting space.
  After both slices are toasted on one side, there are only 3 possibilities:

  (i) turn both slices over (Toasting Method A)
  (ii) turn one slice over and take out the other slice (Toasting Method B)
  (iii) take out both slices.

  Possibility (iii) will take a longer toasting time than Possibility (i), which uses Toasting Method A, because the two slices need to be put back into the grill again.
  Comparing the total toasting time for Possibility (i) using Toasting Method A, and for Possibility (ii) using Toasting Method B, it can be shown that Method B will give the shortest toasting time at 113 seconds.

Stage 6: Generalising (G)

- The solutions of Problems 1-4 are not general results. Generalisation can take place later if the students extend the task to generalise (see Stage 8).

Stage 7: Checking (R)

- Students need to check:
  - all calculations of the time taken are correct;
  - the argument used in the proving of the conjecture is sound.
- Students have three possibilities to choose from at this juncture:
  - they can pose more problems to solve without extending the task, e.g. if they have posed only Problem 1, they can now pose Problem 2, 3 or 4 to solve (go to Stage 2)
  - they can extend the task by changing the given (go to Stage 8)
  - they can stop, i.e. finish the investigation, although for a Type B task like this task, they are expected to extend to generalise.

Stage 8: Extension (E)

- Students need to extend the task by changing the given in order to generalise. The following shows a list of possible problems to extend, which have been classified as trivial or non-trivial (the classification has passed the inter-coder reliability test as discussed in Section 5.4). Both E1 and E2 are the intended extensions, where E2 is at a higher level of generalisation than E1.

– *Extension 1 (E1):* Find the shortest time needed to toast *n* slices if the grill can hold exactly two slices. [Non-trivial; Intended Extension]

– *Extension 2 (E2):* Find the shortest time needed to toast *n* slices if the grill can hold exactly *m* slices. [Non-Trivial; Intended Extension]

- Students can also extend the tasks by changing other variables, such as the timings given in the task statement. However, changing the timings will affect Toasting Method A and Toasting Method B if the time taken to turn a slice over is more than the total time taken to take out a slice and to put in another slice, but this is not realistic.

Stage 3: Specialising and Using Other Heuristics (S/H)

Since students are expected to extend this task by changing the given, then the task analysis will continue for another cycle to include some other outcomes.

- Students should go back to Stage 3 to specialise by trying different values for *n* and / or *m* in order to find a general formula for the shortest toasting time. But for each case of *n* and / or *m* that they investigate, they still have to use other heuristics to find a toasting method that will produce the shortest toasting time.
- Students may think that they should still use Toasting Method B to produce the shortest toasting time for the general case since Method B is the shortest method for toasting 3 slices of bread. But this really depends on the values of *n* and *m*.
- The following shows the solution for E1 using the *First or Usual Interpretation* (students are not expected to solve E2 within the 30 minutes for the test).

Solution for E1:

Redefine Toasting Method A for toasting two slices of bread only:
– put in both slices
– toast one side of both slices
– turn over both slices
– toast the other side of both slices
– take out both slices.

Therefore, the shortest time needed to toast two slices is 73 s.

Toasting Method B remains the same for toasting three slices of bread, where the shortest toasting time is 113 s.

There are two methods to toast *n* slices of bread in a grill that can hold exactly two slices.

First Method:

– If *n* is even, toast two slices at a time using Toasting Method A repeatedly, so total toasting time = $\frac{73}{2}n$ seconds.

– If *n* is odd and $n \neq 1$, toast the first three slices using Toasting Method B, and then the remaining even number of slices using Toasting Method A repeatedly for two slices at a time, so total toasting time = $113+\frac{73}{2}(n-3)=\frac{73}{2}n+\frac{7}{2}$.

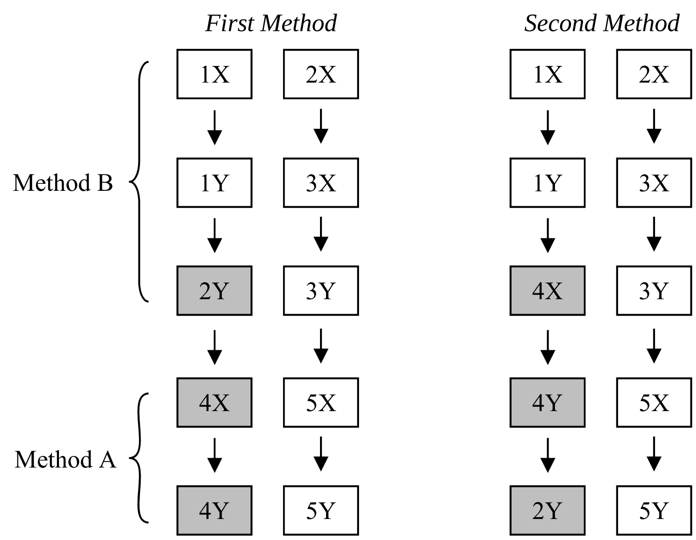
– If $n = 1$, toast using Toasting Method A but applied to one slice only, so total toasting time = 73 s (which is the same as $n = 2$).

Second Method (more complicated but will give same timing as *First Method*):

- If n is even, or if $n = 1$ or 3, the *Second Method* is the same as the *First Method*.
- If n is odd and $n \geq 5$, the *Second Method* is different.

The following will illustrate the difference between the *First Method* and the *Second Method* for $n = 5$ (see shaded slices):

Legend: The 5 slices of bread are numbered 1 to 5. Each slice has two sides: Side X and Side Y. So Slice 1X represents Slice 1 Side X will be toasted.



The rationale for the *Second Method* is at the third stage of toasting where there is another possibility: instead of putting in Slice 2 and toasting its second side, put in a totally untoasted slice (Slice 4) to toast. This method also minimises the time, but as it turns out, both the *First Method* and the *Second Method* will give the same total toasting time of 186 s for $n = 5$, although the timing for each stage of toasting may be different. Similarly, the students can examine other odd values of n and observe that both methods are fundamentally the same.

Stage 4: Conjecturing (C)

- For E1 (find shortest time to toast n slices), students are expected to formulate a conjecture that the *First (or Second) Method* will give the shortest toasting time. This conjecture is non-trivial. The timings are from Stage 3 *First Method* above.

  - If n is even, shortest toasting time = $\frac{73}{2}n$ seconds.
  - If n is odd and $n \neq 1$, shortest toasting time = $\frac{73}{2}n + \frac{7}{2}$ seconds.
  - If $n = 1$, shortest toasting time = 73 seconds.

Stage 5: Justifying (J)

- For E1, students need to justify the conjecture formulated in the previous stage. They can use a non-proof argument. It is easier to use the *First Method* than the *Second Method* since the *First Method* is a straight forward combination of Method A and Method B.

Proof of Conjecture that First Method will give the shortest toasting time

As proven earlier, Toasting Method B will give the shortest time to toast 3 slices of bread. Using a similar argument, Toasting Method A will give the shortest time to toast 2 slices. Therefore, if n is even, a repeated usage of Method A to toast 2 slices at a time will give the shortest toasting time of $\frac{73}{2}n$ seconds.

If n is odd and $n \neq 1$, using Method B will give the shortest time to toast the first three slices. Then what remains is an even number of slices. So by applying Method A repeatedly, we will also get the shortest toasting time, which is $\frac{73}{2}n+\frac{7}{2}$ seconds.

If $n = 1$, there is no way to toast it other than to apply Method A for one slice only, so this will also give the shortest toasting time, which is 73 seconds.

Stage 6: Generalising (G)

- The solution for E1 is a general result, thus generalisation will take place if the students can solve this extension correctly. The second level of generalisation can be obtained by solving LE2. Another level of generalisation can also be obtained by generalising the timings given in the task statement.

Stage 7: Checking (R)

- Students need to check:
  - all calculations of the time taken are correct;
  - the argument used in the proving of the conjecture is sound.
- Students have two possibilities to choose from at this juncture:
  - they can pose more problems to extend the task by changing the given (go to Stage 8);
  - they can stop, i.e. finish the investigation.

Task Analysis for Posttest Task 1 (Kaprekar Sequences[35])

Posttest Investigative Task 1: Add Sum of Digits to Number Choose any number. Add the sum of its digits to the number itself to obtain a new number. Repeat this process for the new number and so forth. Investigate.

Stage 1: Understanding the Task (U)

- Students should try an example to make sense of the task. For example:

$$\begin{aligned} & 28 \\ \rightarrow\ & 28 + 2 + 8 = 38 \\ \rightarrow\ & 38 + 3 + 8 = 49 \\ \rightarrow\ & 49 + 4 + 9 = 62 \end{aligned}$$

- Possible mistakes or problems in understanding the task:
  – misinterpret the 'new number' in the task statement to mean 'choosing a completely new number', and thus does not repeat the process for the new number;
  – does not know how to find the sum of the digits for a one-digit number.

Stage 2: Problem Posing (P)

- Students should pose the general problem of searching for any pattern.
- Students could also pose specific problems to investigate. The following shows a list of trivial and non-trivial problems. The classification of a problem as trivial or non-trivial has been subjected to an inter-coder reliability test in Section 5.4.
- Some trivial problems:
  – Is there any pattern in consecutive new numbers (i.e. consecutive terms of the sequence)?
  – Is there any pattern in the differences between consecutive new numbers (which are the same as the sums of digits of consecutive new numbers)?
  – Is there any pattern in the last digit of consecutive new numbers?
- Some non-trivial problems:
  – Is there a general formula to obtain the next term of the sequence?
  – Are there numbers that will never appear as the second or subsequent terms of any Kaprekar sequence (these are called self numbers)?
  – Can a number appear in two Kaprekar sequences that start with a different self number?
  – Is there any pattern in consecutive self numbers?
  – Is there a general formula for self numbers?
  – Are there infinitely many self numbers?

Stage 3: Specialising (S)

- Students should try examples systematically to search for patterns, instead of using other heuristics.

[35] The term 'Kaprekar Sequences' will be used in this thesis for ease of discussion, but the students in the present study were not expected to know the term or other technical terms such as 'self number'. In fact, the heading for this task in the posttest was 'Add Sum of Digits to Number'.

- Students should discover two shortcuts when specialising:
  - if a number appears in a Kaprekar sequence, there is no need to use that number as a starting number to investigate, e.g. if the student has found the Kaprekar sequence 20, 22, 26, 34, ... , there is no need to investigate what will happen if the starting number is 22, 26 or 34, because all these numbers will belong to the same sequence[36];
  - if a number in a Kaprekar sequence appears in another Kaprekar sequence, then the two sequences can be merged at that number to investigate together.

- For ease of discussion in Stage 4 onwards, the following shows the first seven Kaprekar sequences that start with a self number. There are two types of Kaprekar sequences: Type 1 (all terms divisible by 3 or 9) and Type 2 (all terms not divisible by 3 or 9). Type 1 sequences are further divided into Type 1a (all terms divisible by 3 but not by 9) and Type 1b (all terms divisible by 9).

  - ***Kaprekar Sequence 1 (Type 2):*** 1, 2, 4, 8, 16, 23, 28, 38, 49, 62, 70, 77, 91, 101, 103, **107**, 115, 122, 127, 137, 148, 161, 169, 185, 199, 218, ...

  - ***Kaprekar Sequence 2 (Type 1a):*** 3, 6, 12, 15, 21, 24, 30, 33, 39, 51, 57, 69, 84, 96, 111, 114, 120, 123, 129, 141, 147, 159, 174, 186, 201, 204, 210, ...

  - ***Kaprekar Sequence 3 (Type 2):*** 5, 10, 11, 13, 17, 25, 32, 37, 47, 58, 71, 79, 95, **109**, 119, 130, 134, 142, 149, 163, 173, 184, 197, **214**, ...

  - ***Kaprekar Sequence 4 (Type 2):*** 7, 14, 19, 29, 40, 44, 52, 59, 73, 83, 94, **107** [same number as 16th term of Kaprekar Sequence 1]

  - ***Kaprekar Sequence 5 (Type 1b):*** 9, 18, 27, 36, 45, 54, 63, 72, 81, 90, 99, 117, 126, 135, 144, 153, 162, 171, 180, 189, 207, 216, ...

  - ***Kaprekar Sequence 6 (Type 2):*** 20, 22, 26, 34, 41, 46, 56, 67, 80, 88, 104, **109** [same number as 14th term of Kaprekar Sequence 3]

  - ***Kaprekar Sequence 7 (Type 2):*** 31, 35, 43, 50, 55, 65, 76, 89, 106, 113, 118, 128, 139, 152, 160, 167, 181, 191, 202, 206, **214** [same number as 24th term of Kaprekar Sequence 3]

Stage 4: Conjecturing (C)

- For ease of discussion in this thesis, the following shows six types of patterns.

  - ***Type 1a 'Multiples' Pattern:*** In a Type 1a sequence, all the terms, all the differences between consecutive terms, and all consecutive sums of digits, are divisible by 3 but not by 9.

  - ***Type 1b 'Multiples' Pattern:*** In a Type 1b sequence, all the terms, all the differences between consecutive terms, and all consecutive sums of digits, are divisible by 9.

  - ***Type 2 'Multiples' Pattern:*** In a Type 2 sequence, all the terms, all the differences between consecutive terms, and all consecutive sums of digits, are not divisible by 3 or 9.

[36] The reader may think that this is obvious, but it was not obvious to some students in this study.

– ***Type 1a 'Digital Roots' Pattern:*** The digital roots[37] of the differences between consecutive terms of a Type 1a sequence will repeat with a period of 2, i.e. they will alternate between 3 and 6.

– ***Type 1b 'Digital Roots' Pattern:*** The digital roots of the differences between consecutive terms of a Type 1b sequence will always be equal to 9.

– ***Type 2 'Digital Roots' Pattern:*** The digital roots of the differences between consecutive terms of a Type 2 sequence will repeat with a period of 6, i.e. they will alternate among 1, 2, 4, 8, 7 and 5 in this order. To illustrate this for Kaprekar Sequence 1 (Type 2) in Stage 3 above, the first 24 differences between consecutive terms are listed below and the number in [ ] is the digital root of the difference between consecutive terms, e.g. 13[4] means that the digital root of 13 is 4.

1 [1],	2 [2],	4 [4],	8 [8],	7 [7],	5 [5],
10 [1],	11 [2],	13 [4],	8 [8],	7 [7],	14 [5],
10 [1],	2 [2],	4 [4],	8 [8],	7 [7],	5 [5],
10 [1],	11 [2],	13 [4],	8 [8],	16 [7],	14 [5]...

- The following shows a list of trivial and non-trivial conjectures. The classification of a conjecture as trivial or non-trivial has been subjected to an inter-coder reliability test in Section 5.4. Conjectures 1, 3-5 are trivial while Conjectures 2, 6-9 are non-trivial. Conjectures 1-7 and 9 are true. I do not know whether Conjecture 8 is true or false. This is a good example to illustrate that it is not possible for the teacher to know everything in mathematical investigation.

Conjecture 1: Every Kaprekar sequence is an increasing sequence, so the terms in each sequence will not repeat themselves (i.e. the sequence will not terminate or go into a cycle like the Happy and Sad Numbers in Pretest Task 1 respectively), e.g. 28, 38, 49, 62, … [Trivial; true]

Conjecture 2: Some numbers will never appear as the second or subsequent terms in any Kaprekar sequence, e.g. 1, 3, 5, 7, 9, 20, 31, 42, 53, 64, 75, 86, 97, 108, 110, 121, 132, 143, … (these are called *self numbers* and they will only appear as the first term in one unique Kaprekar sequence: see the seven sequences in Stage 3 above). [Non-trivial; true]

Conjecture 3: All one-digit odd numbers are self numbers (see the first five sequences in Stage 3 above). [Trivial; true]

Conjecture 4: Difference between consecutive one-digit self numbers is always 2 (see the first 5 sequences in Stage 3 above). [Trivial; true]

Conjecture 5: Difference between the last one-digit self number 9 and the first two-digit self number 20, and difference between consecutive two-digit self numbers[38] (i.e. 20, 31, 42, 53, 64, 75, 86, 97) is always 11. [Trivial; true]

Conjecture 6: A 1-digit or a 2-digit number in a Kaprekar sequence will never appear in another Kaprekar sequence with a different self number, but a 3-digit number can appear in two Kaprekar sequences with different self numbers. [Non-trivial; true]

---

[37] The digital root of a number is found by adding the digits of the number continuously until a 1-digit number is obtained, e.g. 339 → 3 + 3 + 9 = 15 → 1 + 5 = 6 means that the digital root of 339 is 6.

