Utah State University

DigitalCommons@USU

All Graduate Theses and Dissertations | Graduate Studies

5-2023

Connections Between Mathematics and Computational Thinking: Kindergarten Students' Demonstration of Mathematics Knowledge in a Computational Thinking Assessment

Lise E. Welch Bond
Utah State University

Follow this and additional works at: https://digitalcommons.usu.edu/etd

Part of the Early Childhood Education Commons

Recommended Citation

Welch Bond, Lise E., "Connections Between Mathematics and Computational Thinking: Kindergarten Students' Demonstration of Mathematics Knowledge in a Computational Thinking Assessment" (2023). *All Graduate Theses and Dissertations*. 8737.
https://digitalcommons.usu.edu/etd/8737

This Dissertation is brought to you for free and open access by the Graduate Studies at DigitalCommons@USU. It has been accepted for inclusion in All Graduate Theses and Dissertations by an authorized administrator of DigitalCommons@USU. For more information, please contact digitalcommons@usu.edu.

CONNECTIONS BETWEEN MATHEMATICS AND COMPUTATIONAL THINKING: KINDERGARTEN STUDENTS' DEMONSTRATION OF MATHEMATICS KNOWLEDGE IN A COMPUTATIONAL THINKING ASSESSMENT

by

Lise E. Welch Bond

A dissertation submitted in partial fulfilment
of the requirements for the degree

of

DOCTOR OF PHILOSOPHY

in

Education

Approved:

Jessica F. Shumway, Ph.D. Major Professor	Patricia S. Moyer-Packenham, Ph.D. Committee Member
Jody Clarke-Midura, Ed.D. Committee Member	Beth MacDonald, Ph.D. Committee Member
Katherine Vela, Ph.D. Committee Member	D. Richard Cutler, Ph.D. Vice Provost for Graduate Studies

UTAH STATE UNIVERSITY
Logan, Utah

2023

Copyright © Lise E. Welch Bond 2023

All Right Reserved

ABSTRACT

Connections Between Mathematics and Computational Thinking: Kindergarten Students' Demonstration of Mathematics Knowledge in a Computational Thinking Assessment

by

Lise E. Welch Bond, Doctor of Philosophy

Utah State University, 2023

Major Professor: Jessica F. Shumway, Ph.D.
Department: Mathematics Education and Leadership

Research has shown that computational thinking and kindergarten mathematics instruction can be integrated; however, evidence of how specific mathematical knowledge relates to computational thinking remains scarce. Additionally, we do not know if and how children's mathematical knowledge co-occurs with computational thinking and how these knowledges relate to students' performance on computational thinking assessments. This qualitative study sought to fill this knowledge gap by examining the following research questions through a joint embodied cognition and enactivist lens: (1) How are kindergarten students' mathematical knowledge (MK) and computational thinking (CT) operationalized during a CT assessment? In what ways, if any, do MK and CT co-occur, and (2) How do students' mathematical knowledge and co-occurring mathematical knowledge and computational thinking relate to their performance on individual assessment items?

Using a dataset collected for a larger research study (NSF project award #DRL-1842116), I analyzed video of 60 kindergarten students engaging with 14 items in an interview-based, computational thinking assessment. I coded and memoed the data to operationalize how students demonstrate mathematical knowledge and computational thinking, then analyzed the coded data to identify co-occurring knowledge. Lastly, I developed case studies to describe how students' knowledge related to their assessment item performance.

Results indicate that students demonstrate varying levels of mathematical knowledge and computational thinking multi-modally through their gestures, language, and actions with the assessment materials. Students' spatial and unit measurement knowledge most frequently co-occurred with computational thinking, occurring most often when students built and read/enacted programs. These co-occurrences were categorized as independent or dependent, depending on the nature of their relationship to the computational thinking outcomes. These findings illustrate the intricate connections between mathematical knowledge and computational thinking and that students' mathematical knowledge relates to their performance on computational thinking tasks. These findings have implications for computational thinking curriculum and assessment design, mathematics curriculum design, and theory. Based on the results of this present study, I recommend that mathematics curriculum developers leverage the spatial and unit measurement connections in computational thinking tasks to design experiences for children to grow their spatial reasoning and measurement knowledge.

(228 pages)

PUBLIC ABSTRACT

Connections Between Mathematics and Computational Thinking: Kindergarten Students' Demonstration of Mathematics Knowledge in a Computational Thinking Assessment

Lise E. Welch Bond

Research shows that computational thinking can be used with kindergarten mathematics instruction, however we still do not know much about how specific math knowledge is related to computational thinking and if (and if so, how) children's mathematical knowledge is related to students' performance on computational thinking assessments. This student fills this knowledge gap by examining the following research questions: (1) How are kindergarten students' mathematical knowledge (MK) and computational thinking (CT)MK and CT operationalized during a CT assessment? In what ways, if any, do MK and CT co-occur, and (2) How do students' mathematical knowledge and co-occurring mathematical knowledge and computational thinking relate to their performance on individual assessment items?

To answer these questions, I analyzed video data that was originally collected for a larger research study (NSF project award #DRL-1842116), which showed 60 kindergarten students taking an interview-based, computational thinking assessment. I coded and notated the data to describe how students demonstrate their mathematical knowledge and computational thinking, then analyzed the coded data to identify how students' mathematical knowledge and computational thinking co-occurred. Lastly, I described how, for four assessment items, students' co-occurring knowledge related to their assessment item performance.

The results show that students demonstrated different levels of mathematical knowledge and computational thinking through their gestures, language, and interactions with the assessment materials. Students' spatial and unit measurement knowledge most frequently co-occurred with computational thinking, and most often when students built and read/enacted programs. I categorized the co-occurrences as independent or dependent, depending on if the co-occurrence related to the students' correct or incorrect response to the assessment items. These findings show that mathematical knowledge and computational thinking are strongly connected, and that students' mathematical knowledge is related to how they performed on the assessment. These findings have implications for computational thinking curriculum and assessment design, mathematics curriculum design, and theory. Based on the results of this present study, I recommend that mathematics curriculum developers take advantage of the particularly strong connections of spatial and unit measurement knowledge with computational thinking to design experiences for children develop their spatial reasoning and measurement knowledge.

ACKNOWLEDGMENTS

I would like to express gratitude to my committee members, Drs. Patricia Moyer-Packenham, Jody Clarke-Midura, Beth MacDonald, and Katherine Vela, for their inexhaustible time, support, and expertise. Thank you all for caring for me, not just as a student but as an individual. I would like to especially thank my chair, Dr. Jessica Shumway, for supporting me through the many ups and downs and pushing me to greater heights. You saw a spark of potential in me and nurtured that spark to a flame. Thank you for your patience and steady support; I will forever treasure your mentorship.

I am also grateful to the Coding in Kindergarten research team PIs (CiK; Drs. Jody Clarke-Midura, Jessica Shumway, Victor Lee) and post-doctoral researcher (Dr. Deborah Silvis) of NSF award DRL#1842116. My experience with the Coding in Kindergarten team molded me into a better researcher and allowed me to press forward during the uncertainty of Covid's possible disruption. Special thanks to the students and teachers who made this research possible. Classroom time is precious, so your willingness to share that time with us is a treasured gift.

To my family, thank you. It is an honor to follow in the academic footsteps of my grandfather, Gerry H Turner, and great-grandfather, J. Whitney Floyd. When discussing the prospect of pursuing a Ph.D., my father's response of "Why not?" made the journey seem possible. Thank you, dad, for spurring me on. Thank you, mom, for always lending a listening ear. Thank you, Jenny, Jamie, and Angie, for being the loving sisters that most people can only dream of.

Finally, thank you, Jeremy, my relentlessly supportive husband. This journey has been better with you by my side.

Lise E. Welch Bond

CONTENTS

LIST OF TABLES

Page

LIST OF FIGURES

Page

CHAPTER I

INTRODUCTION

Extensive effort has been made within the U.S. to increase computer science education opportunities for K–12 students (Smith, 2016). One avenue of integrating computer science education is teaching computational thinking (CT) skills, a thinking process foundational to computer science (Wing, 2006). A potential benefit of integrating CT in K–12 classrooms is growing evidence that CT holds deep connections and co-occurrences with mathematical knowledge (MK; e.g., Pérez, 2018; Rich et al., 2019; Shumway et al., 2021). Coding is one instructional medium used to explore young children's engagement with CT and MK (Angeli & Valanides, 2020).

Contemporary research has examined young children's engagement in curricular-based coding activities with joint MK and CT objectives (e.g., Angeli & Valanides, 2020; Città et al., 2019; Palmér, 2017; Shumway et al., 2021; Strawhacker & Bers, 2019; Welch et al., 2022). While these studies describe general commonalities and co-occurrences between MK and CT, fine-grained research exploring how specific MK co-occurs with and/or supports CT in kindergarten is limited. Research groups have only recently explored how specific MK is exhibited by young students in curricular CT robot coding tasks (Shumway et al., 2021). The present study extends on Shumway and colleagues' work by investigating how kindergarten students use MK with CT in an individual assessment-interview setting with unplugged (i.e., no robot coding toys or online coding tools) materials. The purpose of the present study was to understand how kindergarten students' MK relates to their CT. I investigated these MK CT connections and relationships by identifying specific MK that kindergarten students demonstrated while solving CT assessment items and how the students' MK and the co-occurrences of MK CT knowledge related to their CT assessment performance.

