Using Coding to Promote Mathematical Thinking with Year 2 Students: Alignment with the Australian Curriculum

Jodie Miller
Australian Catholic University
<Jodie.Miller@acu.edu.au>

Kevin Larkin
Griffith University
<k.larkin@griffith.com.au>

In this paper, we present data from a study exploring the use of coding to promote mathematical thinking. A teaching experiment was undertaken with 40 Year 2 students participating in six 45-minute lessons of coding (one lesson per week for six weeks). All lessons were video-recorded and analysed to determine students' mathematical thinking. Insights from the study reveal that coding contexts promoted higher levels of mathematical thinking for Year 2 students including measuring angles, orientation and perspective-taking, and deducing repeating patterns.

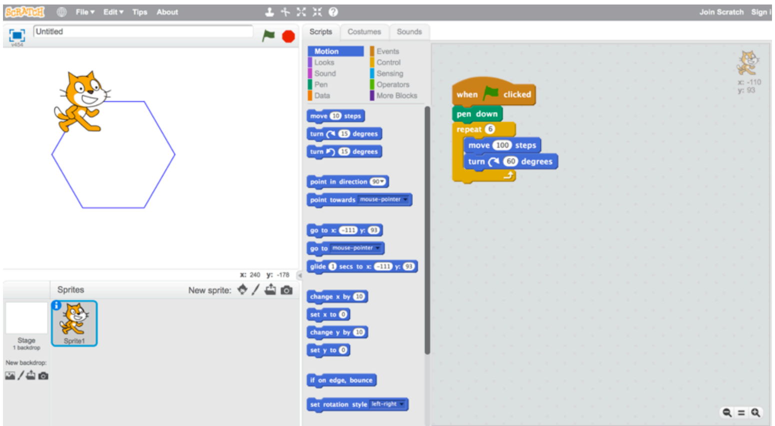
There is an international urgency to improve Science, Technology, Engineering and Mathematics (STEM) education in preparation for a scientifically and technologically advanced society (Office of Chief Scientist, 2014). This push is also in response to the rapid decline in secondary school students' engagement in STEM disciplines (e.g., advanced mathematics, chemistry) (Australian Academy of Science [AAS], 2016). Disengagement in STEM begins at an early age (Larkin & Jorgensen, 2016), with many students from the upper primary years onwards failing the most important STEM subject – mathematics (AAS, 2016). To address this issue, coding has recently been included in the Australian Curriculum: Technology as a way to "re-engage" students in the sciences and potentially develop mathematical thinking. One example of a recent government initiative is the National Innovation and Science Agenda, which funds ($51 million) programs specifically dedicated to coding for primary school students (Department of the Prime Minister and Cabinet, 2016). The convergence of policy and curriculum directions is heartening; however, it is also highly problematic as there is a limited evidence base to inform the implementation of STEM in classrooms (English, 2016). We were particularly concerned with: How does the use of coding in primary school classrooms support, or provide opportunity for, the learning of mathematics? Using the coding program Scratch, we will discuss the ways that working with such programs provide opportunities to develop mathematical thinking. Here, we present the first lesson in a teaching experiment, where students used Scratch to draw a square.

Literature Review

Scratch is a visual programming language developed by the Lifelong Kindergarten group at MIT Media Labs (Resnick et al., 2009). Designed for students eight years of age and older, Scratch promotes creative thinking, reasoning, and innovation for those who engage with the program (Resnick et al., 2009). The rich digital environment utilises building block command structures to manipulate graphics, audio and video functions (Calder, 2012). The building block commands are a form of simplified syntax, so students are not required to type the code themselves; rather, they drag and drop the interlocking blocks of symbolic code together to create chains of code. There are 10 categories of building block command structures action, and each is represented by a specific colour. Examples of these colour categories include: motion blocks (blue); logic/control blocks (gold); and, data blocks (orange) (Francis, Khan, & Davis, 2016). Each coding block

(2017). In A. Downton, S. Livy, & J. Hall (Eds.), *40 years on: We are still learning! Proceedings of the 40th Annual Conference of the Mathematics Education Research Group of Australasia* (pp. 381-388). Melbourne: MERGA.

within a category has text and symbolic commands to assist the user to select the appropriate code for the action that they would like to undertake. Similar to the LOGO turtle (Papert, 1980), the interface enables a cat to move on a two-dimensional screen. Figure 1 presents the Scratch interface with an example of code for drawing a hexagon.



```
when ⚑ clicked
pen down
repeat 6
    move 100 steps
    turn ↻ 60 degrees
```

Figure 1. Scratch interface.