[38] The pattern for consecutive three-digit self numbers is more complicated.

Conjecture 7: If the starting number of a Kaprekar sequence is a multiple of 3 or 9, then all its terms are also multiples of 3 or 9 respectively (see Kaprekar Sequences 2 and 5, which is a Type 1a and a Type 1b sequence respectively, in Stage 3 above). If the starting number is not a multiple of 3 or 9, then all the terms in the Kaprekar sequence are also not multiples of 3 or 9 respectively (see Kaprekar Sequence 1, which is a Type 2 sequence, in Stage 3 above). [Non-trivial; true]

Conjecture 8: If the starting number of a Kaprekar sequence is a multiple of 3 but not a multiple of 9, the digital roots of the differences between consecutive terms will repeat with a period of 2, i.e. they will alternate between 3 and 6. If the starting number is a multiple of 9, then the digital roots of the differences between consecutive terms of the Kaprekar sequence will always be equal to 9. If the starting number is not a multiple of 3 or 9, then the digital roots of the differences between consecutive terms of the Kaprekar sequence will repeat with a period of 6, i.e. they will alternate among 1, 2, 4, 8, 7 and 5 in this order. [Non-trivial]

Conjecture 9: There are infinitely many self numbers. [Non-trivial; true]

Stage 5: Justifying (J)

- The following shows how to prove or refute the above conjectures. Secondary 2 students are not expected to produce the more difficult proofs, such as the formal proof for Conjecture 6 and the proof for Conjecture 9.

Conjecture 1: The first term of a Kaprekar sequence cannot be 0, or else, every term in the sequence will just be equal to 0. If the first term is negative, there is a slight complication, which can be investigated at a later stage. So assume that the first term is positive for a Kaprekar sequence, which is the usual definition. The following shows two proofs.

Non-Proof Argument:

Since the digits of the first term in a Kaprekar sequence are always non-negative and at least one of the digits is always positive (namely, the first digit), then the sum of the digits of a term will always be positive.
Therefore, the next term, which is the preceding term plus the sum of its digits, will always be greater than the preceding term, so every Kaprekar sequence is an increasing sequence.

Formal Proof:

Let the first term in a Kaprekar sequence be x and its digits be a_1, a_2, a_3, … a_n, where a_i is the i-th digit of the term.
Then $a_1 > 0$ and $a_i \geq 0$ for all $i \geq 2$.
Thus the next term $x + a_1 + a_2 + a_3 + \ldots + a_n$ is always greater than x since $a_1 > 0$ and $a_i \geq 0$ for all $i \geq 2$.
Therefore, the next term will always be greater than the preceding term, so every Kaprekar sequence is an increasing sequence.

Conjecture 2: By systematic listing of Kaprekar sequences beginning with 1 to 10, it could be shown that there are self numbers like 1, 3, 5, 7 and 9 which will never appear as the second or subsequent terms in any Kaprekar sequence since it is an increasing sequence.

Conjecture 3: Starting with 1 as a self number, all other one-digit terms will always be even because odd (one-digit number) + odd (sum of digits of the one-digit number) = even; and then even (one-digit number) + even (sum of digits of the one-digit number) = even.
Therefore, 3 will have to be a self number, and similarly for all the odd one-digit numbers. Hence, these numbers will never appear as the second or subsequent terms in any Kaprekar sequence.

Conjecture 4: By exhaustive listing, the difference between consecutive one-digit self numbers 1, 3, 5, 7, 9 is always 2.

Conjecture 5: By exhaustive listing, the difference between the last one-digit self number 9 and the first two-digit self number 20; and the difference between consecutive two-digit self numbers (i.e. 20, 31, 42, 53, 64, 75, 86, 97) is always 11.

Conjecture 6: By exhaustive listing, a 1-digit or a 2-digit number in a Kaprekar sequence will never appear in another Kaprekar sequence with a different self number. By using a counter example, a 3-digit number can appear in two Kaprekar sequences with different self numbers, e.g. 107 (see Kaprekar Sequences 1 and 4 in Stage 3 above).

Formal Proof (beyond most Secondary 2 students):

Suppose T_n is the *first* 1-digit (or 2-digit) number in a Kaprekar sequence that is also equal to K_m in another Kaprekar sequence with a different self number.
Then the term before T_n, i.e. T_{n-1}, will not be equal to the term before K_m, i.e. K_{m-1}, or else it will contradict that T_n is the *first* 1-digit (or 2-digit) number to be equal to K_m in another sequence.
In other words, we need to prove that if $T_{n-1} \neq K_{m-1}$, then T_n will never be equal to K_m.
Suppose $T_{n-1} \neq K_{m-1}$. Then there are three cases.

Case 1: Suppose T_{n-1} and K_{m-1} are both 1-digit numbers.
Then $T_n = T_{n-1} + T_{n-1} = 2T_{n-1}$ and $K_m = K_{m-1} + K_{m-1} = 2K_{m-1}$.
If $T_{n-1} \neq K_{m-1}$, then $2T_{n-1} \neq 2K_{m-1}$, so $T_n \neq K_m$.

Case 2: Suppose T_{n-1} and K_{m-1} are both 2-digit numbers.
Let the tens digit and ones digit of T_{n-1} be x_2 and x_1 respectively.
Then $T_{n-1} = 10x_2 + x_1$ and so $T_n = T_{n-1} + x_2 + x_1 = 11x_2 + 2x_1$.
Similarly, $K_{m-1} = 10y_2 + y_1$ and $K_m = 11y_2 + 2y_1$, where y_2 and y_1 are the tens digit and ones digit of K_{m-1} respectively.
If $T_n = K_m$, $11x_2 + 2x_1 = 11y_2 + 2y_1$, i.e. $11(x_2 - y_2) = 2(y_1 - x_1)$.
Since $y_1 - x_1 < 10$, then $y_1 - x_1$ does not divide 11.
Thus the equation has no solution, unless $x_2 - y_2 = y_1 - x_1 = 0$.
But this will mean $x_2 = y_2$ and $x_1 = y_1$, i.e. $T_{n-1} = K_{m-1}$, a contradiction. Therefore, if $T_{n-1} \neq K_{m-1}$, then $T_n \neq K_m$.

Case 3: Suppose T_{n-1} is a 1-digit number and K_{m-1} is a 2-digit number (or vice versa).
Then $T_{n-1} = x_1$ and $T_n = 2x_1$, where x_1 is the ones digit of T_{n-1}; and $K_{m-1} = 10y_2 + y_1$ and $K_m = 11y_2 + 2y_1$, where y_2 and y_1 are the tens digit and ones digit of K_{m-1} respectively.

If $T_n = K_m$, then $2x_1 = 11y_2 + 2y_1$, i.e. $2(x_1 - y_1) = 11y_2$.
Since $x_1 - y_1 < 10$, then $x_1 - y_1$ does not divide 11.
Thus the equation has no solution, unless $x_1 - y_1 = y_2 = 0$.
But this will mean that K_{m-1} is now a 1-digit number, and that $T_{n-1} = x_1 = y_1 = K_{m-1}$, a contradiction.
Therefore, if $T_{n-1} \neq K_{m-1}$, then $T_n \neq K_m$.

Hence, a 1-digit or a 2-digit number in a Kaprekar sequence will never appear in another Kaprekar sequence with a different self number.
We will now illustrate why the proof does not work for a 3-digit number by considering only one case.

Case 4: Suppose T_{n-1} is a 2-digit number and K_{m-1} is a 3-digit number (or vice versa).
Then $T_{n-1} = 10x_2 + x_1$ and $T_n = 11x_2 + 2x_1$, where x_2 and x_1 are the tens digit and ones digit of T_{n-1} respectively; and
$K_{m-1} = 100y_3 + 10y_2 + y_1$ and $K_m = 101y_3 + 11y_2 + 2y_1$, where y_3, y_2 and y_1 are the hundreds digit, tens digit and ones digit of K_{m-1} respectively.
If $T_n = K_m$, then $11x_2 + 2x_1 = 101y_3 + 11y_2 + 2y_1$
i.e. $101y_3 = 11(x_2 - y_2) + 2(x_1 - y_1)$.
Unlike the equation in Cases 1-3, this equation can be satisfied if, e.g. $x_2 = 9$, $x_1 = 4$, $y_3 = 1$, $y_2 = 0$ and $y_1 = 3$.
In other words, if $T_{n-1} = 94$ and $K_{m-1} = 103$, then $T_n = K_m = 107$.
Of course, there are other solutions for the equation.

Conjecture 7: If a number is a multiple of 3 or 9, then the sum of its digits is also a multiple of 3 or 9 respectively (the proof of this statement may be beyond the ability of most Secondary 2 students). Moreover, the sum of two multiples of 3 or 9 is also a multiple of 3 or 9 respectively.
Therefore, if the starting number is a multiple of 3 or 9, then all the terms in the Kaprekar sequence are also multiples of 3 or 9 respectively.
If a number is not a multiple of 3 or 9, then the sum of its digits is also not a multiple of 3 or 9 respectively. Moreover, the sum of two non-multiples of 3 or 9 is also a non-multiple of 3 or 9 respectively.
Therefore, if the starting number is not a multiple of 3 or 9, all the terms in the Kaprekar sequence are also not multiples of 3 or 9 respectively.

Conjecture 8: I do not know how to prove or refute this conjecture. This is a good example to illustrate that it is not possible for the teacher to know everything in mathematical investigation.

Conjecture 9: (Beyond the ability of Secondary 2 students to prove)
The following is a recurrence formula to generate some self numbers:

$$C_k = 8\left(10^{k-1}\right) + C_{k-1} + 8 \text{, where } C_1 = 9 \text{ (Wikipedia, 2012).}$$

The existence of such a recurrence formula suggests that there are infinitely many self numbers. Of course, one needs to prove that the recurrence formula works, which is beyond me. This is another good example to illustrate that it is not possible for the teacher to know everything in mathematical investigation.

Stage 6: Generalising (G)

- Conjectures 2-5 are not general results while Conjectures 1,6-9 will lead to a generalisation if they are proven.

Stage 7: Checking (R)

- Students need to check:
  - all calculations for repeating the process are correct;
  - the arguments used in the proving of conjectures are sound.
- Students have three possibilities to choose from at this juncture:
  - they can search for more patterns without extending the task, e.g. if they have found and justified Conjecture 1, they can search for more patterns; or they could pose the specific problems listed in Stage 2 (go to Stage 2);
  - they can extend the task by changing the given (go to Stage 8);
  - they can stop, i.e. finish the investigation.

Stage 8: Extension (E)

- Students are not expected to extend the task within the 30 minutes given for the test because there are many underlying patterns to observe, although some students may extend the task. Moreover, extension of a Type A task usually ends up with a completely new task with entirely different patterns.
- The following shows a list of possible extensions for longer investigation:
  - Add the product of its digits to the number.
  - Add the sum of the squares of its digits to the number.
  - Multiply the number by the sum of its digits.
  - Multiply the number by the product of its digits.

Task Analysis for Posttest Task 2 (Sausage)

Posttest Investigative Task 2: Sausages I need to cut 12 identical sausages so that I can share them equally among 18 people. Investigate.

Stage 1: Understanding the Task (U)

- Students should highlight key words such as 'identical' and 'equally'.
- Possible mistakes or problems in understanding the task:
  - need to subscribe to the assumption that it is possible to cut a sausage into three or more equal parts despite its rounded ends (or else the problem cannot be solved);
  - need to subscribe to the hidden assumption that it does not matter whether a person gets a rounded end or a middle portion with no rounded ends (or else the problem cannot be solved).

Stage 2: Problem Posing (P)

- Students need to pose specific problems to solve. The following shows a list of possible specific problems, which have been classified as trivial or non-trivial (the classification has passed the inter-coder reliability test as discussed in Section 5.4). P5 is the intended problem (see Section 2.2.3b for a brief explanation of intended problems).

  - *Problem 1 (P1):* Find how to cut the 12 identical sausages to share them equally among the 18 people. [Trivial]

  - *Problem 2 (P2):* Find the amount of sausages each person will receive when the 12 identical sausages are shared equally among the 18 people. [Trivial]

  - *Problem 3 (P3):* Find the number of cuts needed to share the 12 identical sausages equally among the 18 people. [Non-Trivial]

  - *Problem 4 (P4):* Find a few methods to cut the 12 identical sausages to share them equally among the 18 people. [Non-Trivial]

  - *Problem 5 (P5):* Find the least number of cuts needed to share the 12 identical sausages equally among the 18 people. [Non-Trivial; Intended Problem]

Stage 3: Specialising and Using Other Heuristics (S/H)

- Students should use other heuristics, such as reasoning, to solve the problems posed in Stage 2. They should consider various methods to cut the sausages as shown below:

  - *Cutting Method A (Usual Method):* Cut each of the 12 sausages into three equal parts. Then each person gets two parts. Total number of cuts = $12 \times 2 = 24$.

  - *Cutting Method B (Shortest Method):* Cut each sausage once at the 2/3-mark to divide it into two parts: 2/3 and 1/3 of a sausage respectively. Then each person either gets $1 \times 2/3$ of a sausage, or $2 \times 1/3$ of a sausage. Total number of cuts = 12.

– *Cutting Method C (Long Method)*[39]*:* Cut each sausage into 18 equal parts. Then each person receives one part from each of the 12 sausages, i.e. a total of 12 parts. Total number of cuts = 12 × 18 = 216.

Stage 4: Conjecturing (C)

- Students can solve Problems 1-4 using either Cutting Method A, B or C without forming any conjecture.
- Students need to form a conjecture for Problem 5 that Cutting Method B will give the least number of cuts. This conjecture is non-trivial.

Stage 5: Justifying (J)

- For Problem 5, students need to justify the conjecture that Cutting Method B will give the least number of cuts. They can use a non-proof argument:

  Proof of Conjecture that Cutting Method B will give the least number of cuts

  To share 12 identical sausages equally among 18 people, each person will receive 2/3 of a sausage. This means that there is a need to make at least one cut for each sausage to get 2/3 of a sausage. Since Method B requires exactly one cut for each sausage, then the minimum number of cuts for each sausage is 1, and the least number of cuts for the 12 sausages will be 12. Therefore, Method B will give the least number of cuts.

Stage 6: Generalising (G)

- The solutions of Problems 1-5 are not general results. Generalisation can take place later if the students extend the task to generalise (see Stage 8).

Stage 7: Checking (R)

- Students need to check:
  – the numbers of cuts are counted correctly
  – the argument used in the proving of the conjecture is sound.
- Students have three possibilities to choose from at this juncture:
  – they can pose more problems to solve without extending the task, e.g. if they have posed only Problem 1, they can now pose Problem 2, 3, 4 or 5 to solve (go to Stage 2)
  – they can extend the task by changing the given (go to Stage 8)
  – they can stop, i.e. finish the investigation, although for a Type B task like this task, they are expected to extend to generalise.

Stage 8: Extension (E)

- Students need to extend the task by changing the given in order to generalise. The following shows a list of possible problems to extend, which have been classified as trivial or non-trivial (the classification has passed the inter-coder reliability test as discussed in Section 5.4). E2 is the intended extension.

[39] There is no longest cutting method because the number of parts each of the 12 sausages can be cut, so that they can be shared equally among 18 people, can be any multiple of 18, although there is a practical limit as to how small a sausage can be cut. Some students in the present study actually used Cutting Method C.

- *Extension 1 (E1):* Find the amount of sausages that each person will receive when *n* identical sausages are shared equally among *m* people. [Trivial]

- *Extension 2 (E2):* Find the least number of cuts needed to share *n* identical sausages equally among *m* people. [Non-Trivial; Intended Extension]

- Students can also extend the task by changing other variables, such as the size and shape of the sausages (if it makes any difference), and the context of the task.

Stage 3: Specialising and Using Other Heuristics (S/H)

Since students are expected to extend this task by changing the given, then the task analysis will continue for another cycle to include some other outcomes.