Background to the Problem

Computational Thinking's Origins in Mathematics Education

While CT is typically associated with computer science, CT's roots are steeped in mathematics education. CT was first defined by Wing in 2006 as "solving problems, designing systems, and understanding human behavior, by drawing on the concepts fundamental to computer science" (p. 33). CT's origins as a potential literacy, however, date back to Papert (1972), who described algorithmic thinking, debugging, decomposition, and abstraction (classified today as CT components) as ways of mathematical thinking. Papert (1980) expanded on these ideas in his seminal book *Mindstorms* and introduced constructionism, a computational theory. Papert used constructionism to describe how children can learn mathematics using Logo, a simplified computer programming language in which an individual can code a digital turtle's two-dimensional movement. He described Logo as an object-to-think-with, and—while acknowledging the limitations of 1970s technology—envisioned technologies that would evolve to provide increasingly rich environments to engage children in mathematical problem solving. Papert informally coined the term computational thinking in *Mindstorms*, and again in a 1996 paper discussing mathematics education using Logo (Papert, 1996).

Defining Computational Thinking

CT has been defined in different ways based on the context in which it is used. Many researchers draw on the original definition proposed by Wing (2006), however more recent definitions and frameworks have evolved through efforts to further understand and define CT (e.g., Ehsan & Cardella, 2017; Weintrop et al., 2016; Yadav et al., 2014). For the present study, I adopted a definition proposed by Shute et al. (2017), defining CT as “the conceptual foundation required to solve problems effectively and efficiently (i.e., algorithmically, with or without the assistance of computers) with solutions that are reusable in different contexts” (p. 151).

While Shute et al. (2017) provide an adequately generalized definition, it is important to acknowledge that CT is conceptualized uniquely across age groups and contexts (e.g., Bers et al., 2019; Lye & Koh, 2014; Shute et al., 2017; Yadav et al., 2014). This necessitates adopting a description of CT specific to the present study’s interest in early childhood mathematics. The data for this study draws on existing data from a larger study called Coding in Kindergarten (CiK; Grant no. NSF #1842116). One of CiK’s primary objectives is to operationalize CT in the context of children’s mathematics and programming in kindergarten curriculum and assessments (Clarke-Midura, Shumway, et al., 2021). CiK’s efforts to fulfill this objective remain ongoing, however preliminary findings indicate that CT constructs central to kindergarten CT coding tasks include algorithmic thinking, debugging, and decomposition with strong spatial reasoning influences (Clarke-Midura, Shumway, et al., 2021; Clarke-Midura, Silvis et al., 2021; Shumway et al., 2021). So, in addition to Shute et al.’s (2017) definition, I will adopt the aforementioned constructs to describe CT in kindergarten.

Re-Rooting Computational Thinking in Present Mathematics Education Research

Early childhood research extending on Papert's vision of mathematics education using Logo through the late 1990s yielded mixed results on Logo's impact for general mathematics instruction, however Logo-integrated instruction in geometry and spatial sense—specifically—had positive results (Clements & Sarama, 1997). While researchers during this period did not have the benefit of CT definitions and competencies, students still had to engage with CT to successfully use Logo. For example, using Logo requires students to sequence and enter programming code into a computer to achieve an objective. Sequencing and writing code are an important part of the CT skill algorithmic thinking. Students also use CT skills to break apart tasks (decomposition) and fix problems that arise (debugging). So, while Logo-era mathematics education researchers did not explicitly examine CT, CT ideas and constructs were an undercurrent of Logo-era research and likely influenced children's engagement with Logo.

Increased access to technology such as concrete coding toys (Hamilton et al., 2020) as well as a recent emphasis on implementing CS instruction in K–12 (Smith, 2016) have revived research interests in mathematics education through coding. The increasingly available technologies of today could, arguably, fulfill Papert's (1980) vision of more effective objects-to-think-with to engage children in mathematics problem solving.

Recent research on CT integration with early childhood mathematics instruction has examined the mathematical concepts of patterning (Miller, 2019; Saxena et al., 2020), sequencing (Angeli & Valanides, 2020; Città et al., 2019), decomposition (Lavigne et al., 2020; Rijke et al., 2018), measurement units (Solomon et al., 2015; Sung et al., 2017; Welch et al., 2022), and spatial skills (Moore et al., 2020; Palmér, 2017; Strawhacker & Bers, 2019). While these research examples paint an important picture of how CT integration with mathematics can manifest in early childhood classrooms, these studies provide very broad perspectives of how MK and CT might be integrated and support each other. What is lacking is an improved understanding of the deep connections between MK and CT that can support a meaningful integration of these two domains. Deep connections between MK and CT refer to how these domains' constructs and skills emerge independently and in tandem and can be observed via intentional and empirical evidence-based examination. While research has recently begun to unearth these MK CT connections (Shumway et al., 2021), these efforts are still in their infancy. A recent review of CT studies in K–12 confirms that studies explicitly linking MK and CT is lacking, more specifically research conducted by education researchers (Hickmott et al., 2018). Identifying and understanding the specific connections between MK and CT from an educational research standpoint can provide important direction for future curricular integration.

Additionally, methods for measuring CT in kindergarten are still evolving, as the field is still operationalizing how CT emerges in this population (Clarke-Midura, Shumway et al., 2021). Very few validated CT assessments are currently available. In a review of empirical research on CT assessments, Tang et al. (2020) reported that only

45% of the studies reported on reliability measures, and only 18% on evidence of validity. Since assessment is a way of operationalizing domains, and surface connections have been repeatedly made between CT and MK, it would be prudent that CT assessments for kindergarten consider the MK characteristics of early childhood populations to accurately measure a student's CT ability while attending to related MK appropriate for kindergarten-aged students.

Given that the deeper connections between MK and CT are still unknown and that operationalizing CT within an assessment context is still emerging, I argue that a necessary step towards understanding the connections between CT and mathematics education is to explore the relationship between MK and CT constructs within a CT assessment context where children independently demonstrate their knowledge of algorithmic thinking, decomposition, debugging, and spatial reasoning.

Problem Statement

While research has documented how CT can be integrated in kindergarten mathematics instruction, the evidence informing the field of how specific MK relates to CT is still scant. Further, we do not know if, and if so how, children's MK and MK CT co-occurrences might relate to their performance on CT tasks.

The present study sought to fill this gap by identifying children's use of specific MK while engaging in CT tasks. Knowing the specific MK that children indicate in interview-based CT assessments can contribute to an understanding of how specific MK relates to their CT knowledge, skills, and performance.

Study Purpose and Research Questions

The purpose of this study was to identify specific MK and CT that kindergarten students use during a CT assessment and to describe how MK and MK CT co-occurrences might relate to students' CT assessment performance. Two questions guided this study.

1. How are kindergarten students' MK and CT operationalized during a CT assessment? In what ways, if any, do MK and CT co-occur?
2. How do students' MK and MK CT co-occurrences relate to their performance on individual CT assessment items?

Significance of the Study

The present study's results have implications for both CT assessment and CT curricular design as well as early mathematics and computer science pedagogy. Awareness of how the specific MK that students demonstrate during CT tasks will assist researchers and curriculum developers addressing CT concepts to develop age-appropriate tasks for kindergarten-aged students. Researchers developing CT activities should be informed of the MK that the target demographic uses while completing CT tasks and consider how the expected MK might influence the demographic's access to CT tasks.

Mathematics pedagogy could likewise be informed by this study. These findings could inform mathematics teachers of the MK students will likely apply in CT coding tasks and how educators might integrate CT into regular instruction and capitalize on the opportunity to leverage the common threads connecting MK and CT.

Scope of this Study and Assumptions

This study's context was an unplugged, grid-based, agent-oriented CT assessment developed for kindergarten students. The assessment items directed students to use a coding system (i.e., arrows that represent directions to move the agent: forward, backward, right rotation, left rotation) to build and enact programs to direct an agent's movement in a grid space. This study assumed that, within the described context, MK and CT are made up of complex constructs that are observable—via an embodied and enactivist lens—through students' gestures, language, and actions on objects. Additionally, this study recognized that MK and CT co-occur in foundational ways which can impact students' task access and performance. Each CT assessment item was unique and required differing combinations of MK and CT skills to successfully complete it, and I anticipated that the variation of MK and CT skills would relate to students' assessment performance.

Definition of Terms

Mathematical Knowledge (MK): "A systematic and well-interconnected web of mathematical concepts and skills" (Purpura et al., 2013, p. 453). In other words, the knowledge part of this is emphasizing broadly the concepts and skills that an individual carries and exhibits through language, gestures, and actions on objects.

Computational Thinking (CT): "The conceptual foundation required to solve problems effectively and efficiently (i.e., algorithmically, with or without the assistance of computers) with solutions that are reusable in different contexts" (Shute et al., 2017, p.

151).

Algorithmic Thinking: "Developing and using ordered sequences of instructions," (Clarke-Midura, Shumway et al., 2021, p. 18).

Coding: The act of writing sequential instructions to achieve an objective.

Debugging: "Recognizing bugs/errors that exist, locating the specific error or bug, proposing a fix, and correcting the bug," (Clarke-Midura, Shumway et al., 2021, p. 18).

Decomposition: "Recognizing parts in part-whole relationships, building a whole from parts, and breaking a whole into parts," (Clarke-Midura, Shumway et al., 2021, p. 18).