Studies Focusing on Coding

Research indicates that computer programming (henceforth coding) provides an opportunity for developing students' cognition and mathematical knowledge (Papert, 1980). Noss and Hoyles (1996) state that "writing a computer program provides a broad canvas on which the learner can sketch half-understood ideas, and assemble on the screen a semi-concrete image of the mathematical structures he or she is building intellectually" (p. 55). Research into the teaching and learning of mathematics through coding and programmable robots, including the use of LOGO (Clements, Battista, & Sarama, 2001) and Beebots (Highfield, 2010), have indicated that programmable robots support students in exploring problem solving, measurement, geometry and spatial concepts (Savard & Highfield, 2015). In addition, findings from quantitative studies have revealed that there is a correlation between computer coding using Scratch and mathematics test grades for Year 5 students (Lewis & Shah, 2012). With the exception of research conducted using LOGO, much of this research is in its infancy. Many studies examined the use of Scratch in middle school but few examined the use of Scratch in lower primary years. Finally, Benton, Hoyles, Kalas, and Noss (2017) stress that much of the past research into the impact of coding on students' mathematics acquisition is inconclusive; and, due to the diversity of adopted research paradigms across these studies, it is difficult to compare the results.

During the period 1970-2000, there were pockets of enthusiasm regarding the teaching of coding (e.g., LOGO and BASIC); however, there were several factors hindering its wider application: (a) many students had difficulty mastering the syntax of the program, (b) programing had little connection to young people's interests (e.g., generating lists of prime numbers), and (c) coding was introduced when there were few experts who could provide the students guidance (Resnick et al., 2009). It could be argued that the last point still resonates in the current educational climate. Many generalist primary school teachers are underprepared to teach coding and therefore will likely have difficulty in establishing links between coding and the teaching and learning of mathematics (Benton et al., 2017). The underlying mathematics that students engage in when coding can be unseen by teachers who often focus on the use of the tool (visual coding program or robots) rather than the mathematics within the tasks (Savard & Highfield, 2015). Few studies focus on the classroom implementation of coding and the curriculum (Lye & Koh, 2014). Coding appears in the Australian Curriculum areas of Mathematics and Technology (Australian Curriculum, Assessment and Reporting Authority [ACARA], 2016a, 2016b). The relationship between these documents is important to teachers who use the curriculum to plan, teach and evaluate student learning. Making the mathematics in coding apparent to teachers in curriculum documents is essential.

Linking Mathematics and Coding in the Australian Curriculum

Coding is explicitly embedded within the Digital Technologies (DT) strand of the Technology Curriculum and given that this paper concerns coding; we will limit discussion only to this component. The DT curriculum outlines that students will use "computational thinking and information systems, to define, design and implement digital solutions" (ACARA, 2016a). The DT curriculum in primary school is divided into three bands (F-2, 3-4, and 5-6) that are further subdivided into content descriptors under the sub-headings of Knowledge and Understanding and Processes and Production Skills. When examining the ways in which the DT curriculum builds opportunity for development of mathematical thinking, its alignment with the Mathematics curriculum is an important consideration. As we are solely concerned here with using Scratch with Year 2 students, the relevant Year 2 Mathematics and DT descriptors are displayed in Table 1.

Table 1
Mathematics Curriculum and Digital Technologies Curriculum

Subject	Content Description	Elaboration
Mathematics Year 2: Geometry	Describe and draw two-dimensional shapes, with and without digital technologies (ACMMG042)	Identify key features of squares, rectangles, triangles, kites, rhombuses and circles, such as straight lines or curved lines, and counting the edges and corners.
Digital Technologies Foundation – Year 2: Process and skill production	Follow, describe and represent a sequence of steps and decisions (algorithms) needed to solve simple problems (ACTDIP004)	Experimenting with very simple, step-by-step procedures to explore programmable devices, for example providing instructions to physical or virtual objects or robotic devices to move in an intended manner, such as following a path around the classroom.

The learning sequence in this study was designed to address the content descriptors in Table 1. The research questions explored in this paper are:

1. In what ways does working with coding contexts such as Scratch provide opportunity to develop mathematical thinking?
2. How do these opportunities align with the Australian Curriculum content descriptors for Mathematics and Digital Technologies?