- Students should go back to Stage 3 to specialise or use other heuristics, such as reasoning, to solve the extension posed in Stage 8.
- For E1, it can be solved by reasoning that the amount of sausages that each person will receive is n / m when *n* identical sausages are shared equally among *m* people.
- For E2, students should specialise by trying different values for *n* and *m* in order to find a general formula for the least number of cuts needed to share *n* identical sausages equally among *m* people. But for each case of *n* and *m* that they investigate, they still have to use other heuristics to find a cutting method that will produce the least number of cuts.
- The following tables show the least number of cuts for two cases only:
  - $n = 12$ sausages with different values for *m* people
  - $m = 18$ people with different values for *n* sausages.

  The amount of sausages that each person will receive is also given because it helps to find the least number of cuts. The method used to cut the sausages to give the least number of cuts is not necessarily any of the three cutting methods (A, B and C) described earlier, but it depends on the values of *n* and *m*.

***n* = 12 sausages**

m	Amount of Sausage Per Person	Least No. of Cuts
1	$\frac{12}{1} = 12$	0
2	$\frac{12}{2} = 6$	0
3	$\frac{12}{3} = 4$	0
4	$\frac{12}{4} = 3$	0
5	$\frac{12}{5} = 2\frac{2}{5}$	4
6	$\frac{12}{6} = 2$	0
7	$\frac{12}{7} = 1\frac{5}{7}$	6
8	$\frac{12}{8} = \frac{3}{2} = 1\frac{1}{2}$	4
9	$\frac{12}{9} = \frac{4}{3} = 1\frac{1}{3}$	6

***m* = 18 people**

n	Amount of Sausage Per Person	Least No. of Cuts
1	$\frac{1}{18}$	17
2	$\frac{2}{18} = \frac{1}{9}$	16
3	$\frac{3}{18} = \frac{1}{6}$	15
4	$\frac{4}{18} = \frac{2}{9}$	16
5	$\frac{5}{18}$	17
6	$\frac{6}{18} = \frac{1}{3}$	12
7	$\frac{17}{18}$	17
8	$\frac{8}{18} = \frac{4}{9}$	16
9	$\frac{9}{18} = \frac{1}{2}$	9

m	Amount of Sausage Per Person	Least No. of Cuts
10	$\frac{12}{10} = \frac{6}{5} = 1\frac{1}{5}$	8
11	$\frac{12}{11} = 1\frac{1}{11}$	10
12	$\frac{12}{12} = 1$	0
13	$\frac{12}{13}$	12
14	$\frac{12}{14} = \frac{6}{7}$	12
15	$\frac{12}{15} = \frac{4}{5}$	12
16	$\frac{12}{16} = \frac{3}{4}$	12
17	$\frac{12}{17}$	16
18	$\frac{12}{18} = \frac{2}{3}$	12
19	$\frac{12}{19}$	18
20	$\frac{12}{20} = \frac{3}{5}$	16
21	$\frac{12}{21} = \frac{4}{7}$	18
22	$\frac{12}{22} = \frac{6}{11}$	20
23	$\frac{12}{23}$	22
24	$\frac{12}{24} = \frac{1}{2}$	12

n	Amount of Sausage Per Person	Least No. of Cuts
10	$\frac{10}{18} = \frac{5}{9}$	16
11	$\frac{11}{18}$	17
12	$\frac{12}{18} = \frac{2}{3}$	12
13	$\frac{13}{18}$	17
14	$\frac{14}{18} = \frac{7}{9}$	16
15	$\frac{15}{18} = \frac{5}{6}$	15
16	$\frac{16}{18} = \frac{8}{9}$	16
17	$\frac{17}{18}$	17
18	$\frac{18}{18} = 1$	0
19	$\frac{19}{18} = 1\frac{1}{18}$	17
20	$\frac{20}{18} = \frac{10}{9} = 1\frac{1}{9}$	16
21	$\frac{21}{18} = \frac{7}{6} = 1\frac{1}{6}$	15
22	$\frac{22}{18} = \frac{11}{9} = 1\frac{2}{9}$	16
23	$\frac{23}{18} = 1\frac{5}{18}$	17
24	$\frac{24}{18} = \frac{4}{3} = 1\frac{1}{3}$	12

Stage 4: Conjecturing (C)

- For E1, the result (i.e. the amount of sausages that each person will receive is n / m when n identical sausages are shared equally among m people) is trivial.
- For E2, students are expected to search for patterns in the examples that they had generated, but they are not really expected to formulate a conjecture on the formula for the least number of cuts because the pattern is not easy to discover:

$$m - \text{HCF}(m, n),$$

where m is the number of people and n is the number of sausages[40]. This conjecture is non-trivial.

[40] One student in the present study (S9) was able to formulate this conjecture just by observing patterns in his two examples, so it was not impossible to do so.

Stage 5: Justifying (J)

- For E1, the result (i.e. the amount of sausages that each person will receive is n / m when n identical sausages are shared equally among m people) is clearly true, and so there is no need to prove it.
- For E2, students are not expected to justify the conjecture formulated in the previous stage as the proof is beyond them. Nevertheless, a formal proof is shown below. However, it is still possible for students to think of the main idea behind this proof[41], although they will not be able to prove that the cuts will not coincide with the gaps between the sausages in Case 2a below.

Proof of General Formula for Least Number of Cuts

Case 1: $n \geq m$

If the no. of sausages, n, is greater than or equal to the no. of people, m, then

$$n = qm + r, \text{ for some positive integer } q \text{ and non-negative integer } r < m.$$

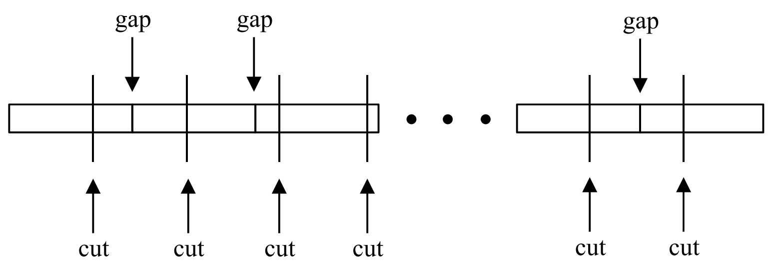
This means that we can give q whole sausages to each of the m people, with a remainder of r sausages.
If $r = 0$ (i.e. n is a multiple of m), then we are done, i.e. there is no need to cut and so the least no. of cuts = 0.
If $0 < r < m$, then we go to Case 2, where the no. of sausages is less than the no. of people.

Case 2: $n < m$

If the no. of sausages, n, is less than the no. of people, m, there are 2 scenarios.

Case 2a: $n < m$ and HCF(n,m) = 1

We are going to prove that the least number of cuts in this case is $m - 1$, where m is the no. of people. The basis of the proof is this: if one sausage is shared among m people, then the least number of cuts is $m - 1$.
If there are n sausages, arrange the sausages in a row as shown below.



If we treat the row of sausages as one sausage, then the least no. of cuts is $m - 1$, provided that the cuts do not coincide with the gaps between the sausages. If a cut coincides with a gap, this means that there will be one cut less, as there is no need to cut at the gap in the first place. So we are going to prove that the cuts will not coincide with the gaps.

[41] The main idea behind the two examples of the same student (S9) was similar to the main idea behind the proof, although he was not able to prove the conjecture.

If we treat the row of sausages as one sausage, then
the gaps will be at the $\frac{1}{n}$-mark, the $\frac{2}{n}$-mark, the $\frac{3}{n}$-mark,…, the $\frac{n-1}{n}$-mark, and
the cuts will be at the $\frac{1}{m}$-mark, the $\frac{2}{m}$-mark, the $\frac{3}{m}$-mark,…, the $\frac{m-1}{m}$-mark.

Suppose there is a cut that coincides with a gap.
Then $\frac{a}{m} = \frac{b}{n}$ for some integers a and b, where $1 \le a \le m - 1$ and $1 \le b \le n - 1$.
$\therefore\ an = mb$.
Since HCF(n,m) = 1, i.e. n and m are relatively prime, then

$$a = pm \text{ and } b = qn \text{ for some positive integers } p \text{ and } q,$$

i.e. a must be a multiple of m, and b must be a multiple of n.
But these will contradict $1 \le a \le m - 1$ and $1 \le b \le n - 1$.
Therefore, the cuts will not coincide with the gaps between the sausages, and so the least no. of cuts is $m - 1$.

Case 2b: *n* < *m* and HCF(*n*,*m*) > 1

Let HCF(n,m) = $h > 1$.
Consider sharing $\frac{n}{h}$ identical sausages equally among $\frac{m}{h}$ people.
Since HCF($\frac{n}{h}, \frac{m}{h}$) = 1, then least no. of cuts = $\frac{m}{h} - 1$ (from Case 2a).
Therefore, to share n identical sausages equally among m people,

$$\begin{aligned}\text{least no. of cuts} &= h \times \left(\frac{m}{h} - 1\right)\\ &= m - h\\ &= m - \text{HCF}(n,m) \qquad \text{[proven]}\end{aligned}$$

Note: If HCF(n,m) = 1, then we have Case 2a: least no. of cuts = $m - 1$.
If n is a multiple of m (Case 1), then HCF(n,m) = m, and so the least no. of cuts is m – HCF(n,m) = $m - m = 0$.

Stage 6: Generalising (G)

- The solutions for both E1 and E2 are general results, thus generalisation will take place if the students can solve these extensions correctly.

Stage 7: Checking (R)

- Students need to check:
  – the number of cuts for each example is counted correctly;
  – the cutting method will indeed give the least number of cuts for each example;
  – the formula for the least number of cuts will work for some other examples;
  – the argument used in the proving of the conjecture is sound.
- Students have two possibilities to choose from at this juncture:
  – they can pose more problems to extend the task by changing the given (go to Stage 8);
  – they can stop, i.e. finish the investigation.

APPENDIX F: INSTRUCTIONS FOR THINKING ALOUD

This appendix contains the instructions for thinking aloud, which were adapted from Foong (1990) with some modifications. These instructions were read to the students just before they practised thinking aloud during the first thinking-aloud practice session. During subsequent thinking-aloud practice sessions, the students were also reminded of some important points.

1. The purpose of the individual test, which will be conducted next week, is to find out how and what you investigate, and this includes what you are thinking as you investigate. But how can I know what you are thinking unless you tell me? So you need to say aloud what you are thinking as you do the investigation. This is called 'thinking aloud'.

2. You need to practise thinking aloud now so that you will be comfortable doing so during the test. This is what you need to do.

3. When you investigate, say aloud what is exactly on your mind. Sometimes, when you are working alone, do you talk to yourself by saying aloud what you are thinking? So just imagine you are alone now and talk to yourself.

4. I will give a demonstration first and then I will explain some pointers when you talk aloud. [Give a short demo using Task 2.]

5. This is what you can say aloud:
   - your thoughts,
   - your feelings,
   - your decisions,
   - your analyses,
   - your conclusions,
   - questions you ask yourself,
   - any mental operations such as addition, subtraction, etc.,
   - whatever you don't understand.

6. If you are stuck, it is ok. Just say aloud that you are stuck. But don't just stop there. Try to think what to do next and say it aloud.

7. If your working leads you nowhere, it is ok. Just say aloud and write down that you want to try something else. Do not erase or cancel your previous working. That is why you need to write down that you want to try something else, so that I know what you are doing.

8. If you visualise some picture mentally, you need to describe the picture. Other than this, do not describe or summarise what you are thinking. You need to say aloud the exact words that come into your mind as you are thinking.

9. When you read, you need to read it aloud because this is part of your thoughts.

10. When you write, you need to say aloud what you are writing. However, if you are thinking of something else while you are writing, continue to write but say aloud what you are thinking. [Give a short demo.]

11. I am just interested to find out what you are thinking. So keep on talking. Do not pause for more than 3 seconds.

12. You must also speak loudly and clearly so that I can hear what you say. Do not mumble to yourself.

13. You really have to think aloud during the test. If you are uncomfortable doing so during the test, it will definitely affect your test. So you must practise thinking aloud today.

14. Is there any question before we begin?

APPENDIX G: INSTRUCTIONS FOR INVIGILATOR DURING PRETEST AND POSTTEST

This appendix contains instructions for the invigilator for the pretest and the posttest, as well as the exact instructions to be read to the student just before the test.

Instructions for the Invigilator

1. Each student is to do two tasks. Each task takes 30 minutes. Give out one task at a time and collect back after 30 minutes.

2. Please ensure that the student uses a dark-coloured ball-point pen to write and that he or she does not use liquid paper to erase any working (see para. 4h on next page).

3. Please check occasionally the position of the student's answer script to ensure that the video cam can capture what the student is writing. At the same time, check that the video cam is still in RECORD MODE (the researcher will show you how). Otherwise, sit at the back of the classroom and do not stand too near the student all the time, or else it may affect the student's performance.

4. (a) If the student remains quiet for more than 3 seconds, please remind him or her, "Please speak out what is on your mind."
   (b) If the student talks too softly, please remind him or her, "Please speak louder."
   (c) If the student mumbles, please remind him or her, "Please talk clearly. Do not mumble."

   It is OK if your voice is recorded.

5. If the student finishes the task or the test before the time is up, do not allow him or her to start a new task or to leave. Please tell him or her, "Please investigate some more." Please do not give hints to the student or help him or her in any way.

[Please turn over]

Instructions to be read to the student

Please read the following to the student at the start of the test. Use the exact words. Do not rephrase.

1. The test consists of two tasks. You have exactly 30 minutes to do each task. The test will last one hour. You will need a calculator. Do you have one?

2. In order for the video camera to capture what you are writing:
   (a) You must put the paper that you are writing at this position [invigilator to indicate position to student].
   (b) Do not cover up what you are writing.
   (c) Sit up straight. Do not bend your head, or else it will block the video camera.
   (d) You must write bigger and clearly; do not scribble.

3. Show all your rough working and final solution in the question paper and writing paper attached behind. You can write on both sides. Write in order: if you write all over the place, we will not know where to read first.

4. You need to think aloud for the test.
   (a) Please speak loudly and clearly so that your voice can be captured by the video camera. Do not mumble.
   (b) Say aloud what you are thinking or feeling at each moment. Keep on talking. Do not pause for more than 3 seconds.
   (c) If you visualise some picture mentally, describe the picture. Other than this, do not describe or summarise what you are thinking. You need to say aloud the exact words that come into your mind as you are thinking.
   (d) When you read, read it aloud.
   (e) When you write, say aloud what you are writing. However, if you are thinking of something else while you are writing, continue to write but say aloud what you are thinking.
   (f) If you are stuck, it is ok. Just say aloud that you are stuck. But don't just stop there. Try to think what to do next and say it aloud.
   (g) If your working leads you nowhere, it is ok. Just say aloud and write down that you want to try something else. Do not erase or cancel your previous working. That is why you need to write down that you want to try something else, so that we know what you are doing.
   (h) If your working is wrong, cancel it neatly with a line across and then write your new working beside or below. Do not erase the wrong working.
   (i) You must still write down your findings instead of just saying it loud.

APPENDIX H: CHECKLIST ON INVESTIGATION PROCESSES

This appendix shows the checklist of investigation processes given to the students during the posttest to refer to.

1. Understand the Task

- Read the task statement carefully
- Try to understand the task by re-reading or rephrasing task statement
- Highlight key information by underlining or circling main points
- Visualise information by drawing diagram
- Try to understand the task by trying some examples

Monitor your own thinking:

- Do I understand all the given information in the task?
- Do I interpret the task correctly?

2. Pose Problem

- Pose the general problem of searching for any pattern
- Pose a specific problem that 'naturally follows' from task
- Pose a specific problem to generalise
- Use association to pose a problem
- Use analogy to pose a problem

Monitor your own thinking:

- What does the task require me to do? What is there to investigate?
- Is the goal or problem feasible or worth pursuing? Why? (E.g. It is a problem that 'naturally follows' the task. I can generalise to find a formula. The problem is interesting to me. The problem should be within my ability to solve.)

3. Attack the Problem

- Search for patterns by choosing systematic examples to investigate
- Draw diagram
- Guess and check
- Use reasoning
- Use algebra
- Pose an easier problem or simplify the problem (e.g. use smaller numbers)

Monitor your own thinking:

- What method or strategy can I use to solve the problem? What have I learnt before that I can apply?
- Is the plan or strategy worth pursuing? Why? (E.g. It is a reasonable plan. It seems to work.)