Spatial Reasoning: "The ability to recognize and (mentally) manipulate the spatial properties of objects and the spatial relations among objects" (Bruce et al., 2017, p. 146).

CHAPTER II

LITERATURE REVIEW

This study's query is positioned in kindergarten students' MK as performed in a CT assessment, so this literature review reflects this context. First, I broadly summarize the empirical research on kindergarten MK and define and describe the mathematics that kindergarten students are expected to know and do. The MK section will also outline validated mathematics assessments to measure kindergarteners' MK, providing further evidence of how research operationalizes kindergarten MK. An overview of these assessments will provide a context for current mathematics assessment availability and practices in kindergarten.

Next, the research on CT in early childhood settings (preschool, kindergarten, first grade; ages 3–7) will be discussed. In this section, I will broadly consider early childhood beyond this study's kindergarten context, as research in this area is still new and reporting is limited. This section of the literature review will include the research on assessments of early-childhood CT, which will highlight the disparity in validated early-childhood CT assessments, providing a case for exploring these assessments as they are being developed.

The subsequent section will describe early-childhood research examining both MK and CT across early childhood settings (preschool, kindergarten, first grade; ages 3–7). This section will also bring attention to the lack of assessments for joint MK CT assessment.

This chapter's final sections describe the theoretical perspectives of embodied cognition and enactivism and present a conceptual framework highlighting documented MK and CT constructs. The presented conceptual framework guided the context for this study to examine the MK, CT, and MK CT co-occurrences that kindergarten students demonstrate during a CT assessment and how students' MK might relate to their CT assessment item performance.

Mathematical Knowledge in Kindergarten Classrooms

Mathematical knowledge is "a systematic and well-interconnected web of mathematical concepts and skills" (Purpura et al., 2013, p. 453). Children bring informal mathematics knowledge with them when they begin formal schooling which can influence children's integration and perception of formally presented mathematics concepts and skills (Purpura et al., 2013). Two generally recognized forms of MK are procedural knowledge and conceptual knowledge (Crooks & Alibali, 2014; Kadijevich, 2018). *Procedural knowledge* is "the knowledge of procedures… [that are a] series of steps, or actions, done to accomplish a goal" (Rittle-Johnson et al., 2015, p. 588). On the other hand, *conceptual knowledge* is understood as "knowledge of concepts, which are abstract and general principles" (Rittle-Johnson et al., 2015, p. 588). Supporting earlier arguments that procedural and conceptual knowledge are entangled (Baroody et al., 2007; Star, 2005), researchers are finding that these knowledge types can influence each other (Crooks & Alibali, 2014; Rittle-Johnson et al., 2015).

The MK that children are anticipated to develop in formal schooling is outlined in curriculum standards based in MK research (e.g., Common Core State Standards

Initiative [CCSSI], 2010; National Research Council [NRC], 1990). The most recent curriculum standards applied by schools across the U.S. are the *Common Core State Standards of Mathematics* (CCSSM; CCSSI, 2010). The CCSSM outlines curriculum standards for each grade, levels K–12. The kindergarten standards provide conceptual and procedural knowledge goals for students aged 5 and 6 years old. While the extent of these standards is outside the scope of this literature review, I focus, instead, on the two *critical areas* (main grade level objectives) outlined by the CCSSM kindergarten standards. These include "(1) representing and comparing whole numbers, initially with sets of objects; (2) describing shapes and space" (CCSSI, 2010, Kindergarten – Introduction section, para. 1). I also summarize the literature on kindergarten measurement learning as measurement can serve as an application to connect number, shapes, and space (Battista et al., 2017).

Representing and Comparing Whole Numbers

The first critical area in kindergarten is centered around understanding and operating with whole numbers. Kindergarteners are expected to count objects, compare values, and perform basic addition and subtraction operations (CCSSI, 2010). *Counting* is the process of listing number words in a sequence (National Research Council [NRC], 2009) and "is the first and most basic and important algorithm" (Clements & Sarama, 2009, p. 22). Many children enter kindergarten being able to count, having developed this skill either in preschool or informally in natural environments (Purpura & Baroody, 2013). Around ages 3 and 4, children demonstrate the ability to coordinate between number word lists and counting correspondence when coordinating each number word with a raised finger (Crollen & Noël, 2015) or counting with fingers by pointing to objects (Clements & Sarama, 2021). This process is a prime example of conceptual and procedural knowledge working in tandem. Counting with fingers and synchronizing counting with pointing to objects coordinates the procedural knowledge of a number word list with the conceptual and procedural understanding that one number word links with each object.

Research indicates that individual differences in children's counting predict later mathematics achievement (Nguyen et al., 2016) as does children's ability to count on (Wilkins et al., 2021) and represent quantities with numerals (Geary et al., 2018; Vanbinst & De Smedt, 2016). By the time children reach kindergarten, children extend upon their counting abilities to count and compare larger sets (accurately to 10; Clements & Sarama, 2021). In addition to representing quantities with number words and numerals, empirical studies show the importance of comparing quantities and numbers (Booth & Siegler, 2008; Clements & Sarama, 2021; De Smedt et al., 2009; Libertus et al., 2011). For example, De Smedt et al. conducted a 1-year study to explore the association between young children's number comparison and math achievement. In this longitudinal study of 47 children (mean age = 6.3 years) in a Belgium school, students were asked to identify the larger of two numbers. The number pair shown to students included possibilities ranging from 1 to 9. DeSmedt's research team concluded that students' number comparison abilities, and the speed in which the students compared numbers, predictively correlated with students' later mathematics achievement. Similar to De Smedt et al., Libertus et al. verified an association between quantity comparison and children's math ability. In Libertus et al.'s study examining 3- to 5-year-old children's ($N = 174$) quantity comparison knowledge, the researchers asked participants to compare two sets of dots and identify which set had the most. Study results revealed a correlation between the children's ability to compare quantities and their mathematical abilities.

Overall, kindergarten students in the U.S. are expected to be able to represent quantities symbolically (words and numerals) and compare values both pictorially and symbolically (Clements & Sarama, 2021; CCSSI, 2010). Further, children's counting, abstraction of quantity with numerals, and number/quantity comparison are important foundations for future mathematical achievement.

Describing Shapes and Space

Kindergarten curriculum attends to geometric ideas including shape and spatial reasoning (CCSSI, 2010). By the end of kindergarten, children should be able to identify two-dimensional (i.e., circle, square, triangle, rectangle) and three-dimensional (i.e., cone, cylinder, cube) shapes (Clements & Sarama, 2021) and begin to recognize shapes' *intrinsic* attributes (i.e., corner, edge, side, orientation) and *extrinsic* attributes (location relative to other objects; CCSSI, 2010; Newcombe & Shipley, 2015). Kindergarteners also combine shapes to construct more complex shapes (Clements & Sarama, 2021; CCSSI, 2010), which combines students' understanding of how intrinsic attributes and extrinsic attributes can interact.

Intrinsic and extrinsic characteristics of shapes and other objects entail aspects of spatial reasoning, an important thinking process in geometry (Battista et al., 2017). *Spatial reasoning* is "the ability to recognize and (mentally) manipulate the spatial properties of objects and the spatial relations among objects" (Bruce et al., 2017, p. 146). Spatial reasoning skills include *spatial orientation* (situating objects, points, and oneself within a space) and *spatial visualization* (mentally manipulating images). Spatial reasoning is considered an essential human skill (Clements & Sarama, 2021) and a fundamental element in mathematics (Clements & Sarama, 2011; Davis & the Spatial Reasoning Study Group, 2015; Mix et al., 2016).

Mix et al. (2016) conducted an extensive study of students in kindergarten (N = 275), third (N = 291), and sixth grade (N = 288) to examine spatial and mathematical skills among children in nine communities located in the Midwestern U.S. Results indicate that mental rotation was a reliable predictor of kindergarteners' mathematical performance, but mental rotation's value as a predictor appears to decline at third, and then sixth grade. Additionally, spatial skills were a predictor for kindergarten students' performance on new mathematical content, such as calculation. This suggests that, particularly in early childhood, spatial skills have a meaningful relationship with students' mathematical performance.

The CCSSM advises that a majority of instruction time be allocated toward number (CCSSI, 2010), however some researchers argue that spatial reasoning, part of kindergarten's second critical area, is under-emphasized in the U.S. (Clements & Sarama, 2011; Davis & the Spatial Reasoning Study Group, 2015; Resnick et al., 2020; Woolcott et al., 2020). The CCSSM standards encourage spatial orientation by calling for kindergarteners to use relational language (e.g., above, below, next to, in front of) to describe objects' positioning in the environment. A spatial orientation learning trajectory developed and validated by Clements and Sarama (2009) corroborates the appropriateness of relational language for 5- and 6-year-old children and add that children at this age can also use relational language in contexts such as referencing basic maps. Further, the spatial orientation trajectory indicates that by age 6, children should also be able to discuss distance on maps.

Kindergarten-aged children often use gesture to accompany spatial language (Davis & the Spatial Reasoning Study Group, 2015; Ferrara et al., 2011; Yang & Pan, 2021). A recent study by Yang and Pan examined how young children (ages 3–7) use spatial language while playing with blocks. Yang and Pan observed that most of the children ($N = 228$) used gesture in tandem with spatial language and that spatial language use increases with age.