Research Design

Participants

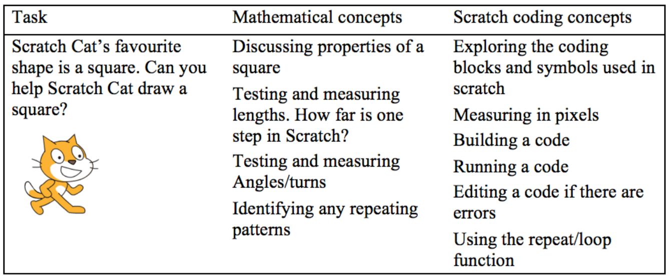
Six classes of Year 2 students (7-8 years old) from two schools located in Brisbane participated in the study. In total, there were 153 Year 2 students: 74 students from School A and 79 students from School B. Both schools were matched for socio-demographic characteristics. Both schools are just above the median for socio-educational advantage (School A = 1,056; School B = 1,037; ICSEA median value = 1,000).

Methodology: Teaching Experiment

A six-week coding and robotics teaching experiment was conducted with Year 2 students. The aim of the teaching experiment was to explore how students developed mathematical knowledge and thinking as they participated in coding and robotics lessons. Teaching experiments were used in this study for the primary purpose of directly experiencing students' mathematical learning and reasoning in relation to their construction of mathematical knowledge (Steffe & Thompson, 2000). One of the researchers (Author 1), in consultation with the class teacher, assumed the role of teacher in these experiments at both school sites. The teaching experiment comprised of: (a) pre-testing, (b) 6 x 45 minute lessons of either coding or robotics lessons (one lesson per week for six weeks with two groups of 10 students at each school site), and (c) post-testing.

All students participated in pre-testing measures at the commencement of the teaching experiment to identify their prior patterning knowledge and coding knowledge. As coding and mathematical patterning are related, it was decided to test the students on these two constructs. The test data are not the focus of this paper, however, do explain how students were selected for the study. This was a pen and paper test that focused on patterning (10 items) and coding from Scratch contexts (10 items). All test items were read to the students, and the test took approximately 30 minutes to complete. The items from these tests were developed from previously trialled patterning test items (Miller, 2015; Warren & Cooper, 2008) and then modified for coding contexts. Data from the pre-tests were analysed to determine a smaller, experimental group of students (n = 40) to participate in the coding and robotics lessons. Students were selected on their prior knowledge of patterning and coding (low-mid-high test scores). Four subgroups of students were identified: low patterning/low coding, low patterning/mid coding, mid patterning/low coding, and mid patterning/mid coding. No students were classified as high in either pattering or coding. Each subgroup consisted of 10 students with an even number of male and female students in each. Students not selected for the study (n = 113 spread over the six participating classes) stayed with their classroom teacher and participated in normal class lessons as planned by their teacher for that time. These teachers (n = 6) did not teach robotics and coding in their classrooms during the experiment. At the conclusion of the six weeks, post-testing (patterning and coding test) was conducted with all students (n = 153).

The teaching experiment consisted of six lessons, three with a coding focus using Scratch and three with a robotics focus using LEGO Mindstorm robots. Each lesson focused on teaching a mathematical concept using coding or robotics (e.g., drawing a square, drawing spirolaterals, moving a robot along a particular path). In this paper, we only present findings from the first lesson in the teaching experiment, where students were required to use the Scratch program to draw a square (See Figure 2). This lesson was aligned to the Year 2 Mathematics and DT Curriculums (see Table 1).



Task	Mathematical concepts	Scratch coding concepts
Scratch Cat's favourite shape is a square. Can you help Scratch Cat draw a square?	Discussing properties of a square Testing and measuring lengths. How far is one step in Scratch? Testing and measuring Angles/turns Identifying any repeating patterns	Exploring the coding blocks and symbols used in scratch Measuring in pixels Building a code Running a code Editing a code if there are errors Using the repeat/loop function

Figure 2. Drawing a Square task using Scratch with links to mathematics and coding concepts.

Two video cameras were used to collect data during each lesson of each teaching experiment, with one camera focussed on the researcher and one on a group of students. These video-recordings were used for in-depth analysis by the authors.