Monitor progress every five minutes:

- Am I going on the right track? How do I know? (E.g. The plan seems to be working. I don't know: maybe I will try for another 5 minutes.)
- I am going nowhere. Should I abandon the plan and think of a new approach, or should I try for another 5 minutes and see how it goes? Should I change to pursue another goal or problem first? Why?

4. Formulate and Test Conjectures

- Formulate a conjecture by searching for patterns
- Formulate a conjecture by using reasoning
- Refute or disprove a conjecture by looking for counter examples
- Prove a conjecture by using a non-proof argument
- Prove a conjecture by using a formal proof (e.g. algebra)

Monitor your own thinking:

- Is my conjecture always true? Why?
- Is there a reason to believe that the pattern will always continue?
- Can I come up with a more rigorous proof?

5. Look Back

- Check the whole solution
- Check answer to see if it is correct (e.g. by substituting answer back into the task)

Monitor your own thinking:

- Is my answer logical, sensible or reasonable?
- Is my method of solution efficient?
- Is there a better method of solution? Is there another method of solution?
- Is the alternative method of solution efficient? Is it better than the previous method? In what ways is it better or worse?
- Can I apply the result (e.g. answer, formula) learnt in this task to some other tasks that I have seen or solved before?
- Can I apply the method of solution learnt in this task to some other tasks that I have seen or solved before?

6. Extension

- Extend specific problems solved but still within scope of original task
- Extend original task by changing the given

Monitor your own thinking:

- Can I extend the specific problem solved to generalise? Are there other ways to extend the specific problem solved?
- Can I extend the original task by changing some given information?
- Is the extension feasible or worth pursuing? Why? (E.g. It is an extension that 'naturally follows' the task. I can generalise to find a formula. The extension is interesting to me. The extension should be within my ability to solve.)

APPENDIX I: SAMPLE CODED TRANSCRIPT FOR POSTTEST TASK 1 (KAPREKAR)

This appendix shows a coded transcript from S5 for Posttest Task 1 (Kaprekar) that is used as a sample in Chapter 4.

Legends:

- *Pause for 3 s or less is indicated by the three dots: …*
- *Student's actions and transcriber's comments are in square brackets.*
- *Different episodes are separated by a double line.*

Line	Time	Transcript	Stage Code	Bhvr Code	Remarks
01 (p.1)	00:00	Choose any number. Add the sum of its digits to the number itself to obtain a new number. Repeat this process for the new number and so forth. Investigate.	U	FR	Answer Script Page 1
02	00:11	[Teacher tells her to speak louder]	X	XO	
03	00:12	Choose any number. Add the sum of its digits to the number itself to obtain a new number. Repeat this process for the new number and so forth. Investigate.	U	RR	
04	00:23	So ... now I must, uh, try some examples first ...	U	DP	
05	00:28	For example, I choose 12 [write 12].	U	TE1	Example 1 (random): p. 1 column 1 in answer script
06	00:31	Add the sum of digits. The question says choose any number.	U	RR	
07	00:35	[Underline: any number]	U	HI	
08	00:36	So add the sum of its ... add the sum —	U	RR	
09	00:38	[Underline: sum]	U	HI	
10	00:39	— of its digits.	U	RR	
11	00:40	[Underline: digits]	U	HI	
12	00:41	So 12 [continue writing] = 1 + 2 = 3 [stop writing].	U	TE1	
13	00:45	Add the sum of its digits to the number itself to obtain a new number.	U	RR	
14	00:50	So I must add the sum of its digits to the number.	U	RT	
15	00:55	— is [continue writing] 12 + 3 = 15 [stop writing].	U	TE1	
16	00:58	So to obtain a new number —	U	RR	
17	01:00	The new number is 15.	U	TE1	
18	01:02	Repeat this process.	U	RR	
19	01:03	[Underline: Repeat this process] …	U	HI	
20	01:06	That means [write 15 + and then cancel +] I must try 15. 15 [write =, then cancel = and write :] 15 is [continue writing] 1 + 5 = 6; then 15 + 6 is 21; 21 is … 2 … 2 + 1 = 3; so 3 add, added, 3 is being added to 21; 21 + 3 = 24; 24 is, the two digits are 2 and 4, so 2 + 4 = 6; 24 + 6 = … 30 … uh, 30, so … 30 is … 3 + 0 = 3 …	U	TE1	
21	01:54	24 + [stop writing] … Twenty — … 24.	U	EM1	Minor Error 1: Should be 30, not 24
22	02:00	Repeat this process for the new number and so	U	RR	

Line	Time	Transcript	Stage Code	Bhvr Code	Remarks
		forth … Add the sum of its digits to the number itself.			
23	02:08	So now is … um, 30 [cancel 24 +].	U	ED1	Discover Minor Error 1
24	02:11	So is [continue writing] 30 + 3 = 33; 33 is 3 + 3 = 6; so use 6 to plus 33 is 39 [stop writing] 39 —	S	TE1	Transition from TE to understand task (U) to TE to specialize (S) implied from Line 28 that she had not found a pattern
25	02:26	[Teacher tells her to speak louder]	X	XO	
26	02:27	39, 39 is, um, the two digits are 3 and 9; 3 and 9 equal to 12; so add 12 to 39 equal to 51; so 51 is … uh, 5 + 1 = 6.	S	TE1	
27	02:46	51 + 6 is 56 [stop writing] …	S	EM2	Major Error 2: Should be 57, not 56 (stuck for quite long without finding pattern as 56 is not divisible by 3)
28	02:51	So I never find the pattern yet …	C	SP	
29	02:54	So … so … I think I go … I think I go nowhere … [write the word nowhere at the bottom of Example 1].	C	MP	
30	03:02	So I try another number ... uh, example is 23, right.	S	DP	
31	03:06	[Start writing] 23 is, the two digits are 2 and 3 equal to 5; 23 + 5 = 28; 28 is 2 + 8 = … 10; then, uh, 28 + 10 = 39, 38 ... 38 is 3 +, 3 + 8 = 11; use 11 to plus 38 equal to 49; 49 is, uh, 4 + 9 = 13; 49 + 13 = 62; 62 is 6 + 2 = 8; use 8 to plus 62 is equal to 70; 70, the two digits are 7 and 0, uh, equal to 7.	S	TE2	Example 2 (random): p. 1 column 2 in answer script
32	04:03	77 + 7 = 84.	S	EM3	Major Error 3: Should be 70 + 7, not 77 + 7 (created numbers that are divisible by 3 when they should not be, which confused her)
33	04:08	84 is … 84, the two digits are 8 and 4, so is equal to 12; 84 + 12 = 96; 96 are 9 + 6 = 15; use 96 + 15 = 111 [stop writing] …	S	TE2	
34	04:33	So … now … uh … now the digits are from two digits to three digits … so … um —	C	MA	Aware she first obtain 3-digit number, think it is wrong, re-read

Line	Time	Transcript	Stage Code	Bhvr Code	Remarks
					task in next line, but find nothing wrong
35	04:45	Repeat this process for the new number and so forth ... Repeat this process.	C	RR	
36	04:50	Now the — this number [start circling 28, 38, 49 until 111] is what I obtain ... is 28, 38, 49 … 62 ... uh, 70, 84, 96 ... um, 111 [stop circling] ... So is ... no ... no … pattern ...	C	SP	
37	05:14	Add the sum of its digits to the number itself to obtain a new number. The question says add the sum of its digits to the number itself ... to obtain a new number.	C	RR	
38	05:28	Yes, I think I am, I am in the ... right way.	C	MP	
39	05:31	But … now I can't find the pattern yet.	C	SP	
40	05:35	So what should I do is to ... what I should do is to ... um ...	X	XH	
41	05:42	What I should do is to try ... uh, continue to try the numbers.	S	DP	
42	05:48	[Continue writing] 111 is ... the three digits are 1 + 1 + 1 = 3; so 111 + 3 = 114; 114, the three digits are 1 + 1 + 4 = 6 … so use 6 to plus 114 is 120 [stop writing] ... 1 ... 120 is [write: 120 → and continue writing] 1 + 2 + 0 = 3; 1 ... 120 + 3 = 123 [stop writing] ...	S	TE2	Continue Example 2: p. 1 column 2 in answer script
43	06:27	123 … But ... I can't find the pattern ...	C	SP	
44	06:34	So ... investigate ... what?	X	XH	
45	06:37	[Read from given checklist] Understand the task first [stop reading].	X	XC	
46	06:39	This I understand already.	X	MP	
47	06:41	[Continue to read from checklist] Then set goal or pose problem. Decide to search for patterns. Pose a problem that naturally follows from task [stop reading].	X	XC	
48	06:50	I want to find the ... I want to find ...	P	PT	
49	06:54	My task is … [Start writing: Task: Find pattern] My task is to find, uh ... find the pattern for these numbers ... yes, find the pattern [stop writing] Find pattern … [continue writing: for these numbers] for … uh, for these numbers [stop writing].	P	PP0	Posed General Problem: Search for any pattern (p. 1 column 3 in answer script)
50	07:11	But until now I can't find the pattern yet. So ...	X	MP	
51	07:16	[Read from given checklist] Search for patterns by choosing systematic examples to investigate. Check, guess and check. Draw diagram [stop reading]	X	XC	
52	07:24	Of course cannot draw.	X	MF	
53	07:26	So I use ... um ... I find the … sum of digits first.	P	PP1	Specific Problem 1 (trivial): Look for pattern in sums of digits
54	07:32	— is [start underlining the following sums of digits in Example 2] 5, 10, 11 [stop underlining] ... uh ... [point pen at following numbers] 5, 10, 11 [continue underlining] 13, 8, 7, 12, 15, 3, 6 … 3 [stop underlining] ... but ... um ...	C	SP	Refer to Example 2

Line	Time	Transcript	Stage Code	Bhvr Code	Remarks
55	07:51	To obtain a new number. Repeat this process for the new number and so forth ... To obtain a new number —	C	RR	
56	07:59	I obtain already. Uh ... but ... now my task is to find a pattern for these numbers [underline the task in p. 1] ...	C	MP	
57	08:07	So what I should try is ... now I find ... now I find ...	X	XH	
58	08:16	What's the pattern for these numbers? ... Is it uh ... um ...? This one cannot [point pen at the new numbers 38, 49 and 62 in Example 2] this one cannot ... So ... um ...	C	SP	Refer to Example 2
59	08:29	[Point pen at following sums of digits in Example 1 starting from the last 6 in 51 → 5 + 1 = 6] 6, 12, 6, 3 ... 6 ... 3 ... 6, 3 [this is the first 3 in 12 + 1 + 2 = 3]. [Point pen at the first two sums again] 3, 6 [stop pointing] ... um, this one is 3 [circle first 3] ... um ... [start underlining following numbers starting from first 3] 3, 6, 3, 6, 3, 6 ... um ... 12, 6 [stop underlining] ... 12, 6 [cancel the word: nowhere] ... so ... [point pen at last two sums] 12, 6 [stop pointing].	C	OP1	Refer to Example 1 to observe Pattern 1 (non-trivial / correct): Sums of digits divisible by 3 (implied from Line 61 when she rejected this pattern)
60	08:57	So I now try 56 [should be 57: previous Error 2] ... [Start writing: 56 → 5 + 6 = 11] 56 is 5 + 6.	S	TE1	Continue Example 1
61	09:03	So cannot, 11 [should be 12: result of previous Error 2] [stop writing]. 11 cannot be divided by 3 ... But others can be divided by 3 ...	C	RP1	Reject Pattern 1 wrongly (based on Major Error 2 but pattern is actually correct)
62	09:10	So ... um ... you see this one is ... 28 [point to first 28 in Example 2] 28, um ... [point to the following sums of digits in Example 2] 5 ... 5, 10, 11 [stop pointing] ... no pattern what ... But ... so —	C	SP	Refer to Example 2
63	09:29	[Read from given checklist] Attack the problem is search for patterns by choosing systematic examples to investigate. Draw diagram [stop reading].	X	XC	
64	09:39	Draw diagram cannot. So guess and check also cannot.	X	MF	Counted as 1×MF in Lines 64, 67, 69 since this is same instance
65	09:41	[Continue reading from checklist] Use deductive or logical reasoning. Use formal proof [stop reading].	X	XC	
66	09:47	Formal proof also cannot. For this problem, formal proof also cannot.	X	MF	
67	09:54	[Continue reading from checklist] Pose, pose an easier problem or simplify the problem. Pose an easier problem or simplify the problem [stop reading].	X	XC	
68	10:03	This is ... uh, an easier problem already. So ... uh	X	MF	

Line	Time	Transcript	Stage Code	Bhvr Code	Remarks
		—			
69	10:09	[Continue reading from checklist] Pose a related problem [stop reading].	X	XC	
70	10:11	Now what I want to find is a pattern for these numbers ...	P	PT	
71	10:15	Hmm, pattern for these numbers is ...	P	XH	
72	10:19	To obtain a new number. Repeat this process for the new number.	P	RR	
73	10:23	So I try ... I try the difference between these numbers.	P	PP2	Specific Problem 2 (trivial): Look for pattern in difference between consecutive sums of digits
74	10:29	[Draw an arc between consecutive sums of digits in Example 1, starting from 12 = 1 + 2 = 3 and 15: 1 + 5 = 6, and write corresponding difference] — is 3, 3, 3, 3, 3 and ... this is 6, then 6, then 5 [stop drawing and writing] ...	C	SP	Refer to Example 1
75	10:43	Eh, this is …	C	MA	The number 5 in previous line does not fit pattern: cause her to spot Error 2
76	10:46	57 [change 56 to 57]. [10:48] 5 + 7 [amend 6 to 7] = 12 [amend 11 to 12]. So is 6 [amend 5 to 6] …	C	ED2	Key Moment 1: Discover Major Error 2
77	10:52	Ok [start writing] 56 … uh ... 57 + 12 = … 69; 69 is ... 15 [stop writing].	S	TE1	Continue Example 1
78	11:06	15 also can be divided by 3, but the difference between 6 and 15 is 9.	C	SP	
79	11:11	Then I ... use [continue writing] 57 + 15 = … 72 [stop writing]	S	TE1	
80	11:20	— also can be divided by 3 ... This ... the ... but is it always the same?	C	OP1	Back to Pattern 1
81	11:28	I try ... I continue to try this one.	C	DP	
82	11:32	This one I find [draw a line below Example 1] I — I think that the sum of digits can be divided by ... can be divided by 3 [write below Example 1: divided by] ...	C	OP1	Still Pattern 1
83	11:44	But I try another, another number first.	C	DP	
84	11:47	23 [Example 2 starting number] is 2 + 3, 5 ... 28 ... 28 … but ... um ...	C	SP	Refer to Example 2
85	11:55	This one ... of course cannot lah.	C	RP1	Reject Pattern 1 again
86	11:57	Then ... 49 is ... 49 is 4 + 9 = 13. So ... um ... 49 + 13 = 62. No pattern.	C	SP	Refer to Example 2
87	12:16	I try … another number.	S	DP	
88	12:19	Choose any number ...	S	RR	
89	12:22	Now is two-digit number. I still try two-digit number. This one [refer to Example 2 which now has three-digit numbers] ... this one [draw a line below Example 2 and put a cross] ... I don't calculate yet.	S	DP	