In an extensive meta-analysis, Uttal et al. (2013) reported on 217 vetted research studies investigating spatial reasoning training effects. Statistical results conclude that spatial skills can be improved through training at any age regardless of gender. Spatial reasoning's malleability has promising implications, as children's spatial reasoning abilities have reliably predicted future STEM achievement (Sorby et al., 2018; Uttal et al., 2013).

Measurement Learning

Curricular activities exploring measurement in kindergarten include comparing objects side by side to describe and compare objects' *measurable attributes* (i.e., height, size, weight) using comparison language such as more/less or taller/shorter (Clements & Sarama, 2021; CCSSI, 2010). Comparing a common attribute of two objects side by side to identify which object has more/less of the attribute extends the whole-number skill of comparing quantities. For example, children might compare a set of three triangles to a set of six squares and conclude that the set of triangles is the smaller set as three is fewer than six. Similarly, students might compare a purple pencil and blue pencil that have been aligned side by side with matching endpoints (see Figure 1) and conclude that the purple pencil is longer than the blue because the purple pencil extends past the blue pencil. Preschool-aged children often recognize that alignment is important to compare objects' lengths but may not be able to explain why (Szilágyi, 2013).

Figure 1

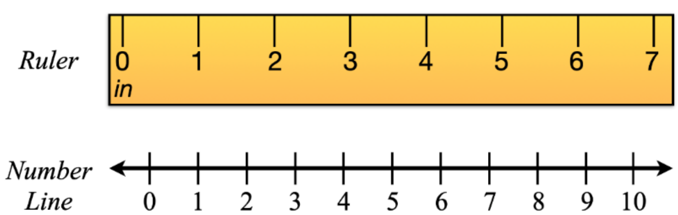
Aligned and Misaligned Length Comparison

When presented with misaligned pencils (see Figure 1), children in preschool and kindergarten might instead indicate that both pencils are long—thus attributing length as a feature of the objects rather than a comparable attribute—or state that one of the pencils are longer than the other using either vague explanations or stating that they just know (Szilágyi et al., 2013).

Students later learn to quantitize length measurement by iterating (repeating without gaps or overlaps) identical units of measure along the length of an object to be measured. Curricular standards (CCSSI, 2010) indicate that students should first measure using non-standard units of measurement (e.g., paperclips, tiles) in first grade then standard units (i.e., inches, feet) in second grade. Contrary to current curricular standards, recent research indicates that many students enter kindergarten with a basic understanding of length measurement (Kotsopoulos et al., 2017; MacDonald & Lowrie, 2011) and should be taught how to measure using standard units as early as kindergarten (Clements & Sarama, 2021; Kotsopoulos et al., 2017). Measurement requires students to consider continuous quantities (a unit that can be infinitely divided; e.g., a meter) instead of discrete quantities (distinct pieces of a whole; e.g., 5 chairs), which students typically begin working with. Working with continuous quantities in measurement is a similar skill to working on a number line, as rulers mark whole units equidistant from each other the same as number lines do (see Figure 2). Further, rulers and number lines represent continuous units, which allow students to consider operations such as addition and subtraction as distances from 0, which is typically included on rulers.

Figure 2

Ruler and Number Lines as Continuous Quantities



Alternatively, some advocate for a curriculum based in continuous quantities rather than beginning with discrete quantities (Davydov, 1975; Doughtry, 2008), which would forefront measurement and introduce students to measurement comparison before building toward discrete quantities. This reverse-approach to current practice positions mathematical structures (via qualitative measurement comparisons) above discrete numeracy as foundational to MK (Venenciano & Dougherty, 2014). Further, this approach coincides with number line instruction (Baroody & Purpura, 2017), which is a frequent tool used in U.S. mathematics instruction (e.g., Gunderson et al., 2012; Obersteiner et al., 2013; Simms et al., 2016).

Assessing Mathematical Knowledge in Kindergarten Classrooms

Mathematics assessments—whether developed by teachers for formative use or researchers for mathematics education research—provide insights into how the critical areas described above (number, spatial reasoning/geometry, measurement) are operationalized. Mathematics assessments informed this study in terms of the mathematics knowledge might be observed during the data analysis phase. This section will provide a brief overview of validated mathematics assessment for kindergarteners. Assessment *validity* is "the degree to which evidence and theory support the interpretations of test scores for proposed uses of tests" (American Educational Research Association, American Psychological Association & National Council on Measurement in Education [AERA, APA, NCME], 2014, p. 11). An assessment is valid if it has undergone rigorous evaluation to prove that it meets the above criteria. The types of mathematics assessments discussed in this section include assessments for classroom use and mathematics education research. I will provide examples of validated classroom and research assessments, then briefly describe the characteristics of these assessments and the mathematical concepts that these assessments measure. This will provide a context for current mathematics assessment availability and practices in kindergarten.

Mathematics Assessments for Classroom Use

Assessments can be used to inform teachers and education specialists of children's conceptual knowledge for the purpose of mathematics instruction and interventions. Some of these validated assessments include the Tools for Early Assessment in Mathematics (TEAM; Clements et al., 2011), the Curriculum-Based Measurement for Math (CBM; Fuchs et al., 2008), the Number Sense Screener (NSS; Jordan et al., 2010), and Individual Growth and Development Indicators (myIGDIs-EN; Hojnoski et al., 2009). The CBM and myIGDIs-EN can be used regularly in classrooms (e.g., once a week, every other week, quarterly) to track children's number knowledge to inform instruction and intervention design. The TEAM and NSS can be used as diagnostic or formative assessments. As is typical of early childhood assessments, all these assessments are interview-based, meaning that each assessment is conducted one-on-one between an instructor and a student. All four assessments use images for question items, however only the TEAM uses manipulatives (small movable objects; e.g., plastic discs, blocks), for students to respond with. These assessments primarily measure children's performance in comparing quantities, counting, and identifying numbers. Only the TEAM assesses spatial reasoning and geometric measurement.

Mathematics Assessments for Research Use

Whereas classroom assessments take relatively little time to administer (less than

20 minutes) and can be administered by any adult, mathematics assessments for research often take longer to give (20 minutes or more) and often have administration requirements. For example, the Research-Based Early Maths Assessment (REMA; Clements et al., 2019) requires potential administrators to complete and pass administration training. The Woodcock-Johnson IV Tests of Early Cognitive and Academic Development (ECAD; Schrank et al., 2014) also encourages formal training but additionally requires each administrator to hold a bachelors or advanced degree in an education-related field (e.g., speech therapy, education, counseling).

In addition to the REMA and ECAD, other validated assessments for kindergarten mathematics education research include the Child Math Assessment (CMA; Klein & Starkey, 2006) and Test of Early Mathematics Ability–Third Edition (TEMA-3; Ginsburg & Baroody, 2003; Hoffman & Grialou, 2005). Similar to the classroom assessments, these assessments are structured as one-on-one interviews between the administrator and student. The CMA only uses pictures in administration, whereas the REMA, ECAD, and TEMA-3 use both pictures and manipulatives for assessment. All four assessments heavily emphasize number skills such as counting, comparing, sequencing, and composing/decomposing number. The REMA and CMA assessments address geometric measurement. REMA has one item that directly assesses spatial reasoning.

Themes Across Kindergarten Mathematics Assessments for Classroom and Research Use

All the assessments adopt an interview-based approach, likely to account for a pre-literate population. Additionally, all the assessments use images periodically to support assessment items and half of these assessments use manipulatives (TEAM, REMA, TEMA-3, CMA). A common theme across classroom and research assessments is the focus on number concepts (see Table 1). This focus mirrors the CCSSM’s emphasis of number, but does not fully address spatial reasoning and shape, which are also critical areas in kindergarten mathematics (CCSSI, 2010). While spatial reasoning is touched on in four of the six assessments, there are very few spatial reasoning assessment items, and these items assess spatial reasoning indirectly. For example, the CMA and REMA assess length concepts and patterning, however these are limited applications of spatial concepts and only address spatial orientation and neglect spatial visualizing, which is only measured by the ECAD. One research group is currently working to fill this gap by developing a spatial reasoning assessment (Sparks et al., 2021).

Table 1 summarizes how eight MK kindergarten assessments operationalize number, shape, and space. The MK listed in Table 1 provides a summary of what students are expected to learn and do in kindergarten and includes skills that were expected to emerge in this study’s analysis.