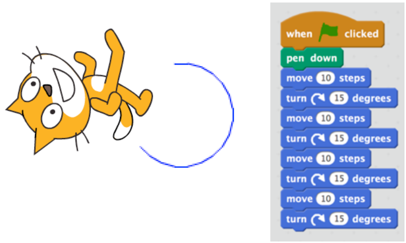
Data Analysis

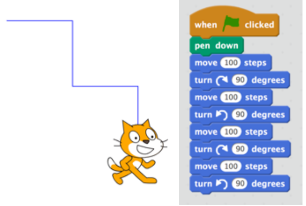
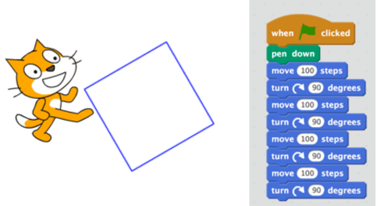
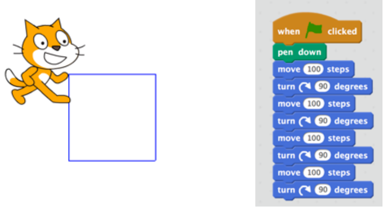
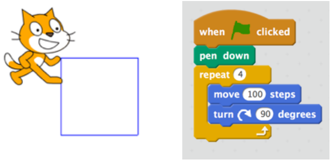
An iterative approach, using iterative refinement cycles for videotape analyses of changes in students' thinking, was adopted to analyse the data from the teaching experiment lessons (Lesh & Lehrer, 2000). Due to the unique application of mathematics, coding and robotics, this data-analysis model, utilised in prior studies (Miller, 2015), comprises two key stages. First, the lesson video-footage was transcribed to capture students' verbal responses. These transcriptions were then analysed to consider emerging mathematical thinking evident during the lessons. Second, the data were analysed to align the curriculum descriptors with student responses to the coding and robotics lessons.

Findings and Discussion

The findings are presented in two sections. Firstly, the emerging themes of the students' response to the "Draw a Square" Task are discussed. Secondly, the alignment of the task against the Australian Curriculum Mathematics and Digital Technologies descriptors is reviewed. Each of the 40 students provided a response to the task. After analysis of the student responses, it was evident that there were five common types of responses to the "Draw a Square" Task. Table 2 displays the student approaches, the frequency of the types of responses, and an example of a student's work fitting this type.

Table 2
Student Response and Explanation of Approach, Frequency of Student Responses, and Example of Students' Work

Response and Explanation	Frequency	Example
I can't draw a square but I can draw a hexagon Student attempted to draw a square but used 15 degree turns. While this does not draw a hexagon, the majority of the Year 2 students articulated they were creating a hexagon. They clicked on the code four times to make this shape.	8	 when ⚑ clicked pen down move 10 steps turn ↻ 15 degrees move 10 steps turn ↻ 15 degrees move 10 steps turn ↻ 15 degrees move 10 steps turn ↻ 15 degrees

Response and Explanation	Frequency	Example
I can draw stairs: Why is the cat not turning the right way? Students did not construct a square as they alternated the turns to the right and left.	4	 when ⚑ clicked pen down move 100 steps turn ↻ 90 degrees move 100 steps turn ↺ 90 degrees move 100 steps turn ↻ 90 degrees move 100 steps turn ↺ 90 degrees
Is this still a square? Students correctly coded a square but were unsure whether it was a square or not because of the orientation.	3	 when ⚑ clicked pen down move 100 steps turn ↻ 90 degrees move 100 steps turn ↻ 90 degrees move 100 steps turn ↻ 90 degrees move 100 steps turn ↻ 90 degrees
I made a square. Students were able to write a code to make the Scratch cat draw a square parallel to the bottom of the screen.	20	 when ⚑ clicked pen down move 100 steps turn ↻ 90 degrees move 100 steps turn ↻ 90 degrees move 100 steps turn ↻ 90 degrees move 100 steps turn ↻ 90 degrees
I can see a pattern: move, turn, move, turn, move, turn…. Students identified that they can see a repeating pattern and used the repeat coding block to draw a square.	5	 when ⚑ clicked pen down repeat 4 move 100 steps turn ↻ 90 degrees

When considering the above responses, in relation to the content requirements of the Year 2 Mathematics curriculum (see Table 1), it is evident that some students were working at a higher level than required. There were three key insights from the data that demonstrated higher levels of mathematical thinking: (a) working with 90 degree turns, (b) orientation and perspective taking, and (c) deducing a repeating pattern to provide a generalised code for making a square. When drawing a two-dimensional shape, using a digital tool such as Scratch, it is conjectured that this provides the opportunity for students to engage with higher levels of mathematics. This may occur for three reasons: first, as a consequence of the visual programing language (icons) and representations; second, the perspective which performing the task requires; and third, using a coding chain that represents the structure of the mathematical shape the students have drawn.