Line	Time	Transcript	Stage Code	Bhvr Code	Remarks
90	12:30	So I try ... um, 3 ... I try [start writing] 36 ... 36 is 3 + 6 = 9; so 36 + 9 = 45 [stop writing].	S	TE3	Example 3 (purposeful choice of 2-digit number): p. 1 column 3
91	12:48	45 can be divided by 3.	C	OP2a	Pattern 2a (non-trivial / correct): Each term of the sequence is divisible by 3 (true if starting number is divisible by 3)
92	12:50	So [continue writing] 4 + [stop writing] ... so 4 + ... um [continue writing] 5 = 9; 45 + 9 = 54 [stop writing].	S	TE3	
93	13:01	These numbers … can be divided by 9.	C	OP3	Pattern 3 (non-trivial / correct): Each term of the sequence is divisible by 9 (true if starting number is divisible by 9)
94	13:06	[Continue writing] 5 + 4 = 9; 54 + 9 = 63; 6 + 3 = 9 [stop writing] …	S	TE3	
95	13:18	I find that all the … sums of the digits [underline all these sums] are 9 ...	C	OP3	Still Pattern 3 because she is still looking at the terms (see Lines 97, 99 and 102)
96	13:25	So [continue writing] 63 + 9 = 72 [stop writing].	S	TE3	
97	13:29	72 can be divided by 9 —	C	OP3	Still Pattern 3
98	13:31	— is [continue writing] 7 + 2 = 9; 72 + 9 = 81 [stop writing and underline the previous 9].	S	TE3	
99	13:39	These also can be divided by 9 ... It's 9 itself.	C	OP3	Still Pattern 3
100	13:44	So [continue writing] 81 is, um ... 8 + 1 = 9; 81 + 9 = ... 81 + 9 = 90 [stop writing].	S	TE3	
101	13:56	Also can be divided by 9 [draw a line below Example 3] ...	C	SP	
102	14:00	So ... now this one can be [start writing below Example 3: divided by 9] divided by ... divided by 9 [stop writing] ...	C	OP3	Still Pattern 3
103	14:10	Now I try ... now I should try ... another number ...	S	DP	
104	14:18	Example is ... 3 ... um ... 4 ... no, is ...	X	XH	
105	14:26	84 ... 84 is ... [point pen at 12 in following statement] 84 is 8 + 4 = 12, so [point pen at following numbers] 96, um, 15 ... 111 and 3 and 114 and 6 and ... 6 and 120 and 3 and 123 [stop pointing] ...	C	SP	Refer to Example 2
106	14:50	So start ... start from here [draw a line after 70 → 7 + 0 = 7 and a vertical line all the way to the last statement in Example 2] ... start from 70 + 7	C	OP2b	Pattern 2b (non-trivial / wrong): Terms

Line	Time	Transcript	Stage Code	Bhvr Code	Remarks
		... the rest are ... all can be divided by 3 [write beside vertical line: divided by 3]. So ... my thinking is that ... um ... um, my thinking is that whatever which number it is ... so, um ... if continue to use this method to calculate, finally the number can be divided by 3 ...			of sequence divisible by 3 eventually (found this pattern as a result of her Major Error 3)
107 (p.2)	15:27	So I, I try ... I try another number is ... I try another number. [Turn to new p. 2] ... I try … another example [write: Example:] ... My example is that ... use ...	S	DP	Answer Script Page 2
108	15:50	[Start writing] 47 is 4 + 7 = 11; 47 + 11 = 58 [she reads as 59]; 58 [she reads as 59] is 5 + 8 = 13; and fifty, fifty, 5 + 8 = 13; 58 + 13 is ... 71; 7 + 1 = 8; then 71 + 8 = 79; 7 + 9 = 16; 79 + 16 = ... 95 ... 9 + 5 = 14.	S	TE4	Example 4 (purposeful choice of 2-digit number): p. 2 top part column 1
109	16:32	So 79 + 14 = 93.	S	EM4	Major Error 4: Should be 95, not 79 **(**confuse her as sum of digits 12 in next line is divisible by 3 when it should not)
110	16:36	9 + 3 = 12 [stop writing].	S	TE4	
111	16:40	Oh, now, now the 12 can be divided by 3. So 9 + 3 = 12 ... 12 ... use ... um ...	C	SP	
112	16:52	[Start writing] 79 + ... 12 = 91; 9 + 1 = 10; so 91 + 10 = ... 91 + 10 is 101 ... 1, 0, 101 is 1 + 1 = 2; 101 + 2 = 103; 1 + 3 = 4; 103 + 4 is 107 ... 1 + 7 is 8; so ... so now is ... um, 107 + 8 is 115 [stop writing].	S	TE4	Continue Example 4 (same Error 4: should be 93 + 12, not 79 + 12)
113 (p.1)	17:43	These cannot be divided by 3. But just now [turn back to p. 1] the number is ... 23 [Example 2 starting number]; 23 can be ... twenty ...	C	SP	Refer to Example 4 and Example 2
114	17:53	Oh no, this one is wrong.	C	RP2b	Reject Pattern 2b
115	17:55	70 ... [point pen at following working] 28... 28 is 2 + 8 = 10, so 28 + 10 is 38 [she reads as 39] [stop pointing pen] ... But ... here [use pen to circle first half of Example 2] I can't find a pattern.	C	SP	Refer to Example 2
116 (p.2)	18:15	[Turn back to p. 2] So my ... so my findings is that [write: Finds:] ... if the original number [flip to look at p. 1] can be divided by 3, then ... then the ... rest also can be divided by 3 ... But is it always true? ...	C	OP2c	Pattern 2c (non-trivial / correct): based on Example 3 and modified from Pattern 2: If starting number is divisible by 3, each term of sequence is divisible by 3

Line	Time	Transcript	Stage Code	Bhvr Code	Remarks
117 (p.1)	18:39	I continue to [turn to p. 1] try this example. Um [continue writing for Example 3] 9 + 0 is 9; 90 + 9 = 99; 9 + 9 = 18.	S	TE3	Continue Example 3
118	18:53	So 90 + 18 is 108 [stop writing].	S	EM5	Minor Error 5: Should be 99, not 90 (but numbers still divisible by 9, so does not affect pattern)
119	18:59	Yes, this can be divided by ... 9.	C	OP3	Back to Pattern 3
120 (p.2)	19:03	So ... if ... if the — [Turn to p. 2] One of my findings is that ... [continue writing after the word: Finds:] if the original number can be divided by 3 … or 9, if the original number can be divided [she writes dived instead of divided] by 3 or 9, by 3 or 9 ... then the, then the new numbers obtained also can be divided by 3 or 9, also can be divided by 3 or 9 [stop writing].	C	FC1	Conjecture 1 (non-trivial / correct): combination of Patterns 2c and 3: If starting number is divisible by 3 or 9, each term of sequence is divisible by 3 or 9 respectively (p. 2 top part column 2)
121	20:04	But what if ... it's ... 6?	P	PP3/ OP4	Specific Problem 3 (non-trivial) with Pattern 4 (trivial / wrong): If starting number is divisible by 6, will each term of sequence be divisible by 6?
122	20:08	[Flip to p. 1] I try 6 [flip back to p. 2] ... For 6 [write: 6:] is ... 6 [write another 6] itself, no, no, it's 12 [cancel 6 and write 12] ... 12 is ... [start writing] 1 + 2 [she reads as 3] ... eh, equal to 3; 12 + 3 = 15 [stop writing].	S	TE5	Example 5 (purposeful choice of number divisible by 6): p. 2 bottom part column 1
123	20:32	15 cannot be divided by 6.	C	RP4	Solve Specific Problem 3: Reject Pattern 4
124	20:35	So ... if it is 3 or 9, the new numbers obtained also can be divided by 3 or 9 ...	X	XR	Re-read Conjecture 1 but miss out some words
125	20:43	Then what if the sum of the digits is 2? [she thinks that sum of digits is divisible by 2 means	P	PP4/ OP5	Specific Problem 4

Line	Time	Transcript	Stage Code	Bhvr Code	Remarks
		number is divisible by 2 (just like divisibility tests for 3 and 9), which is why she tried 11 below]			(non-trivial) with Pattern 5 (trivial / wrong): If sum of digits of starting number is 2, will each term of sequence be divisible by 2?
126	20:47	I try 11 ... [Start writing] 11 is 1 + 1 = 2; then, um, 11 + 2 = ... 13; 13 is ... 1 + 3 = ... 1 +, 1 + 3 = 4.	S	TE6	Example 6 (purposeful choice of number whose sum of digits is 2): p. 2 bottom part column 2
127	21:06	11 + 4 = ... 15 [stop writing].	S	EM6	Minor Error 6: Should be 13 + 4, not 11 + 4 (pattern not true anyway)
128	21:12	My finding is that all the numbers obtained is odd number [write: odd number] odd number.	C	OP6	Pattern 6 (trivial / wrong): if starting number is odd, each term of the sequence will be odd (no longer Specific Problem 4)
129	21:21	What if I try 20?	S	DP	Decide to try even number to see if it works like OP6
130	21:23	[Write 20 and continue writing] is … 2 + 0 = 2; 20 + 2 [stop writing] ... 20 + 2 = ... 22 [write: = 22].	S	TE7	Example 7 (purposeful choice of even number): p. 2 bottom part column 3
131	21:35	So if the start number is … odd number, then the final number also be odd number ... If the … start number is even number [write: even number] even number, the rest will be even number ...	C	FC2	Conjecture 2 (trivial / wrong): if starting number is odd or even, each term of the sequence will be odd or even
132	21:53	But can I prove my, this finding? ... If … Can I use algebra to try … this example? ...	J	TP	
133	22:02	Hmm, the two digits are *a* and *b* [write: $\overline{ab}$] ... *a*	J	AL	

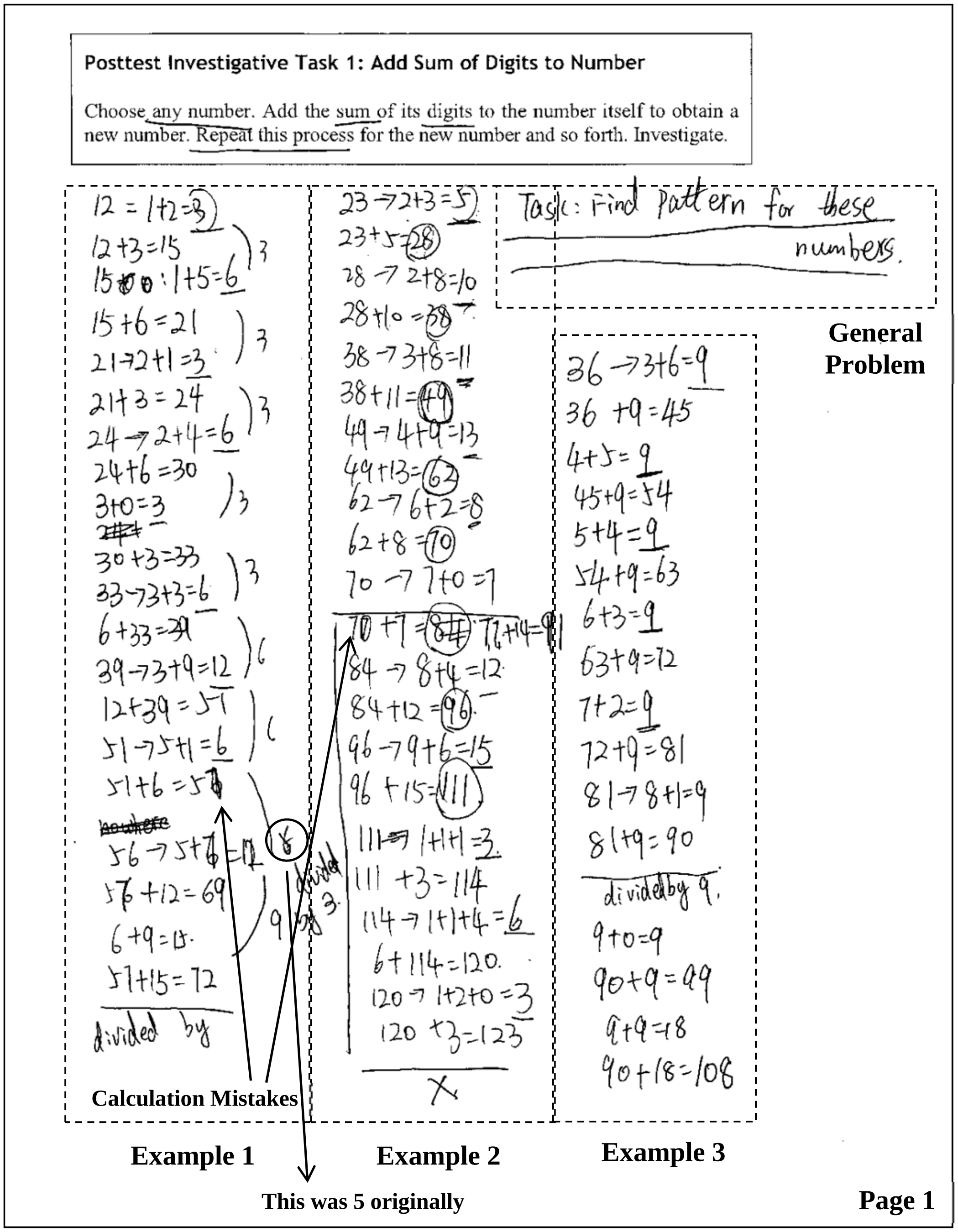
Line	Time	Transcript	Stage Code	Bhvr Code	Remarks
		and *b*, so now I use $\overline{ab}$ + *a* + *b* ... so $\overline{ab}$ [continue writing] + *a* + *b* = ... 11*a* + 2*b* [stop writing] ...			
134	22:25	Here what I want is that ... *a* + ... *a* + ... *b* = ... no, *a* + ...	J	TP	
135	22:38	Oh, cannot use algebra [cancel her algebra working].	J	AL	Fail to use algebra to justify
136	22:41	So my finding is this ... I … I can ...	X	XH	
137	22:44	[Refer to given checklist] … [Read from checklist] Look back. Devise and test conjectures ... Look back. Check the whole solution. Check answer to see if it is correct by substituting answer back into the task [stop reading] ...	X	XC	
138	23:04	So ... my finding, this is my finding one [cancel her algebra working again] ... I find that if the [start writing] if the original number is even, is an odd number, is an odd number, all the number [she did not write the word ‘numbers’] obtained will be odd number, all the number obtained will be odd [she did not write the word ‘odd’] number. Same as for even numbers, even numbers [stop writing]	X	XW	Write down Conjecture 2 which she verbalised in Line 131
139	23:54	Because the [rewrite 11 in Example 6 properly] ... odd, odd number ... odd number is ... odd number is ... um ...	J	XH	Refer to Example 6
140	24:09	I try 21 [write: 21 → 2 + 1 = 3] is 3, right? Then [start writing] 21 + 3 = 24 [stop writing]; 2 + ... 4 = 6; then 21 ... 24 + 6 = 30.	J	TE8	Example 8 (purposeful choice of odd number): p. 2 bottom part column 2 (above E.g. 6)
141	24:28	So this one is not working [cancel Conjecture 2], cannot ...	J	RC2	Refute Conjecture 2
142	24:34	So if the original number can be divided by 3 or 9, the new numbers obtained also can be divided by 3 or 9 ...	J	XR	Re-read Conjecture 1
143	24:44	This ... this can be proofed [*sic*] [start writing] this can be proofed [*sic*] ... that [stop writing] Once a two-number [*sic:* two-digit number] can be divided by 3, the sum of its digits must be divided by 3. So the sum of its digits can be divided by 3, then I add to them ... uh, odd number, the original number. So it increase, it increase ... um ... it increase ... 3 times a number.	J	RE	Key Moment 2: Discover proof for Conjecture 1
144	25:23	So this number can be divided by 3 ...	J	JC1	Proven First Part of Conjecture 1 correctly
145	25:27	This can be proofed [*sic*] [cancel: that] [continue writing] from ... If a number can be divided by 3, can be, if a number can be divided by 3, then the ... then the ... then [cancel: the] its digits, its [cancel: di] its sum of digits, must be, must be divided by 3, must be divided by 3 ... If add, if	J	XW	Write proof that has already been verbalised above