The table shows that these assessments measure students’ number understanding using quantity comparison (e.g., showing two sets of objects and asking which is the larger/smaller set), number comparison (e.g., displaying two numbers and asking which is larger/smaller), and counting (e.g., asking a child to count a set of objects or images). Shape is assessed through shape identification (e.g., asking a child to name a given shape) and shape composition (e.g., asking which two shapes when combined will make another shape). Space is measured by these assessments using length (e.g., identifying the

Table 1

Skills Assessed by Kindergarten Mathematics Assessments

	Skills assessed		
Assessment	Number	Shape	Space
Classroom			
Curriculum-Based Measurement for Math (CBM)	Counting Missing number Number identification Quantity Comparison		
Individual Growth and Development Indicators (myIGDIS)	1:1 correspondence Counting Oral counting Number naming Quantity comparison		
Number Sense Screener (NSS)	Counting Number combinations Number comparison Number recognition Nonverbal calculation Story problems		
Tools for Early Assessment in Mathematics (TEAM)	1:1 correspondence Base 10 Counting Form of a number Number connections to the real world		Proportional reasoning
Research			
Child Math Assessment (CMA)	*Basic operations Constructing equivalent sets Counting Number order Ordinal number Quantity comparison	Shape identification	Length measurement Pattern duplication Pattern extension Shape transformation
Research-Based Early Maths Assessment (REMA)	*Basic operations Counting Number composition Number decomposition Number recognition Quantity comparison Subitizing	Shape composition Shape identification	Patterning Length comparison Length measurement

(*table continues*)

	Skills assessed		
Assessment	Number	Shape	Space
Test of Early Mathematics Ability—Third Edition (TEMA-3)	*Basic operations Number comparison		
Woodcock-Johnson IV Tests of Early Cognitive and Academic Development (ECAD)	Counting Magnitude representation Number line estimation Number recognition Number sequencing		Mental rotation Shape size comparison Visual patterning

Note. *Basic operations indicate addition with sums of 20 or less and subtraction from numbers 10 or less.

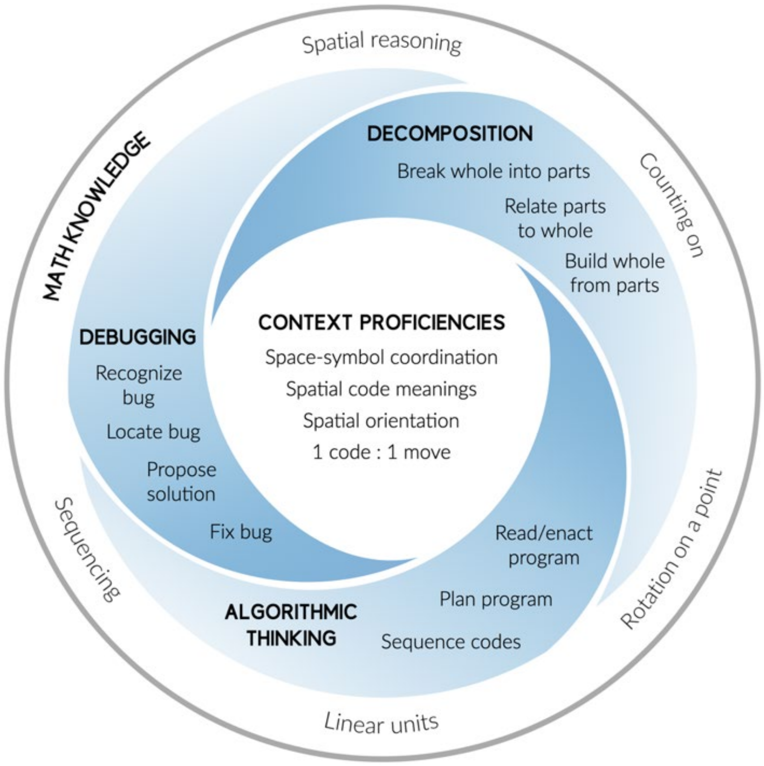
longer/shorter length, measuring with standard or nonstandard units), patterning (e.g., extending or replicating a given pattern), and proportional reasoning (e.g., comparing shapes and objects' sizes). The ECAD assessment is the only assessment to measure spatial visualization using mental rotation.

Computational Thinking in Kindergarten and Connections to Mathematics

Computational thinking definitions vary widely. As mentioned previously, in the present study, I use Shute et al.'s (2017) broad definition of CT as "the conceptual foundation required to solve problems effectively and efficiently (i.e., algorithmically, with or without the assistance of computers) with solutions that are reusable in different contexts" (p. 151). However, in terms of a specific CT model specific to kindergarten-aged children, I will use the Early Childhood Computational Thinking (ECCT) competency model (see Figure 3; Clarke-Midura, Shumway et al., 2021; Clarke-Midura, Silvis et al., 2021; Shumway et al., 2021). This is model being developed by a larger

Figure 3

Early Childhood Computational Thinking (ECCT) Competency Model



research project (Coding in Kindergarten [CiK]; Grant no. NSF #1842116) and summarizes their operationalization of CT within a kindergarten CT coding context.

The ECCT (Figure 3) describes the MK (Spatial reasoning, Sequencing, Linear units, Rotation on a point, Counting on) and CT (Debugging, Algorithmic Thinking, Decomposition) that kindergarten students demonstrate when engaging with CT tasks using screen-free robot coding toys. The ECCT was developed within a coding robot toy context, which resulted in specific Context Proficiencies (Space-symbol coordination, Spatial code meanings, Spatial orientation, correspondence of 1 code : 1 move) that are important to students' success in engaging with the robot coding toys. This model's circular design highlights the interrelated relationship between—and entanglement of—the MK, CT, and Context Proficiencies.

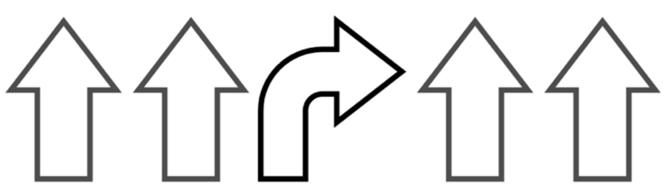
The CT constructs within the ECCT model—algorithmic thinking, decomposition, debugging, and spatial reasoning—are summarized in the following sections along with a summary of research on each construct and their relation to MK.

Algorithmic Thinking

Algorithmic thinking is a problem-solving process of developing a sequence of logical steps (Città et al., 2019). This requires an individual to flexibly adapt problem solving techniques to a specific problem (Stephens, 2018), which may involve applying previously developed algorithms. For example, in mathematics, an algorithm to find the area of a square might be (1) measure the length of one side of the square and (2) multiply the length of one side by 2. This algorithm could be represented by the formula, *side x 2 = area*. Similarly, in CT coding contexts, an algorithm for instructing a robot coding toy to travel around a blocked item might be represented as a sequence of codes (see Figure 4). The sequenced codes shown in Figure 4 would instruct a robot to (1) travel forward two spaces, (2) rotate 90 degrees right, (3) travel forward two spaces.

Figure 4

Coded Directions



Children as young as 4-years-old have displayed CT algorithmic thinking skills (Lavigne et al., 2020), and evidence suggests that algorithmic thinking skills may be developmentally related (Saxena et al., 2020). Saxena et al. developed unplugged CT activities for preschool children aged 3–6 to build children's CT skills in sequencing and pattern recognition, which are often associated with algorithmic thinking skills. One pattern recognition activity asked students to use differently sized and colored building blocks to continue a set pattern. An activity designed to build sequencing skills asked students to use a set of pictures depicting story events to order the events by how they occurred in the story. The older students performed better in patterning and sequencing skills—which are regularly categorized as algorithmic thinking skill—than their younger peers. Additionally, Boticki et al. (2018) described a correlation between children with stronger mathematics skills and the children's ability to solve algorithmic CT tasks. In this study, Boticki et al. examined 23 first-grader's performance on CT programming tasks and found that not only did students with better mathematics skills perform better on CT tasks, but the students also complete the tasks more quickly and with fewer errors along the way.

As a problem-solving process, algorithmic thinking relies heavily on sequencing skills (Lavigne et al., 2020; Nam et al., 2019; Saxena et al., 2020), an extensively researched connection within MK and CT algorithmic thinking contexts (i.e., Angeli & Valanides, 2020; Città et al., 2019; Lavigne et al., 2020; Nam et al., 2019; Saxena et al., 2020; Sullivan et al., 2017). CT coding tasks have been shown to improve children's sequencing abilities (Kazakoff et al., 2013; Nam et al., 2019; Strawhacker et al., 2013) with evidence of transfer to contexts outside of coding activities, such as sequencing events in a story (Sullivan et al., 2017). For example, Nam et al. conducted a quasi-experimental design and found that kindergarteners who participated in a 12-lesson CT coding robotics curriculum ($N = 25$) outperformed kindergarteners in a comparison group ($N = 28$) in sequencing and problem solving. The treatment group performance showed a large effect size for sequencing ($p < 0.01$; $\eta^2 = 0.15$) and problem solving ($p < 0.01$; $\eta^2 = 0.171$), indicating that the CT coding curriculum effectively and positively influenced students' sequencing and problem-solving abilities. Another research group exploring kindergarteners' performance in CT coding tasks (Strawhacker et al., 2013), found that children using a tangible, screen-free robot toy ($N = 15$) outperformed students in sequencing performance compared to students who used either a strictly digital interface ($N = 12$) or a hybrid digital and tangible interface ($N = 7$). The findings of both studies (Nam et al., 2019; Strawhacker et al., 2013) highlight the positive effects that sequencing, which is both an aspect of algorithmic thinking and MK, have on children's CT performance.

Another important aspect of algorithmic thinking that connects MK and CT is abstraction (e.g., Cetin & Dubinsky, 2017; Rich et al., 2019; Rijke, 2018). *Abstraction* is, in essence, a "process of reducing complexity by ignoring irrelevant details in order to focus attention on important elements of a problem, situation, or phenomenon" (Rich et al., 2019, p. 269). While abstraction has been specifically linked with mathematics-based sequencing and patterning (Pasnak et al., 2015; Sung et al., 2017), sequencing and patterning skills are equally valued in CT contexts as a part of algorithmic thinking. An example of the most basic, yet foundational MK abstraction, is representing quantities with words and digits (Geary et al., 2018). A picture of six apples could be abstracted with the statement “6 apples.” A child’s ability to perform this abstraction requires the student to not only count the number of apples, but to abstract the quantity of apples with the word *six* and the digit *6*.