When using Scratch to draw a square, it appears the language and representations depicted in the coding blocks require higher levels of mathematical thinking and knowledge for these young students. For example, Scratch uses measures of degrees for turns, rather than language such as 1/4 turn. This program appears to offer more opportunities to explore some aspects of mathematics (e.g., measurement of angles) than programmable robotics toys, such as the commonly used Beebots, that have an arrow indicating left or right which results in the Beebot performing a 90 degree turn. When using Scratch students have to determine the number of degrees themselves in order to perform an accurate 90 degree turn. Aligning this mathematical thinking to the Year 2 Mathematics curriculum, students are only required to identify the corners of shapes and use the language of "quarter and half turns". It is not until students are in Year 5 that they are required to estimate and measure angles using degrees (ACARA, 2016b). Our initial supposition is that the representations and language in Scratch supports the development of higher levels of mathematical thinking for these young students.

Second, the way in which students engage in the task of drawing a square using Scratch is vastly different to drawing a square on paper using a pencil and ruler. As students code the Scratch cat to draw a square, they are required to take the perspective of the cat (i.e., the square will be drawn in the same orientation as the cat is initially facing). Students who started their cat either facing up or down, or an alternative sideways orientation other than the cat facing directly left or right of the screen (student's perspective while looking at the screen), meant that the square could be drawn from different initial perspectives and thus end up looking different to the prototypical depictions of squares (parallel to bottom and sides of the screen). This was evident in the case where students were unsure if they had still drawn a square for the reasons outlined above. While they could code a square, their limited mathematical understanding of "squareness" meant they were unable to reason if their shape was a square or not.

Finally, unlike drawing a square on paper using a pencil and ruler, some students could see on the screen the "structure", that is a semi-concrete mathematical structure (Noss & Hoyles, 1996), of their drawing in the Scratch code. This led to five students, unprompted by the researcher, identifying units of repeat (e.g., move 100 steps, turn 90 degrees) and deducing that their code (repeat four times – move 100 steps, turn 90 degrees) would draw a square. While, students in Year 2 should be able to identify a repeating pattern, this moves beyond the typical patterns presented to students (e.g., ABAB). This led to students then deducing a generalisation for the perimeter (e.g., move *n* length, turn 90 degrees and repeat four times) of the square and even further to discussions about measuring the perimeter of squares using the code (e.g., If my square has a length of 10, the total perimeter will be 40). We suggest that these students were demonstrating, and engaging in, early algebraic thinking (deducing patterns) and higher levels of measurement (identifying the perimeter), beyond the current required curriculum standard, such as Year 6 – Continue and create sequences involving whole numbers, fractions and decimals. Describe the rule used to create the sequence (ACMNA133) and Year 5 – Calculate perimeter and area of rectangles using familiar metric units (ACMMG109) (ACARA, 2016b).

Conclusion

The implementation of coding into the lower primary years presents a challenge for generalist primary school teachers. Commonly, past research in this area has provided insights into how students in the middle years of schooling work with coding contexts, but at times the impact on mathematics acquisition for these students are inconclusive (Benton et al., 2017). This study adds to the current literature by examining the use of Scratch in a lower primary context, and identifying the types of mathematical thinking these students engaged with while undertaking the task. Research with primary school students, when using robotics programs, have identified that young students use problem solving, measurement, geometry and spatial concepts (Savard & Highfield, 2015), in these contexts. Our early conjecture is that coding also provides an opportunity to identify and deduce patterns and therefore is a platform to engage with early algebraic thinking.

With few studies focusing on the classroom and curriculum implementation of coding (Lye & Koh, 2014), there is a need to examine the relationship between the Mathematics and Digital Technologies Curricula and coding contexts to maximise learning opportunities for primary students. As teachers use the curriculum to manage their planning, teaching and evaluating of student learning, making the mathematics in coding apparent for teachers within both curriculum documents is essential. Although this is an initial, small-scale study over a relatively short intervention, it begins to indicate the potential of coding programs such as Scratch to support students' mathematical thinking and concept development.