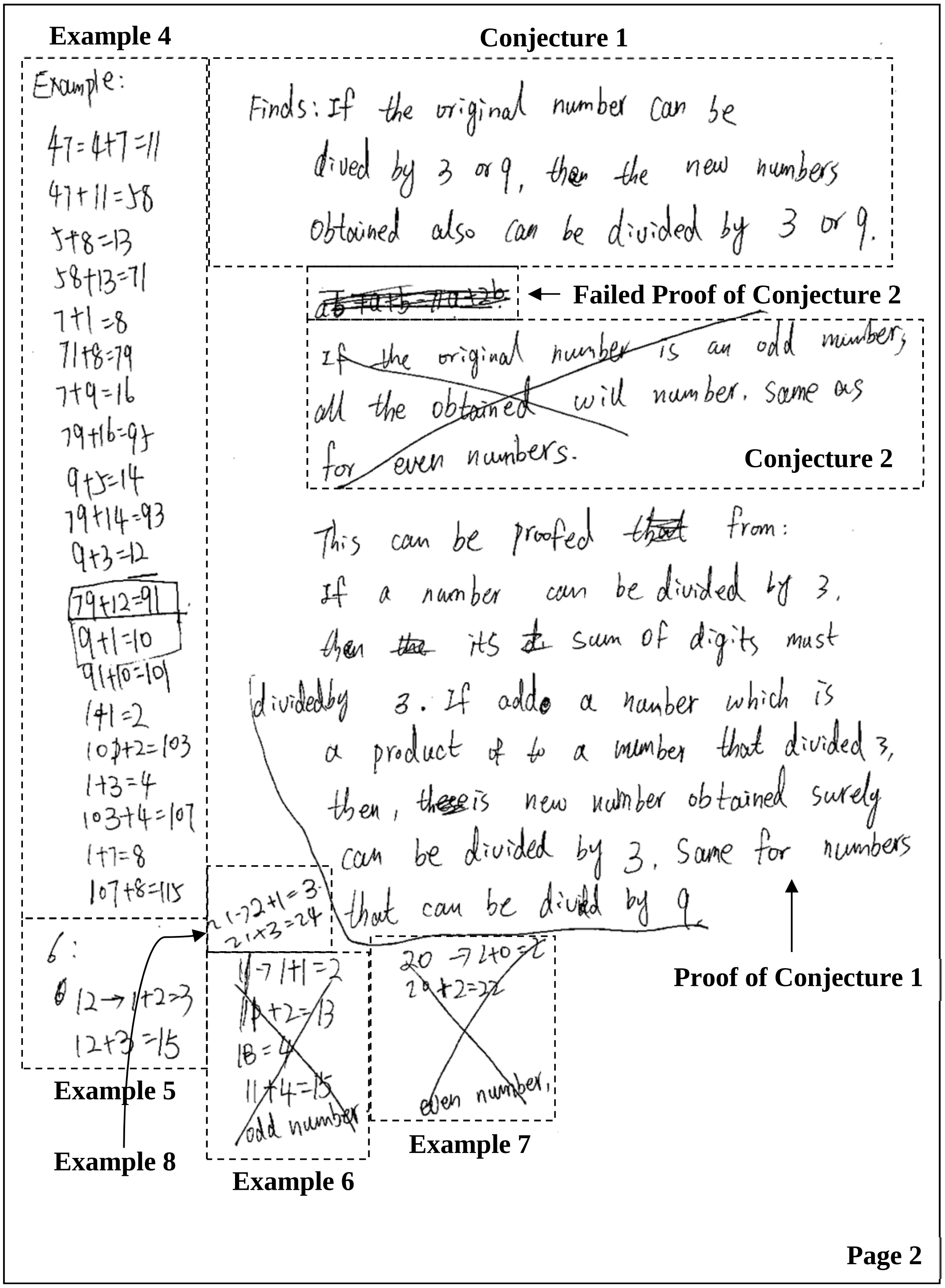
Line	Time	Transcript	Stage Code	Bhvr Code	Remarks
		add a number which is, which is a product of 3, if add a number which is a product of 3 [she did not write the number 3] to a number can be divided by 3, to a number that can be divided by 3, then this new number [cancel: these] then this [write: is] [continue writing] new number obtained, new number obtained, surely can be divided by 3, can be divided by 3.			
146	27:12	Same, same for, for numbers that can be divided by 9, divided by 9 [stop writing].	J	RE	
147	27:34	So this is my [write a line to separate the proof from Examples 6-8 at bottom part of p. 2] this is my finding ...	G	JC1/ SG	Proven Second Part of Conjecture 1, leading to Generalisation
148	27:41	Now, it says [read from given checklist] Look back ... Check the whole solution. Check answer to see if it is correct ... Check the whole solution [stop reading].	R	XC	
149 (p.1)	27:52	I check the number is ... um, can be divided by 3 ... Yes, these, all numbers [mostly likely only Examples 5 and 8] can be divided by 3 ... The rest is [flip to p. 1] 12, 12, the numbers, the new numbers obtained from 12 also can be divided by 3.	R	CW	Refer to Examples 5 and 8 Refer to Example 1
150	28:11	[Pause 5 s]	X	XP	
151 (p.2)	28:16	That means ... [flip back to p. 2] to find the [flip to p. 1] to find the pattern for these numbers ... [flip back to p. 2] so except for 3 and 9, the rest I can't find the pattern ... The ... odd number and even number is not counted [cancel Examples 6 and 7] as they are not working.	R	MR	Review if solution satisfies goal (Counted as 1×MR in Lines 151 and 153 since this is same instance)
152	28:43	[Pause 5 s]	X	XP	
153	28:48	So this is my … whole solution.	R	MR	Solution has satisfied goal
154	28:53	[Read from given checklist] Extend specific problems solved but still within the scope of the original task [stop reading] ...	R	XC	
155	29:03	You mean the ... original task means ... the task [*sic:* should be goal] is to find the pattern. Now I find the pattern for numbers that can be divided by 3 and 9 ...	R	MR	Counted as 1×MR in Lines 155 and 157 since this is same instance
156	29:14	What the pattern for the rest is ... 4 ... 4 + 7 [from Example 4] ... 11 ... As for the numbers that ... 12 [point pen at 12] ... 79 + 12 is [use calculator for the first time in this task] 91 [stop using calculator]. 91 cannot be divided by 9 ...	R	CW	Briefly check solution that the rest has no pattern
157	29:44	So this is the ... answer, this is the investigation for ... the numbers that can be divided by 3 ... or ... 3 or 9. [Time's up at 29:57]	R	MR	Solution has satisfied goal

Total time = **29:57**

APPENDIX J: SAMPLE ANSWER SCRIPT FOR POSTTEST TASK 1 (KAPREKAR)

This appendix shows an answer script from S5 for Posttest Task 1 (Kaprekar Sequences) that is used as a sample in Chapter 4. For ease of reference, the problems, examples, conjectures and pages are numbered as shown.




Example 4
Conjecture 1
Example:
47=4+7=11
47+11=58
5+8=13
58+13=71
7+1=8
71+8=79
7+9=16
79+16=95
9+5=14
79+14=93
9+3=12
79+12=91
9+1=10
91+10=101
1+1=2
101+2=103
1+3=4
103+4=107
1+7=8
107+8=115
Finds: If the original number can be dived by 3 or 9, then the new numbers obtained also can be divided by 3 or 9.
Failed Proof of Conjecture 2
If the original number is an odd number, all the obtained will number. Same as for even numbers.
Conjecture 2
This can be proofed from:
If a number can be divided by 3, then its sum of digits must divided by 3. If add a number which is a product of a number that divided 3, then, the new number obtained surely can be divided by 3. Same for numbers that can be divid by 9.
Proof of Conjecture 1
6:
12→1+2=3
12+3=15
Example 5
Example 8
21→2+1=3
21+3=24
11→1+1=2
11+2=13
1+3=4
11+4=15
odd number
Example 6
20→2+0=2
20+2=22
even number,
Example 7
Page 2

APPENDIX K: INTER-CODER RELIABILITY TEST FOR QUALITY OF PROBLEMS POSED

This appendix shows the inter-coder reliability test for the quality of problems posed for the two pretest tasks and the two posttest tasks. Table K1 shows a sample of a list of problems for Pretest Task 1 given to two coders to classify whether each problem was trivial or non-trivial by ticking the appropriate column. The problems were obtained from the task analysis in Appendix E.

Table K1 Sample List of Problems Given to Coders

Tick whether the following problems are trivial or non-trivial. A problem does not need to be solvable.

No.	Problems Posed	Trivial	Non-trivial
1.	General Problem: Is there any pattern?		
2.	Is there any pattern in consecutive new numbers (i.e. consecutive terms of the sequence)?		
3.	Is there any pattern in the differences between consecutive new numbers?		
4.	Is there any pattern in the last digit of consecutive new numbers?		
5.	Is there a general formula to obtain the next term of the sequence?		
6.	Is there a general formula for a happy number or a sad number[42]?		
7.	Are there more happy numbers than sad numbers?		
8.	Are there infinitely many happy numbers and sad numbers?		
9.	Is the sum of two happy numbers happy or sad?		
10.	Is the product of two happy numbers happy or sad?		

[42] Students were not expected to know the terms 'happy number' and 'sad number', but the terms will be used here for ease of discussion.

Table K2 to Table K5 show the detailed results of the inter-coder reliability test for Pretest Task 1, Pretest Task 2, Posttest Task 1 and Posttest Task 2 respectively, which was discussed in detail in Section 5.4.

Table K2 Inter-Coder Reliability Test for Quality of Problems for Pretest Task 1

No.	Problems Posed	Researcher	Coder 1	Coder 2
1.	General Problem: Is there any pattern?	Trivial	Trivial	Trivial
2.	Is there any pattern in consecutive new numbers (i.e. consecutive terms of the sequence)?	Trivial	Trivial	Trivial
3.	Is there any pattern in the differences between consecutive new numbers?	Trivial	Trivial	Trivial
4.	Is there any pattern in the last digit of consecutive new numbers?	Trivial	Trivial	Trivial
5.	Is there a general formula to obtain the next term of the sequence?	Non-trivial	Non-trivial	Non-trivial
6.	Is there a general formula for a happy number or a sad number?	Non-trivial	Non-trivial	Non-trivial
7.	Are there more happy numbers than sad numbers?	Non-trivial	Non-trivial	Non-trivial
8.	Are there infinitely many happy numbers and sad numbers?	Non-trivial	Non-trivial	Trivial
9.	Is the sum of two happy numbers happy or sad?	Non-trivial	Trivial	Non-trivial
10.	Is the product of two happy numbers happy or sad?	Non-trivial	Trivial	Non-trivial
No. of Agreements between Researcher and Coder			**8**	**9**
Percentage of Agreements between Researcher and Coder			**80%**	**90%**
Average Percentage of Agreements between Researcher and Coders			**85%**	

Table K3 Inter-Coder Reliability Test for Quality of Problems for Pretest Task 2

No.	Problems Posed	Researcher	Coder 1	Coder 2
1.	Find how to toast the three slices of bread.	Trivial	Trivial	Trivial
2.	Find the time taken to toast the three slices of bread.	Trivial	Trivial	Trivial
3.	Find a few methods to toast the three slices of bread.	Non-trivial	Trivial	Non-trivial
4.	Find the shortest time to toast the three slices of bread.	Non-trivial	Trivial	Non-trivial
5.	Find the shortest time needed to toast n slices if the grill can hold exactly two slices.	Non-trivial	Non-trivial	Non-trivial
6.	Find the shortest time needed to toast n slices if the grill can hold exactly m slices.	Non-trivial	Non-trivial	Non-trivial
7.	Find the shortest time needed to toast n slices if the grill can hold exactly two slices, and it takes a seconds to toast one side of a slice of bread, b seconds to put a slice in or to take a slice out, and c seconds to turn a slice over.	Non-trivial	Non-trivial	Non-trivial
8.	Find the shortest time needed to toast n slices if the grill can hold exactly m slices, and it takes a seconds to toast one side of a slice of bread, b seconds to put a slice in or to take a slice out, and c seconds to turn a slice over.	Non-trivial	Non-trivial	Non-trivial
No. of Agreements between Researcher and Coder			**6**	**8**
Percentage of Agreements between Researcher and Coder			**75%**	**100%**
Average Percentage of Agreements between Researcher and Coders			**87.5%**	

Table K4 Inter-Coder Reliability Test for Quality of Problems for Posttest Task 1

No.	Problems Posed	Researcher	Coder 1	Coder 2
1.	General Problem: Is there any pattern?	Trivial	Trivial	Trivial
2.	Is there any pattern in consecutive new numbers (i.e. consecutive terms of the sequence)?	Trivial	Trivial	Trivial
3.	Is there any pattern in the difference between consecutive new numbers (which is the same as the sums of digits of consecutive new numbers)?	Trivial	Trivial	Trivial
4.	Is there any pattern in the last digit of consecutive new numbers?	Trivial	Trivial	Trivial
5.	Is there a general formula to obtain the next term of the sequence?	Non-trivial	Non-trivial	Non-trivial
6.	Are there numbers that will never appear as the second or subsequent terms of any Kaprekar sequence (these are called *self numbers*[43])?	Non-trivial	Non-trivial	Non-trivial
7.	Can a number appear in two Kaprekar sequences that start with a different self number?	Non-trivial	Non-trivial	Non-trivial
8.	Is there any pattern in consecutive self numbers?	Non-trivial	Trivial	Non-trivial
9.	Is there a general formula for self numbers?	Non-trivial	Non-trivial	Non-trivial
10.	Are there infinitely many self numbers?	Non-trivial	Non-trivial	Trivial
No. of Agreements between Researcher and Coder			**9**	**9**
Percentage of Agreements between Researcher and Coder			**90%**	**90%**
Average Percentage of Agreements between Researcher and Coders			**90%**	

Table K5 Inter-Coder Reliability Test for Quality of Problems for Posttest Task 2

No.	Problems Posed	Researcher	Coder 1	Coder 2
1.	Find how to cut the 12 identical sausages to share them equally among the 18 people.	Trivial	Trivial	Trivial
2.	Find the amount of sausages each person will receive.	Trivial	Trivial	Trivial
3.	Find the number of cuts needed to share the 12 identical sausages equally among the 18 people.	Non-trivial	Non-trivial	Trivial
4.	Find a few methods to cut the 12 identical sausages to share them equally among the 18 people.	Non-trivial	Non-trivial	Trivial
5.	Find the least number of cuts needed to share the 12 identical sausages equally among the 18 people.	Non-trivial	Non-trivial	Non-trivial
6.	Find the amount of sausages that each person will receive when n identical sausages are shared equally among m people	Trivial	Trivial	Trivial
7.	Find the least number of cuts needed to share n identical sausages equally among m people.	Non-trivial	Non-trivial	Non-trivial
No. of Agreements between Researcher and Coder			**7**	**5**
Percentage of Agreements between Researcher and Coder			**100%**	**71.4%**
Average Percentage of Agreements between Researcher and Coders			**85.7%**	

[43] Students were not expected to know the term ‘self numbers’, but the term will be used here for ease of discussion.

APPENDIX L: INTER-CODER RELIABILITY TEST FOR QUALITY OF CONJECTURES FORMULATED

This appendix shows the inter-coder reliability test for conjectures formulated for the two pretest tasks and the two posttest tasks. Table L1 shows a sample of a list of conjectures for Pretest Task 1 given to two coders to classify whether each conjecture was trivial or non-trivial by ticking the appropriate column. The conjectures were obtained from the task analysis in Appendix E.

Table L1 Sample List of Conjectures Given to Coders

Tick whether the following conjectures are trivial or non-trivial. A conjecture that is false can be non-trivial as well.

No.	Conjecture Formulated	Trivial	Non-trivial
1.	If a number in a zapping sequence is happy or sad[44], then all the numbers in the same sequence are also happy or sad respectively, e.g. 28 → 68 → 100 → 1, so all the numbers, 28, 68, 100 and 1, are happy numbers.		
2.	The rearrangement of the digits of a number does not matter in determining whether the number is happy or sad, e.g. 28 and 82 are both happy.		
3.	The insertion or removal of any number of zeros anywhere in a number does not affect whether the number is happy or sad, e.g. 47, 407 and 7040 are all sad.		
4.	When two zapping sequences first merge at the same number, the preceding terms before this number in the two sequences will be different numbers but with the *same unique combination*, i.e. the difference in the preceding terms will just be rearrangements of the digits and/or insertion of zeros anywhere in the terms. For example, if two zapping sequences merge at the same number 100, then the preceding terms will be one of these numbers: 68, 86, 608, 680, 806, 860, 6008, etc. **[False]**		
5.	A positive integer is either a happy or a sad number.		
6.	There are more sad numbers than happy numbers.		
7.	There are infinitely many happy numbers. Similarly, there are infinitely many sad numbers.		
8.	The sum of two happy numbers is always happy. Similarly, the sum of two sad numbers is always sad. **[False]**		
9.	The product of two happy numbers is always happy. Similarly, the product of two sad numbers is always sad. **[False]**		

[44] Students were not expected to know the terms ‘zapping sequence’, ‘happy number’ and ‘sad number’, but the terms will be used in this thesis for ease of discussion.