An example of abstraction in a CT coding task is representing a sequence of steps with syntax aligned to the mechanism being enacted upon. For example, a coding toy may abstract (or represent) physical actions (i.e., movements, turns) with arrows, each arrow representing an action. In this case, arrows are an abstracted representation of an action. This is represented in Figure 4, which shows a sequence of arrows to represent movements by an agent. The example algorithm (or set of instructions) pictured in Figure 4 is abstracted with arrows, each arrow representing a movement. As illustrated in this section, abstraction, as well as problem solving and sequencing, have strong similarities between CT’s algorithmic thinking and MK.

Decomposition

Decomposition takes form in both mathematics and CT as a method to simplify tasks by breaking the tasks into manageable portions and work on one portion at a time (Wang et al., 2021). Wing (2006) listed decomposition as one of five CT cognitive processes, and it remains an essential aspect of current CT practices and frameworks (e.g., Shute et al., 2017; Wang et al., 2021).

Recent research has used CT as a mechanism to support and build upon natural experiences of embodiment by exploring repeating units (represented by program codes) in CT coding tasks. Young children have decomposed CT programming tasks into smaller subtasks (i.e., Angeli & Valanides, 2020; Lavigne et al., 2020; Rijke et al., 2018), demonstrating that children can use decomposition as a problem-solving tool for complex tasks. Students have also used decomposition to self-scaffold in CT coding tasks by coding one movement or action (a unit) at a time (Angeli & Valanides, 2020; Sung et al., 2017) rather than multiple codes at a time. Angeli and Valanides studied 50 kindergarten students' interactions with CT coding tasks and observed students frequently programming a robot one movement at a time. Angeli and Valanides contend that coding one movement at a time provides evidence of students' abilities to decompose a complex task into smaller pieces to achieve the larger objective.

While decomposing to the individual unit is a less efficient strategy, this decomposition method is a logical starting point when constructing more efficient decomposition strategies, such as chunks of similar steps. For example, if a CT coding task requires a child to program a coding toy to travel in the shape of a square, the child might recognize the pattern of linear and rotational movements so that, rather than coding each forward and rotate movement at a time, the child can consider a forward and rotation as one chunk, then repeatedly code that chunk as many times as is necessary to accomplish the task.

Debugging

An important CT skill in coding contexts is identifying and fixing developed or existing errors (Città et al., 2019). CT's term for this process is *debugging*. Debugging can be accomplished by working forward or backward from specific points in a problem to distinguish the correct from incorrect to identify and fix the "bug" or error (Lavigne et al., 2020). The Computer Science Teachers Association (CSTA; 2017) developed computer science standards for the U.S. The CSTA highlight debugging as part of their sixth core practice, testing and refining computational artifacts. This practice is written to span K–12, with note that younger students may first exercise debugging practices using trial and error.

Research engaging early childhood populations in CT suggest that children can improve their abilities to debug. For example, García-Valcárcel-Muñoz-Repiso & Caballero-González (2019) conducted a quasi-experimental study with 3- to 6-year-old students (N = 131) in which the treatment group received six CT coding robotics lessons. While the control group made limited gains in every other area, this group did not make gains in debugging. The treatment group, however, made statistically significant gains in every area, including debugging. This indicates that the students improved their debugging capabilities after engaging in coding robotics lessons.

Additionally, an instructor can provide debugging scaffolds to increase young children's debugging efficiency and success (Silvis et al., 2021). Silvis et al. analyzed 48 kindergarteners at three different schools engaging in CT coding lessons using robot toys. This group observed 472 bugs presented across the 30 hours of video. They concluded that tangible, screen-free coding toys lend themselves toward opportunities to debug not only as authentically arising programming errors as student sequence code, but also bugs due, in part, to the mechanical nature of the robot toys. Silvis et al. illustrated how teachers scaffolded programming debugging situations with guiding questions and scaffolded mechanically influenced bugs by guiding students on how to use and maintain the coding toys correctly.

Spatial Reasoning

While spatial reasoning is rooted in early Logo work (e.g., Clements & Sarama, 1997; Papert, 1980), spatial reasoning was not included as a construct in the first CT definition (Wing, 2006). Recent empirical studies link CT coding tasks with improved spatial skills, including children's improved use of directional language and ability to abstract directional movement with corresponding symbols (Palmér, 2017; Rijke et al., 2018). Research evidence also indicates that spatial skills, such as using relational language, can be difficult for children as young as ages 3 and 4, however these skills noticeably improve by ages 5 and 6 (Saxena et al., 2020). Another study observed age-specific errors, suggesting that levels of cognitive development may be a factor in children's spatial reasoning abilities (Strawhacker & Bers, 2019). This evidence suggests that spatial reasoning is, in part, developmental and correlates with research suggesting that mental rotation (a spatial visualization skill) is also developmental factor in young children (Città et al., 2019). Given this evidence that spatial reasoning is developmentally influenced, it follows that it would impact children's performance on similar tasks.

For example, researchers have used maps and grids in joint MK CT research (Moore et al., 2020; Palmér, 2017; Rijki et al., 2018; Saxena et al., 2020). Second graders and preschoolers have transferred paths from a small paper map to a larger representation of the map on which a coding toy would travel (Moore et al., 2020; Rijki et al., 2018). The preschoolers struggled with the translating across maps compared to the second-grade students (Moore et al., 2020), however the preschoolers improved their success rates by adapting the task. Children adapted tasks by reorienting themselves, using gestures, and physically moving objects to help themselves translate between a smaller and larger map (Moore et al., 2020). This suggests that while spatially-situated map tasks can be difficult for younger children, these children can improve their performance by scaffolding tasks independent of teacher intervention.

Saxena et al. (2020) explored young children's activities with coding toy robots on grids and found that while students aged 4- to 6-years-old struggled with precise spatial language (i.e., *turn right*, *forward*, *turn left*) when using the robot, the students could represent spatial thinking using pictures of directional arrows, which helped the students sequence the arrow codes and solve complex tasks. Further, by manipulating directional arrow cards, students mediated their developing understanding of spatial directional language with pictorial representations.

Assessing Computational Thinking

Attempts to measure CT constructs in early childhood have taken form in rubrics (Angeli & Valanides, 2020; Dickes et al., 2020; Lavigne et al., 2020; Miller, 2019; Saxena et al., 2020) and pre- and post-assessments (Angeli & Valanides, 2020; Miller, 2019), however few of these measures have been validated using a large population (Tang et al., 2020). One CT assessment—*TechCheck*—has been extensively validated for early childhood populations (Relkin et al., 2020). *TechCheck* is a an unplugged, 15-item interview-based CT assessment designed for 5- to 9-year-old children. Each item shows students a main image with four multiple-choice options. The items are designed to measure students' understanding of algorithms, debugging, representation (abstraction), modularity (decomposition), control structures, and programmable objects. My proposed dissertation study on the CT components of algorithms, debugging, and decomposition with the inclusion of abstraction as part of algorithmic thinking, so the following section will describe how *TechCheck* operationalizes these constructs.

TechCheck operationalizes algorithms using items to measure students' ability to extend a six-item pattern, sequence four events, and identify the shortest path in a grid-based puzzle. Decomposition is measured by asking students to identify which collection of shapes could be used to recreate a simple image. An item measuring abstraction provides students with a key for what each symbol represents, then asks students to interpret what a string of symbols would represent. A debugging item asks students to problem-solve how to make a see-saw work by positioning the fulcrum and riders symmetrically.

As a part of the Coding in Kindergarten research team, I assisted the development of a CT assessment for kindergarten-aged students (Clarke-Midura, Silvis, et al., 2021). To assess algorithmic thinking, we developed assessment items in which children used paper arrows to program a moveable agent to move across a paper-based grid (Clarke-Midura, Silvis, et al., 2021). For example, the assessor asks the child to use the arrows to show how to direct the movable agent (i.e., a bug) from a start square on the grid to an indicated end square. We used an Evidence Centered Design approach to develop the CT assessment (Clarke-Midura, Silvis, et al., 2021) and found that—because of the assessment's tangible and unplugged design—spatial reasoning is a key component of the algorithmic thinking assessment items.

While there has been a recent call for domain-specific CT assessments such as CT within science or mathematics (Tang et al., 2020), these have yet to be developed. This is likely due to the field's emerging understanding of how to measure CT and how specific domains support CT. This study attempted to partially address this gap by identifying how kindergarteners' MK might relate to their CT assessment performance. In doing so, the present study informs the field about how specific MK is associated with CT tasks and constructs as well as how MK and CT constructs co-occur.