References

Australian Academy of Science. (2016). *The mathematical sciences in Australia: A vision for 2025*. Canberra: Author.

Australian Curriculum, Assessment and Reporting Authority. (2016a). *The Australian curriculum: Digital technologies*. Canberra: Author.

Australian Curriculum, Assessment and Reporting Authority. (2016b). *The Australian curriculum: Mathematics*. Canberra: Author.

Benton, L., Hoyles, C., Kalas, I., & Noss, R. (2017). Bridging primary programming and mathematics: Some findings of design research in England. *Digital Experiences in Mathematics Education*.

Calder, N. (2012). The layering of mathematical interpretations through digital media. *Educational Studies in Mathematics*, *80*(1-2), 269-285.

Clements, D. H., Battista, M. T., & Sarama, J. (2001). Logo and geometry. *Journal for Research in Mathematics Education Monograph*. Reston, VA: National Council of Teachers of Mathematics.

Department of the Prime Minister and Cabinet. (2016). *National innovation and science agenda report.* Canberra: Commonwealth of Australia.

English, L. D. (2016). STEM education K-12: Perspectives on integration. *International Journal of STEM Education*, *3*(3). doi:10.1186/s40594-016-0036-1

Francis, K., Khan, S., & Davis, B. (2016). Enactivism, spatial reasoning and coding. *Digital Experiences in Mathematics Education*, *2*(1), 1-20.

Highfield, K. (2010). Robotic toys as a catalyst for mathematical problem solving. *Australian Primary Mathematics Classroom*, *15*(2), 22-27.

Larkin, K., & Jorgensen, R. (2016). 'I hate maths: Why do we need to do maths?' Using iPad video diaries to investigate attitudes and emotions towards mathematics in Year 3 and Year 6 students. *International Journal of Science and Mathematics Education*, *14*(5), 925-944.

Lesh, R., & Lehrer, R. (2000). Iterative refinement cycles for videotape analyses of conceptual change. In R. Lesh & A. Kelly (Eds.), *Research design in mathematics and science education* (pp. 665-708). Hillsdale, NJ: Erlbaum.

Lewis, C. M., & Shah, N. (2012). Building upon and enriching Grade 4 mathematics standards with programming curriculum. In *Proceedings of the 43rd Association of Computing Machinery technical symposium on Computer Science Education* (pp. 57-62). Raleigh, NC: ACM.

Lye, S. Y., & Koh, J. H. L. (2014). Review on teaching and learning of computational thinking through programming: What is next for K-12? *Computers in Human Behavior*, *41*, 51-61.

Miller, J. (2015). Young Indigenous students' engagement with growing pattern tasks: A semiotic perspective. In M. Marshman, V. Geiger, & A. Bennison (Eds.), *Proceedings of the 38th Annual Conference of the Mathematics Education Research Group of Australasia* (pp. 421-428). Sunshine Coast, QLD: MERGA.

Noss, R., & Hoyles, C. (1996). *Windows on mathematical meanings: Learning cultures and computers*. London, England: Kluwer.

Office of Chief Scientist. (2014). *STEM: Australia's future.* Canberra: Author.

Papert, S. (1980). *Mindstorms: Children, computers, and powerful ideas*. New York, NY: Basic Books.

Resnick, M., Maloney, J., Monroy-Hernández, A., Rusk, N., Eastmond, E., Brennan, K... Kafai, Y. (2009). Scratch: Programming for all. *Communications of the ACM*, *52*(11), 60-67.

Savard, A., & Highfield, K. (2015). Teachers' talk about robotics: Where is the mathematics? In M. Marshman, V. Geiger, & A. Bennison (Eds.), *Proceedings of the 38th Annual Conference of the Mathematics Education Research Group of Australasia* (pp. 540–546). Sunshine Coast, QLD: MERGA.

Steffe, L. P., & Thompson, P. W. (2000). Teaching experiment methodology: Underlying principles and essential elements. In R. Lesh & A. E. Kelly (Eds.), *Research design in mathematics and science education* (pp. 267–307). Hillsdale, NJ: Lawrence Erlbaum Associates.

Warren, E., & Cooper, T. (2008). Patterns that support early algebraic thinking in the elementary school. In C. Greenes & R. Rubenstein (Eds.), *Algebra and algebraic thinking in school mathematics: Seventieth Yearbook* (pp. 113–126). Reston, VA: National Council of Teachers of Mathematics.