Table L2 to Table L5 show the detailed results of the inter-coder reliability tests for Pretest Task 1, Pretest Task 2, Posttest Task 1 and Posttest Task 2 respectively, which was discussed in detail in Section 5.4.

Table L2 Inter-Coder Reliability Test for Quality of Conjectures for Pretest Task 1

No.	Conjecture Formulated	Researcher	Coder 1	Coder 2
1.	If a number in a zapping sequence is happy or sad, then all the numbers in the same sequence are also happy or sad respectively, e.g. 28 → 68 → 100 → 1, so all the numbers, 28, 68, 100 and 1, are happy numbers.	Trivial	Trivial	Trivial
2.	The rearrangement of the digits of a number does not matter in determining whether the number is happy or sad, e.g. 28 and 82 are both happy.	Trivial	Trivial	Trivial
3.	The insertion or removal of any number of zeros anywhere in a number does not affect whether the number is happy or sad, e.g. 47, 407 and 7040 are all sad.	Trivial	Trivial	Trivial
4.	When two zapping sequences first merge at the same number, the preceding terms before this number in the two sequences will be different numbers but with the *same unique combination*, i.e. the difference in the preceding terms will just be rearrangements of the digits and/or insertion of zeros anywhere in the terms. For example, if two zapping sequences merge at the same number 100, then the preceding terms will be one of these numbers: 68, 86, 608, 680, 806, 860, 6008, etc. **[False]**	Non-trivial	Non-trivial	Non-trivial
5.	A positive integer is either a happy or a sad number.	Non-trivial	Non-trivial	Non-trivial
6.	There are more sad numbers than happy numbers.	Non-trivial	Non-trivial	Non-trivial
7.	There are infinitely many happy numbers. Similarly, there are infinitely many sad numbers.	Non-trivial	Non-trivial	Trivial
8.	The sum of two happy numbers is always happy. Similarly, the sum of two sad numbers is always sad. **[False]**	Non-trivial	Trivial	Non-trivial
9.	The product of two happy numbers is always happy. Similarly, the product of two sad numbers is always sad. **[False]**	Non-trivial	Trivial	Non-trivial
No. of Agreements between Researcher and Coder			**7**	**8**
Percentage of Agreements between Researcher and Coder			**77.8%**	**88.9%**
Average Percentage of Agreements between Researcher and Coders			**83.3%**	

Table L3 Inter-Coder Reliability Test for Quality of Conjectures for Pretest Task 2

No.	Conjecture Formulated	Researcher	Coder 1	Coder 2
1.	Shortest time to toast the three slices of bread (by using Toasting Method B) = 113 s	Non-trivial	Trivial	Non-trivial
*2.	Shortest time to toast the three slices of bread (by using Toasting Method A) = 146 s **[false]**	Trivial	Trivial	Trivial
3.	Shortest time needed to toast *n* slices if the grill can hold exactly two slices $= \begin{cases} \frac{73}{2}n \text{ seconds if } n \text{ is even} \\ \frac{73}{2}n + \frac{7}{2} \text{ seconds if } n \text{ is odd and } n \neq 1 \\ 73 \text{ seconds if } n = 1 \end{cases}$	Non-trivial	Non-trivial	Non-trivial
No. of Agreements between Researcher and Coder			**2**	**3**
Percentage of Agreements between Researcher and Coder			**66.7%**	**100%**
Average Percentage of Agreements between Researcher and Coders			**83.3%**	

* This conjecture was not in the task analysis in Appendix E, but was formulated by the students in this study.

Table L4 Inter-Coder Reliability Test for Quality of Conjectures for Posttest Task 1

No.	Conjecture Formulated	Researcher	Coder 1	Coder 2
1.	Every Kaprekar[45] sequence is an increasing sequence, so the terms in each sequence will not repeat themselves, e.g. 28, 38, 49, 62, …	Trivial	Trivial	Trivial
2.	Some numbers will never appear as the second or subsequent terms in any Kaprekar sequence, e.g. 1, 3, 5, 7, 9, 20, 31, 42, 53, 64, 75, 86, 97, 108, 110, … (these are called *self numbers* and they will only appear as the first term in one unique Kaprekar sequence).	Non-trivial	Trivial	Non-trivial
3.	All one-digit odd numbers are self numbers.	Trivial	Trivial	Trivial
4.	Difference between consecutive one-digit self numbers is always 2.	Trivial	Trivial	Trivial
5.	Difference between the last one-digit self number 9 and first two-digit self number 20; and difference between consecutive two-digit self numbers (i.e. 20, 31, 42, 53, 64, 75, 86, 97) is always 11.	Trivial	Trivial	Trivial
6.	A 1-digit or a 2-digit no. in a Kaprekar sequence will never appear in another Kaprekar sequence with a different self number, but a 3-digit number can appear in two Kaprekar sequences with different self numbers, e.g. 107.	Non-trivial	Non-trivial	Non-trivial
7.	If a self number is a multiple of 3 or 9, then all the terms in the Kaprekar sequence are also multiples of 3 or 9 respectively.	Non-trivial	Non-trivial	Non-trivial
8.	If a self number is not a multiple of 3 or 9, then the sums of the digits of the *differences* between consecutive terms of the Kaprekar sequence repeat themselves (i.e. 1, 2, 4, 8, 7, 5) with a period of 6.	Non-trivial	Non-trivial	Non-trivial
9.	There are infinitely many self numbers.	Non-trivial	Trivial	Non-trivial
No. of Agreements between Researcher and Coder			**7**	**9**
Percentage of Agreements between Researcher and Coder			**77.8%**	**100%**
Average Percentage of Agreements between Researcher and Coders			**88.9%**	

[45] Students were not expected to know the term 'Kaprekar', but it will be used in this thesis for ease of discussion. Similarly for other terms such as 'self numbers'.

Table L5 Inter-Coder Reliability Test for Quality of Conjectures for Posttest Task 2

No.	Conjecture Formulated	Researcher	Coder 1	Coder 2
1.	Amount of sausage each person will get = $\dfrac{\text{No. of Sausages}}{\text{No. of People}}$	Trivial	Trivial	Trivial
2.	Least number of cuts to share 12 identical sausages equally among 18 people (by using Cutting Method B) = 12	Non-trivial	Non-trivial	Non-trivial
*3.	Least number of cuts to share 12 identical sausages equally among 18 people (by using Cutting Method A) = 24 **[False]**	Trivial	Trivial	Trivial
*4.	Least number of cuts to share 12*n* identical sausages equally among 18 people = 12*n* **[False]**	Trivial	Trivial	Trivial
5.	Least number of cuts to share *n* identical sausages equally among *m* people = *m* – HCF(*m*,*n*).	Non-trivial	Non-trivial	Non-trivial
*6.	It is always possible to share *n* identical sausages equally among *m* people with no remainder.	Trivial	Trivial	Trivial
*7.	It is always possible to share *n* identical sausages equally among *m* people by dividing each sausage into $\dfrac{\text{LCM}(n,m)}{n}$ equal parts.	Non-trivial	Non-trivial	Trivial
*8.	It is always possible to share 6*n* identical sausages equally among 6*n* + 6 people by cutting each sausage at the $\dfrac{1}{n+1}$ -mark.	Non-trivial	Trivial	Non-trivial
No. of Agreements between Researcher and Coder			**7**	**7**
Percentage of Agreements between Researcher and Coder			**87.5%**	**87.5%**
Average Percentage of Agreements between Researcher and Coders			**87.5%**	

* These conjectures were not in the task analysis in Appendix E, but they were formulated by the students in the present study.

APPENDIX M: SUMMARY TABLES OF PROCESSES AND OUTCOMES FOR PRETEST

This appendix shows the Summary Table of Processes and Outcomes (TPO) for Pretest Task 1 (Happy) and Pretest Task 2 (Toast) obtained by examining the thinking-aloud protocols and answer scripts of the 10 students in the present study. The explanations of the codes were given in the Final Coding Scheme in Section 4.6.2, and the findings based on these data were discussed in Section 8.3.

N1. Pretest Task 1 (Happy)

(a) Understanding the Task (Stage 1)

Table M1.1 shows a summary of the understanding processes engaged by the 10 students and the types of outcomes for Pretest Task 1. The 'Total' column shows the total frequency for RR, RT, HI and MU (excluding TE) for each student.

Table M1.1 Understanding Processes and Outcomes for Happy Task

	Processes						Outcomes	
	TE	**RR**	**RT**	**HI**	**MU**	**Total**	**Understood**	**Misinterpreted**
S1	1	5*				**5**		Did not recover
S2	1					**0**		Recovered after 2 min
S3	2	5+8=13*	0+1=1	0+1=1		**15**		Did not recover
S4	1	1	3			**4**	✓	
S5	1	3	1		1	**5**	✓	
S6	1	5				**5**		Did not recover
S7	5	7	2			**9**		Did not recover
S8	1	1				**1**		Did not recover
S9	1					**0**		Did not recover
S10	2	1				**1**		Recovered after 9 min
Total	**16**	**36**	**7**	**1**	**1**	**45**	**2**	**8 misinterpreted; 2 recovered**

* S3 engaged in RR for 5 + 8 = 13 times means that RR happened 5 times in the first episode and 8 times in the second episode; $5 \times$ RR for S1 means that this process happened 5 times in the first episode.

(b) Problem Posing and Extension (Stages 2 and 8)

Table M1.2 shows a summary of the problem-posing processes engaged by the 10 students and the types of outcomes for Pretest Task 1. Since 7 students (S2-S7,S10) did not pose any specific problem, they were excluded from the table. All the students did not pose the general problem of searching for any pattern, but they just went ahead to search for patterns after understanding the task, except for S1 who started with a non-trivial problem (i.e. S1 did not search for any pattern). All the students also did not extend the task by changing the given. No student analysed the feasibility of their goal (MG).

Table M1.2 Problem-Posing Processes and Outcomes for Happy Task

	Processes					Outcomes	
	Refer to following to think of problem to pose						
	Task Statement	**Current Working**	**Previous Result**	**Given Checklist**	**MG**	**General Problem**	**Specific Problem**
S1	1						1 non-trivial
S8		1					2 trivial
S9		1					1 non-trivial
Total	**1**	**2**	**0**	**0**	**0**	**0**	**2 trivial; 2 non-trivial**

(c) Specialising (Stage 3)

Table M1.3 shows a summary of the specialising processes engaged by the 10 students and the types of outcomes for Pretest Task 1. Since all the students used the random example(s) generated for understanding the task to search for patterns as well, these examples were also included in the table for specialising. But examples used to test conjectures (E.g. 6,8,9 for S1; E.g.13-14 for S5; E.g. 5 for S8) in the justifying stage were excluded. The examples generated were considered representative if the students were able to generate at least one sequence that belongs to happy numbers and at least one sequence that belongs to sad numbers. None of the students used other heuristics for this task. No student analysed the feasibility of their plan (MF) in this stage, so this metacognitive process was omitted from the table.

Table M1.3 Specialising Processes and Outcomes for Happy Task

	Processes				Outcomes		
	Specialising			MA	Total No. of E.g.	Rep. E.g.	Not Rep. E.g.
	Random	Purposeful	Systematic				
S1	6 (E.g. 1-5,7)				6	NA*	
S2	6 (E.g. 1-6)			3	6		✓ (sad nos.)
S3	2 (E.g. 1-2)	4 (E.g. 3-6)			6	NA*	
S4	6 (E.g. 1,3,4,7-9)	3 (E.g. 2,5,6)			9	✓	
S5	10 (E.g. 1-4,7-12)	4 (E.g. 5,6,15,16)		2	14	✓	
S6	3 (E.g. 1,3,5)	3 (E.g. 2,4,6)			6	NA*	
S7	7 (E.g. 1-7)				7	NA*	
S8	7 (E.g. 1-4,6-8)				7	NA*	
S9	1 (E.g. 1)				1	NA*	
S10	18 (E.g. 1-13,15-19)	1 (E.g. 14)		2	19	✓	
Total	**66 (81%)**	**15 (19%)**	**0**	**7**	**81**	**3**	**1**

* 6 students misinterpreted the task so badly (e.g. took square root instead of squaring, or did not repeat the process) that the patterns, if any, were no longer the same as the original task. As a result, the examples could not be classified as representative because either there was no known pattern or only one type of examples.

(d) Conjecturing (Stage 4)

Table M1.4a shows a summary of the types of patterns and conjectures produced by the 10 students for Pretest Task 1. Other types of patterns refer to patterns not related to the original task which students discovered because they had misinterpreted the task. Each number in brackets indicates the number of correct patterns or conjectures, e.g. 4(3) trivial patterns for S3 indicates that S3 had observed 4 trivial patterns, out of which 3 of them were correct.

Table M1.4a Patterns and Conjectures for Happy Task

	Related to Original Task				Other Types of Patterns				Total No. of Patt.	Total No. of Conj.
	Patterns		Conjectures		Patterns		Conjectures			
	non-trivial	trivial	non-trivial	trivial	non-trivial	trivial	non-trivial	trivial		
S1						1		3	**1**	**3**
S2	1(1)		1						**1(1)**	**1**
S3						4(3)			**4(3)**	**0**
S4	2(2)	2(2)							**4(4)**	**0**
S5	2(2)	2(2)	1(1)						**4(4)**	**1(1)**
S6						1			**1**	**0**
S7						1			**1**	**0**
S8						2(2)		2(2)	**2(2)**	**2(2)**
S9						1(1)		2(1)	**1(1)**	**2(1)**
S10	3(2)	1(1)							**4(3)**	**0**
Total	**8(7)**	**5(5)**	**2(1)**	**0**	**0**	**10(6)**	**0**	**7(3)**	**23(18)**	**9(4)**

Table M1.4b shows a summary of the conjecturing processes engaged by the 10 students and the types of outcomes for Pretest Task 1. Each number in brackets indicates the number of correct patterns or conjectures. No student analysed the feasibility of their plan (MF) in this stage, so this metacognitive process was omitted from the table. Although 2 students (S1,S9) exhibited metacognitive awareness (MA) in this stage, it was not included in this table because it would be analysed in the checking stage (see Table M1.6) since it involved the students sensing something amiss and checking their working.

Table M1.4b Conjecturing Processes and Outcomes for Happy Task

	Specialising Outcomes		Conjecturing Processes			Conjecturing Outcomes related to Original Task	
	No. of E.g.	Rep. E.g.	Searched for patterns in: Terms	Diff. b/w Terms	Others	Observed Patterns	Formulated Conjectures
S1	6	NA*			✓	NA*	
S2	6	Sad nos. only	✓	✓		1(1)	1
S3	6	NA*		✓		NA*	
S4	9	✓	✓			4(4)	
S5	14	✓	✓			4(4)	1(1)
S6	6	NA*		✓		NA*	
S7	7	NA*	✓			NA*	
S8	7	NA*			✓	NA*	
S9	1	NA*		✓		NA*	
S10	19	✓	✓	✓	✓	4(3)	
Total	**81**	**3 rep.**	**5**	**5**	**3**	**13(12)**	**2(1)**

* 6 students misinterpreted the task so badly (e.g. took square root instead of squaring, or did not repeat the process) that the patterns, if any, were no longer the same as the original task. As a result, the examples could not be classified as representative because either there was no known pattern or only one type of examples.

(e) Justifying and Generalising (Stages 5 and 6)

Table M1.5 shows a summary of the justifying processes engaged by the 10 students and the types of outcomes for Pretest Task 1. No student exhibited any metacognitive behaviour (MF or MA) in these stages, so these processes were omitted from the table. Since 5 students did not formulate any conjecture to justify, they were excluded from the table.

Table M1.5 Justifying / Generalising Processes and Outcomes for Happy Task

	Conjecture	Justifying Processes: Naïve Testing	Non-proof Argument	Formal Proof	Justifying and Generalising Outcomes
S1	Wrong Trivial Conjecture 1			✓	Tried but failed to justify
	Wrong Trivial Conjecture 2	✓			Wrongly accepted conjecture as true based on naïve testing
	Wrong Trivial Conjecture 3	✓			Test ended during naïve testing
S2	Wrong Non-Trivial Conjecture 1		✓		Tried but failed to justify
S5	Correct Non-trivial Conjecture 1	✓			Wrongly accepted conjecture as true based on naïve testing
S8	Correct Trivial Conjecture 1	✓			Wrongly accepted conjecture as true based on naïve testing
	Correct Trivial Conjecture 2				Wrongly accepted conjecture as true without testing
S9	Correct Trivial Conjecture 1	✓			Wrongly accepted conjecture as true based on naïve testing
	Wrong Trivial Conjecture 2	✓			Refuted conjecture based on naïve testing
Total	**9**	**6**	**1**	**1**	**0 proven; 0 generalisation***

* There was no generalisation for all the conjectures either because the conjecture was wrong or it was not proven.