Summary: Mathematical Knowledge Computational Thinking Connections and Co-Occurring Concepts in Kindergarten

While decades of research on early childhood MK exist, research on early childhood CT is still emerging. Even more nascent is the theoretical and empirical research on connections and co-occurring concepts between MK and CT (e.g., Rich et al, 2019; Miller, 2019; Shute et al., 2017). The MK literature provided an overview of how number, shape, space, and measurement are operationalized in kindergarten classrooms. According to the MK literature, students' ability to represent number abstractly in symbol and words, as well as comparing values, is an important predictor of later achievement (Booth & Siegler, 2008; De Smedt et al., 2009; Libertus et al., 2011). In kindergarten, students experience geometry by naming, classifying, and constructing shapes while attending to shape attributes (Clements & Sarama, 2021). While geometry and spatial skills appear to be underrepresented in MK instruction (Battista et al., 2017), spatial skills are a predictor of early childhood mathematical performance (Mix et al., 2016). Kindergarten children can use spatial relational language (i.e., next to, behind) and maps to build spatial orientation and visualization abilities (Clements & Sarama, 2021). Kindergarten students engage with measurement by using comparison language (i.e., longer/shorter, heavier/lighter) to describe objects' measurable attributes (i.e., weight, distance, length). Measurement comparison and description is treated as a precursor skill to formal measurement instruction in first and second grade, wherein students learn to iterate standardized units of length measurement (CCSSI, 2010). It has been argued, however, that kindergarten students are capable and eager for formal measurement instruction (e.g., Kotsopoulos et al., 2017; Szilágyi et al., 2013). Similar to space and shape, measurement instruction takes a backseat to number instruction in U.S. kindergarten classrooms, however some argue that space, shape, and measurement could be used as a foundational element in early childhood MK instruction (e.g., Doughtery, 2008; Venenciano & Daughtery, 2014).

The CT literature describes how algorithmic thinking, decomposition, debugging, and spatial reasoning are operationalized in kindergarten and early childhood classrooms. The CT algorithmic thinking literature highlights how patterning, sequencing, abstraction, and spatial reasoning are used within a CT coding context (Clarke-Midura. Shumway, et al., 2021; Relkin et al., 2020). In this vein, algorithmic thinking and MK share similar features, which is where I anticipated to find deep MK and CT connections. For example, abstracting robot movements to signs and symbols, such as representing one forward movement with a picture of a straight arrow, is similar to kindergarteners abstracting quantities with number words and digits. Additionally, students associating a robot's linear movement in space with a corresponding symbol expands on students' development of spatial language and spatial representational skills, another early childhood skill.

Decomposition in CT coding tasks is likewise connected with MK. For example, kindergarten students sequencing programming code might code individual units of movement at a time to achieve an objective (Angeli & Valanides, 2020). This requires students to recognize that the individual parts, when combined, create the whole program. Understanding that a whole is made up of parts is an important component of kindergarten number sense (Hunting, 2003).

The CT literature section also highlighted how spatial skills are particularly relevant to CT coding contexts, especially within a physical space. Using spatial directional (i.e., forward, turn right) and sequencing (i.e., first, then, next) language expands on kindergarten curriculum standards which encourage students to use spatial relational language (i.e., next to, in front of) to express objects' relative location (K.G.A.1; CCSSI, 2010)

The CT literature suggests that algorithmic thinking, decomposition, and spatial skills may have the most synergistic connections with MK, and these domains are where I hypothesized that algorithmic thinking, decomposition, and spatial skills play especially significant roles in early childhood MK CT research.

Theoretical Perspectives in Mathematical Knowledge Computational Thinking Research

Commonly utilized theories exploring young children's MK CT connections include embodied cognition, Bruner's (1961) modes of representation theory, and sociocultural theory. Following with previous MK CT research adopting embodied perspectives (Città et al., 2019; Moore et al., 2020; Sung et al., 2017) and enactivism (Francis et al., 2016), this study applies a joint embodied cognition and enactivism theoretical perspective. Embodied cognition and enactivism are described below with rationales of how these theories informed the present study.

Embodied Cognition

In describing human cognition, Campbell (2010) states that "every subjective sensation, memory, thought, and emotion—anything at all that any human being can ever experience—is in principle enacted in some objective, observable way as embodied behavior" (p. 313). Embodied cognition theory views a person's biological body and brain (cognition) as separate yet interconnected within an environmental context (Davis & Francis, 2020). By viewing cognition as an independent entity of the body, one is thereby viewing knowledge in isolation and seeing knowledge as something that can then be applied through gesture and environment. Of particular relevance is an embodied cognition perspective that views cognition as grounded in bodily systems and situated activity (Núñez et al., 1999). An embodied cognition perspective considers cognition as grounded in bodily systems and situated activity.

Three threads of embodied cognition theory are applicable to in mathematics education research (Reid, 2014). These threads include neuroscientific approaches (applying tools such as brain scans and eye-tracking technology; e.g., Kiefer et al., 2017; Kucian et al., 2011; Sprenger & Bez, 2020), mathematical metaphors (understanding concepts through embodied metaphors; e.g., Arnoux & Soto-Andrade, 2019; Gallagher & Lindren, 2015), and gesture (bodily movements; e.g., Congdon et al., 2018; Cooperrider et al., 2016; Walkington et al., 2019). The gesture thread is most pertinent to the present study. This thread of research uses embodied cognition theory in mathematics education research and typically involves investigations about how children use gestures and how gesturing can impact learning.

In a study investigating how gesture-based instruction influences length-measurement learning compared to instruction favoring interaction with objects, Congdon et al. (2018) discovered that young children's preexisting levels of length understanding was a factor in the student's ability to learn from gesture-based instruction. Students with a more advanced understanding of length measurement outperformed students with a rudimentary understanding of length measurement in gesture-based instruction. The research team hypothesized that this is due to an increased level of abstraction required for interpreting gestures as opposed to action-on-objects based lessons.

Another study about gesture within an embodied cognition framework explored children's engagement with CT coding tasks using maps (Moore et al., 2020). This research team observed second-grade students using gesture, object manipulation, and language simultaneously to solve the tasks. As the tasks increased in difficulty, the students appeared to increasingly rely on gesture, object manipulation, and language to reduce extraneous cognitive load.

Both studies emphasize the connection between spatial reasoning and gesture (Congdon et al., 2018; Moore et al., 2020). Moore et al. suggested that gesture and movement might support students' spatial reasoning development. Abstraction is also a common theme as each study situated gesture as a high-level form of abstraction and tool to assist students' sense-making.

Similar to these studies, the present study on students' use of MK during a CT assessment necessitated an embodied cognition perspective as gesture is a spatial abstraction of knowledge and informed my interpretation of students' gesture, language, and actions on objects as MK indicators. The research questions in this study were addressed by observing children's mathematical behaviors (i.e., gestures, language, actions on objects) as the students interact with CT assessment items. Embodied cognition theory informed my interpretation of children's enactment of assessment items and permitted me to interpret the participants' actions as their MK embodied. However, because this research query investigated children's engagement in a novel context (one-on-one with administration with an assessor; unique CT assessment materials) and embodied cognition focuses on knowledge through previous actions (Khan et al., 2015), an additional theory was necessary to account for the novel context of students' interactions with assessment materials and situations that the students have not previously encountered. Enactivism was used to address the study's novel context.

Enactivism

Similar to embodied cognition theorists, enactivists view cognition as biological, nested within and inseparable from the knower's biological body (Li et al., 2010). This assertion draws, in part, from biological science which reports that learning produces physical changes in the brain (Doidge, 2007). Whereas embodied cognition is the enaction of learned knowledge, enactivism is "learning in action" (Khan et al., 2015, p. 272) or knowledge-as-action (Davis, 1996). What a person perceives is what is known, and enactivism sees knowledge as performative. Thus, knowledge is inseparable from doing (Brown & Coles, 2011). Further, enactivism considers the environment as not just an extension of cognition, but a part of it. This nondualistic theory holds that cognition, the body, and one's environment are inseparable.

Enactivism positions the knower as inseparable from the known; the *known* being the makeup of knowledge an individual can enact within an environment (Maheux & Proulx, 2015). *Enaction* itself, however, is a transformative process of embodied understanding (Davis & Francis, 2020), so the very act of enacting knowledge will further transform the knower and the known. Enaction is likewise transformative to the environment as the environment is inseparable from the knower (Li et al., 2010). Enaction is observable through a person's "facial expression, posture, movement, [and] gestures" within an environment (Gallagher, 2017, p. 42). As such, an observer watching a child enact an assessment item is not seeing a product of the child's knowledge, but the child actively knowing in their actions (Maheux & Proulx, 2015).

Enactivism considers learning as an active exercise of experiences and exploration rather than knowledge acquisition (Li et al., 2010). The enactivist interpretation of learning as action highlights enactivism's positioning of the *knower* as a dynamic vessel of cognition, body, and environment rather than centralizing knowledge in cognition. For example, an enactivist-minded instructor teaching a pair of kindergarteners how to code a robot to navigate to a destination three spaces forward might organize a play-based opportunity wherein the students attempt the task through exploration with a robot and its coding mechanisms. The teacher would expect the students to learn that one forward code correlates with the robot moving forward one space (a one-to-one code/movement relationship) by engaging with the robot, its coding mechanisms, and discussing observations with each other (cognition/body/environment). An enactivist observer would interpret an individual's actions as a transformative coordination of the individual's cognition and body within an environment, so the enactivist-minded teacher would monitor the students' discussions and enactions to interpret each student's knowledge of a forward code's relationship with the robot's movement. The teacher would also understand that individuals are dynamic, so knowledge observed in the moment will change with additional experiences. In response to this, the teacher would engage the class in similar activities to further develop the children's knowledge of equality through engaging learning experiences.