(f) Checking (Stage 7)

Table M1.6 shows a summary of the monitoring and checking processes engaged by the 10 students and the types of outcomes for Pretest Task 1. The time indicated in brackets for 'Errors Discovered' refers to the time interval between making the major mistake and discovering the mistake. No time is indicated for the discovery of a minor error because the discovery time is not an important factor when the error is minor.

Table M1.6 Monitoring / Checking Processes and Outcomes for Happy Task

	Processes							Outcomes	
	Check Working				MP	MR	MA	Errors Made	Errors Discovered
	Most Parts	Some Parts	Glance Briefly	Total					
S1	1			**1**	2	1	2	2 minor	2 minor
S2				**0**	0	0	0	1 major + 4 minor = 5	1 major (6 min later) + 1 minor = 2
S3		1		**1**	1	0	0	2 major + 6 minor = 8	6 minor
S4	2	2		**4**	4	1	0	1 major + 1 minor = 2	1 minor
S5				**0**	0	2	0	1 major + 4 minor = 5	1 minor
S6				**0**	2	0	0	1 major + 3 minor = 4	2 minor
S7				**0**	11	0	0	1 minor	1 minor
S8				**0**	4	0	0	0	0
S9		4	1	**5**	6	1	2	4 major + 6 minor = 10	2 major (6 min later) + 6 minor = 8
S10		1		**1**	1	0	0	1 minor	0
Total	**3**	**8**	**1**	**12**	**31**	**5**	**4**	**10 major + 28 minor = 38**	**3 major + 20 minor = 23**

N2. Pretest Task 2 (Toast)

(a) Understanding the Task (Stage 1)

Table M2.1 shows a summary of the understanding processes engaged by the 10 students and the types of outcomes for Pretest Task 2. The 'Total' column shows the total frequency for RR, RT, HI and MU for each student.

Table M2.1 Understanding Processes and Outcomes for Toast Task

	Processes						Outcomes	
	RR	RT	HI	VI	MU	Total	Understood	Misinterpreted
S1	1*	1	3			**5**		Did not recover
S2	0+1=1			0+1=1	0+1=1	**3**	✓	
S3	0+3=3*	0+2=2				**5**		Did not recover
S4						**0**		Did not recover
S5	4			1	2	**7**		Recovered after 4 min
S6	2	1		2	2	**7**	✓	
S7	1			1		**2**		Did not recover
S8					1	**1**	✓	
S9		2		1	2	**5**		Recovered after 3 min
S10	2+2=4		0+1=1		1+2=3	**8**		Recovered after 7 min
Total	**16**	**6**	**4**	**6**	**11**	**43**	**3**	**7 misinterpreted; 3 recovered**

* S3 engaged in RR for 0 + 3 = 3 times means that RR happened 0 times in the first episode and 3 times in the second episode; 1 × RR for S1 means that this process happened one time in the first episode.

(b) Problem Posing and Extension (Stages 2 and 8)

During the task analysis of Pretest Task 2 in Appendix E, some problems that students could pose were identified and classified as trivial or non-trivial. These were reproduced below. Problem 4 was the intended problem for this task. The solutions of these problems were not general results.

- *Problem 1 (P1):* Find how to toast the three slices of bread. [Trivial]
- *Problem 2 (P2):* Find the time taken to toast the three slices of bread. [Trivial]
- *Problem 3 (P3):* Find a few methods to toast the three slices of bread. [Non-Trivial]
- *Problem 4 (P4):* Find the shortest time to toast the three slices of bread. [Non-Trivial; Intended Problem]

Table M2.2a shows a summary of the types of problems posed by the 10 students.

Table M2.2a Problems Posed for Toast Task

	P1 (trivial)	P2 (trivial)	P3 (non-trivial)	P4 (non-trivial)	Other Trivial Prob.	Other Non-Trivial Prob.	Total No. of Trivial Prob.	Total No. of Non-Trivial Prob.	Total No. of Prob.
S1		✓		✓	1		2	1	**3**
S2				✓	1		1	1	**2**
S3		*		✓	1		2	1	**3**
S4				✓			0	1	**1**
S5				✓			0	1	**1**
S6		*		✓			1	1	**2**
S7		✓					1	0	**1**
S8		✓	✓				1	1	**2**
S9				✓			0	1	**1**
S10				✓			0	1	**1**
Total	**0**	**5**	**1**	**8**	**3**	**0**	**8**	**9**	**17**

* Did not pose problem explicitly

Table M2.2b shows a summary of the problem-posing processes engaged by the 10 students and the types of outcomes for Pretest Task 2. If the students struggled to pose the problem eventually, it was denoted by 'E'; if they posed the problem naturally without struggling, it was denoted by 'N'.

Table M2.2b Problem-Posing Processes and Outcomes for Toast Task

	Processes				Outcomes					
	Refer to following to think of problem			MG	P1	P2	P3	P4	Other Trivial Prob.	Other Non-Trivial Prob.
	Task	Current Working	Previous Result							
S1	1	1	1			N		N	1N	
S2	2							E	1N	
S3	2		1			N*		E	1N	
S4	1							N		
S5	1							N		
S6	2					E*		E		
S7	1					N				
S8	1		1			N	N			
S9	1							N		
S10	1							N		
Total	**13**	**1**	**3**	**0**	**0**	**5**	**1**	**8**	**3**	**0**

* Did not pose problem explicitly

During the task analysis of Pretest Task 2 in Appendix E, some extensions that students could pose were identified and classified as trivial or non-trivial. These were reproduced below. Both extensions were the intended extensions and for the purpose of generalising.

- *Extension 1 (E1):* Find the shortest time needed to toast *n* slices if the grill can hold exactly two slices. [Non-trivial; Intended Extension]

- *Extension 2 (E2):* Find the shortest time needed to toast *n* slices if the grill can hold exactly *m* slices. [Non-Trivial; Intended Extension]

Table M2.2c shows a summary of the types of extensions posed by the 10 students. Since 6 students did not extend the task, they were excluded from the table.

Table M2.2c Extensions for Toast Task

	E1 (non-trivial)	E2 (non-trivial)	Other Extensions to generalise		Other Extensions but not to generalise		Total No. of Extensions		
			Trivial	Non-Trivial	Trivial	Non-Trivial	Trivial	Non-Trivial	All
S3					2		2	0	**2**
S4	✓						0	1	**1**
S7				1	1		1	1	**2**
S9	✓				2		2	1	**3**
Total	**2**	**0**	**0**	**1**	**5**	**0**	**5**	**3**	**8**

Table M2.2d shows a summary of the extension processes engaged by the 10 students and the types of outcomes for Pretest Task 2. Since 6 students did not extend the task, they were excluded from the table. If the students struggled to pose the problem eventually, it was denoted by 'E'; if they posed the problem naturally without struggling, it was denoted by 'N'.

Table M2.2d Extension Processes and Outcomes for Toast Task

	Processes				Outcomes			
	Refer to following to think of extension			MG	E1	E2	Other Extensions to generalise	Other Extensions but not to generalise
	Task	Current Working	Previous Result					
S3			2					2N
S4			1		N			
S7	1		1				1E	1E
S9	1		2		E			2E
Total	**2**	**0**	**6**	**0**	**2**	**0**	**1**	**5**

(c) Specialising and Using Other Heuristics (Stage 3)

Table M2.3a shows a summary of the processes for using other heuristics engaged by the 10 students and the types of outcomes for the original task for Pretest Task 2. No student used algebra to solve the problems posed for this task, so this process was omitted from the table. The explanations of what constituted 'effective reasoning', 'quite effective reasoning' and 'ineffective reasoning' were given in Section 8.2.1. Although 3 other students (S5,S7,S10) exhibited metacognitive awareness (MA) in this stage, it was not included in this table because it would be analysed in the checking stage (see Table M2.6) since it involved the students sensing something amiss and checking their working.

Table M2.3a Using Other Heuristics for Toast Task

	Processes				Outcomes				
	Effective Reasoning	VI	MF	MA	Toasting Methods				
					A	B	C	Others	Total
S1	No				✓		✓	1	**3**
S2	No				✓			1	**2**
S3	No							1	**1**
S4	Yes	1	1	1	✓	✓			**2**
S5	Quite	5	1		✓	✓			**2**
S6	No				✓		✓		**2**
S7	No	6			✓				**1**
S8	Quite	1			✓	✓		2	**4**
S9	No		2		✓		✓	2	**4**
S10	Quite	4			✓	✓		1	**3**
Total	**1 Yes, 3 Quite**	**17**	**4**	**1**	**9**	**4**	**3**	**8**	**24**

Table M2.3b shows a summary of the processes for using other heuristics engaged by the 10 students and the types of outcomes for the extension of Pretest Task 2. Since 6 students did not extend the task, they were omitted from the table. The omitted students did not exhibit any metacognitive behaviour (MF or MA) in this stage.

Table M2.3b Using Other Heuristics for Extension of Toast Task

	Processes				Outcomes				
	Effective Reasoning	VI	MF	MA	Toasting Methods				
					A	B	C	Others	Total
S3	No				✓				1
S4	Yes	2				✓			1
S7	No	10			✓				1
S9	No	3		1	✓				1
Total	**1 Yes, 3 No**	**15**	**0**	**1**	**3**	**1**	**0**	**0**	**4**

(d) Conjecturing (Stage 4)

Table M2.4a shows a summary of the conjecturing processes engaged by the 10 students and the types of outcomes for the original task for Pretest Task 2. The conjectures were classified as trivial or non-trivial based on the task analysis in Appendix E. No student exhibited the metacognitive behaviours, MF and MA, in this stage. Although 8 students (S1-S6, S9,S10) had posed the problem of finding the least number of cuts and so their cutting method was only a conjecture to be proven or refuted, one of them (S6) did not pursue the problem, and 3 of them (S2,S5,S10) did not finish solving the problem. Thus only 4 students were included in the table because they had solved this problem that led to a conjecture. The remaining 2 students (S7,S8) did not pose this problem and so there was no conjecture.

Table M2.4a Conjecturing Processes and Outcomes for Toast Task

	Processes	Outcomes		
	Reasoning	Conjecture	Trivial or non-trivial?	Correct or wrong?
S1	Ineffective	Conjecture 1	Non-trivial	Wrong (use Toasting Method A)
S3	Ineffective	Conjecture 1	Non-trivial	Wrong (use Toasting Method A)
S4	Effective	Conjecture 1	Non-trivial	Correct (use Toasting Method B)
S9	Ineffective	Conjecture 1	Non-trivial	Wrong (use Toasting Method A)
Total	**1 effective**	**4 conjectures**	**4 non-trivial**	**1 correct; 3 wrong**

Table M2.4b shows a summary of the conjecturing processes engaged by the 10 students and the types of outcomes for the extension of Pretest Task 2. The conjectures were classified as trivial or non-trivial based on the task analysis in Appendix E. Since 6 students (S1,S2,S5,S6, S8,S10) did not extend the task, they were omitted from the table. One more student (S7) was excluded because she did not finish solving the extension and so there was no conjecture.

Table M2.4b Conjecturing Processes and Outcomes for Extension of Toast Task

	Processes		**Outcomes**		
	MF	**MA**	**Conjecture**	**Trivial or non-trivial?**	**Correct or wrong?**
S3			Conjecture 2	Non-trivial	Both wrong because his toasting method will not give least toasting time
			Conjecture 3	Non-trivial	
S4			Conjecture 2	Trivial	Correct: last digit of time taken, in seconds, to toast *n* slices repeats
			Conjecture 3	Non-trivial	Correct because Toasting Method B will give least toasting time
S9			Conjecture 2	Non-trivial	Both wrong because Toasting Method A will not give least toasting time
			Conjecture 3	Non-trivial	
			Conjecture 4	Trivial	Correct: repeating pattern in toasting time from odd no. of slices to even no.
Total	**0**	**0**	**7 conjectures**	**2 trivial; 5 non-trivial**	**3 correct; 4 wrong**

(e) Justifying and Generalising (Stages 5 and 6)

Table M2.5a shows a summary of the justifying processes engaged by the 10 students and the types of justifying and generalising outcomes for the original task for Pretest Task 2. None of the students exhibited any metacognitive behaviour (MF or MA) in these stages, so these processes were omitted from the table. Only 4 students were shown in the table because they were the only ones who had formulated conjectures for the original task to justify.

Table M2.5a Justifying / Generalising Processes and Outcomes for Toast Task

	Conjecture	**Justifying Processes**			**Justifying and Generalising Outcomes**
		Naïve Testing	**Non-proof Argument**	**Formal Proof**	
S1	Wrong Non-trivial Conjecture 1				Wrongly accepted conjecture as true without testing
S3	Wrong Non-trivial Conjecture 1				Wrongly accepted conjecture as true without testing
S4	Correct Non-trivial Conjecture 1				Wrongly accepted conjecture as true without testing
S9	Wrong Non-trivial Conjecture 1				Wrongly accepted conjecture as true without testing
Total	**1 correct; 3 wrong**	**0**	**0**	**0**	**0 proven; 0 generalisation***

* There was no generalisation for all the conjectures because none of the conjectures were proven.

Table M2.5b shows a summary of the justifying processes engaged by the 10 students and the types of justifying and generalising outcomes for the extension of Pretest Task 2. None of the students exhibited any metacognitive behaviour (MF or MA) in these stages, so these processes were omitted from the table. Only 3 students were shown in the table because they were the only ones who had formulated conjectures for the extended task to justify.

Table M2.5b Justifying / Generalising Processes and Outcomes for Extension of Toast Task

	Conjecture	Justifying Processes			Justifying and Generalising Outcomes
		Naïve Testing	Non-proof Argument	Formal Proof	
S3	Wrong Non-trivial Conjecture 2				Wrongly accepted conjecture as true without testing
	Wrong Non-trivial Conjecture 3				Wrongly accepted conjecture as true without testing
S4	Correct Trivial Conjecture 2		✓		**Proven conjecture (generalisation)**
	Correct Non-trivial Conjecture 3				Wrongly accepted conjecture as true without testing
S9	Wrong Non-trivial Conjecture 2				Wrongly accepted conjecture as true without testing
	Wrong Non-trivial Conjecture 3				Wrongly accepted conjecture as true without testing
	Correct Trivial Conjecture 4				Wrongly accepted conjecture as true without testing
Total	**3 correct; 4 wrong**	**0**	**1**	**0**	**1 proven; 1 generalisation***

* There was no generalisation for all the conjectures except for the proven conjecture.

(f) Checking (Stage 7)

Table M2.6 shows a summary of the monitoring and checking processes engaged by the 10 students and the types of outcomes for Pretest Task 2. The time indicated in brackets for 'Errors Discovered' refers to the time interval between making the major mistake and discovering the mistake. No time is indicated for the discovery of a minor error because the discovery time is not an important factor when the error is minor.

Table M2.6 Monitoring / Checking Processes and Outcomes for Toast Task

	Processes								Outcomes	
	Check Working					MP	MR	MA	Errors Made	Errors Discovered
	Most Parts	Some Parts	Glance Briefly	Others	Total					
S1		1			**1**	0	1	0	1 minor	1 minor
S2	2				**2**	0	0	0	1 minor	0
S3	2	1			**3**	0	0	0	2 minor	2 minor
S4		2		1	**3**	1	3	0	2 minor	1 minor
S5		1			**1**	0	0	1	2 minor	2 minor
S6					**0**	1	3	0	2 minor	2 minor
S7		1			**1**	0	0	1	1 minor	1 minor
S8					**0**	0	3	0	5 major + 1 minor = 6	0
S9		2	1		**3**	2	5	0	2 minor	1 minor
S10		1			**1**	0	0	1	0	0
Total	**4**	**9**	**1**	**1**	**15**	**4**	**15**	**3**	**5 major + 14 minor = 19**	**10 minor**