In an assessment setting, such as this study's context, an enactivist observer would consider all assessments as formative in that, according to enactivist theory, knowledge is dynamic and never stagnant. Further, an enactivist would interpret a student's action within an assessment as indicative of the student's evolving knowledge being expressed.

For example, when asked to select a directional arrow to make a movable agent rotate right a student might look at the agent before selecting an arrow. An enactivist might interpret the student's gaze before selecting an arrow as an indication of the student learning how to interpret the assessment instruction within the assessment space.

Enactivism is rooted in Merleau-Ponty's (1964) phenomenological work and Bateson's (1972) biological perspectives. Enactivism was formally conceived by Varela et al. (1991) in *The Embodied Mind*, in which the authors contend, in part, that our consciousness is a developing relationship with our environment so that our bodies are fluid products of our joint biology and lived experiences. Later work by Davis (1996) applied enactivism to mathematics education. Using an enactivism perspective, Davis urges a reduction in dichotomic thinking so far as it is pragmatic to do so. For example, he argues that separating the mind/body, knower/known, thinking/doing, or self/other is not always in the learners' best interests, and urges readers to acknowledge the connections among these seemingly unique constructs (i.e., knower/known). An example of enactivism framing MK research related to this review is Città et al.'s (2019) work exploring mental rotation within a CT coding context. This study examined 6- to 10-year-old students (N = 92) as they engaged in CT coding curriculum. Città et al.'s curriculum reflected an enactivist approach using interactive activities and social experiences that sought to actively engage the mind-body-environment connection highlighted in enactivism. This study, however, did not frame the results within an enactivism perspective, which could have reinforced the study's theoretical positioning.

Francis et al. (2016) applied enactivism to a explore spatial reasoning within a CT coding context. Their study examined 9- and 10-year-old students (*N* = 18) as they programmed robots. Conversely with the Città et al. (2019) study, Francis et al. used enactivism to interpret how students employed spatial reasoning to engage with and learn coding concepts. For example, the researchers described how, while programming on a computer, a student curls her fingers, swings her legs in, and leans forward while adding a code to a programming sequence. The researchers gave additional descriptions of how student positioning, gestures, and movement coincided with programming activities. In this way, the authors effectively used enactivism to describe how students engaged their mind-body-environment connection to accomplish CT coding tasks using spatial reasoning.

While the present study's context is situated in CT assessment, and enactivism centralizes learning as doing, I used an enactivist lens to account for novel context of an assessment, for which embodied cognition does not provide. Because the CT assessment uses tools and activities different from what students had previously engaged with in curriculum activities, the assessment context required an enactivist lens to account for the novel nature of the assessment. Thus, embodied cognition informed my interpretation of students' gesture, language, and interaction with the CT assessment as evidence of existing MK while enactivism accounts for the unique context of an assessment wherein the student is actively learning how to engage in novel tasks. Embodied cognition provides a useful lens for understanding the kindergarteners' knowledge as performed, while enactivism views knowledge as performative. This dual-lens is particularly important in kindergarten, as much of students' knowledge is in transition, hence using two theoretical lenses accounted for both knowledge as performed and knowledge-in-action. These perspectives informed my interpretation of students' performance on CT assessment items.

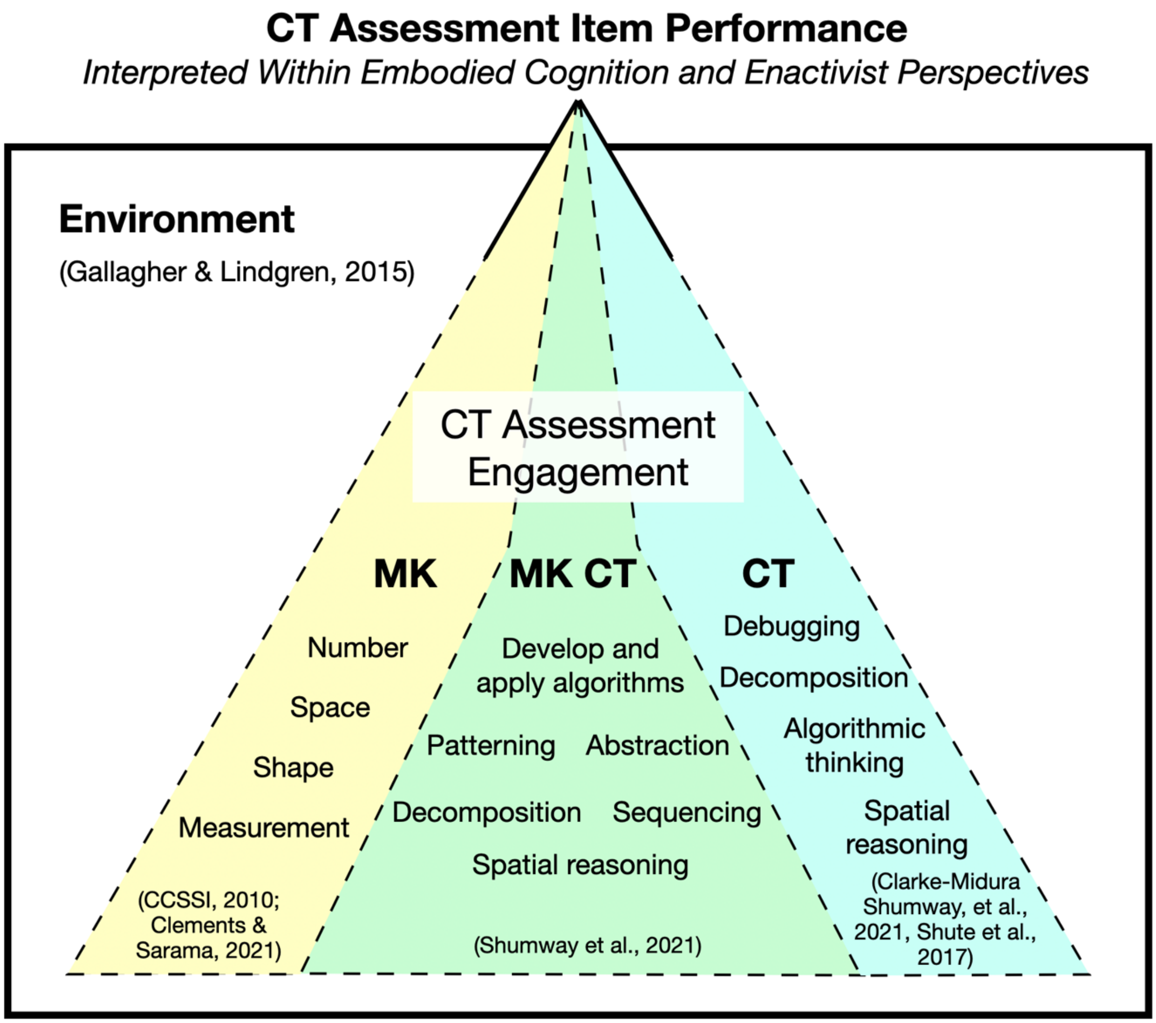
Conceptual Framework

The CT Assessment Item Performance Conceptual Framework (CT-AP; see Figure 5) is based in evidence connecting MK and CT (i.e., Clarke-Midura, Shumway, et al., 2021; Miller, 2019; Rich et al., 2019; Shumway et al., 2021; Weintrop et al., 2016) and depicts a child's application of MK, and CT skills to solve CT assessment items. This framework is specific to kindergarteners' assessment item performance within a CT coding context. The literature review situated this study in current literature by synthesizing recent research within MK and CT early childhood research and related assessment methods. The literature review also drew connections between MK and CT to inform this study as to how MK and CT might co-occur within a CT assessment context. The CT-AP depicts the MK, CT, and co-occurring MK CT knowledge that a kindergartener was expected to indicate as the student interacted with the CT assessment items.

The framework constructs (shown in Figure 5) include MK, CT, the expected MK CT co-occurrences, and the environment. Each construct is informed by embodied cognition, enactivism, and early childhood research in MK and CT. The triangle's wide base represents the entirety of a student's MK and CT knowledge. The narrowing at the top represents the knowledge that the child finds relevant and demonstrates while

Figure 5

Computational Thinking Assessment Item Performance Conceptual Framework (CT-AP)



interacting with a CT assessment item. The yellow shaded area (left section of the triangle) represents a child's possible MK (CCSSI, 2010; Clements & Sarama, 2021) and the blue shaded area (right section of the triangle) represents the anticipated CT skills that students would indicate (Clarke-Midura, Shumway, et al., 2021; Shute et al., 2017). The present study was situated in the framework's yellow (MK) and green (MK CT) sections, in considering the connections and relationships between MK and MK CT that students use when solving CT assessment items (building from Shumway et al., 2021). Using a dual embodied and enactivist perspective I examined the MK and CT that students indicated during an assessment and how students' MK and MK CT co-occurrences might relate to students' CT assessment item performance. In Chapter III, I outline the methods used to conduct this study.

CHAPTER III

METHODS

The purpose of this study was to operationalize kindergarten students' use of MK and CT during a CT assessment and understand how kindergarten students' MK and MK CT co-occurrences might relate to their CT assessment performance through the theoretical lenses of Embodied Cognition and Enactivism. This study employed qualitative methods across three sequential phases:

Phase 1. Identified the mathematical knowledge (MK) and computational thinking (CT) that kindergarten students demonstrated during a CT assessment;

Phase 2. Explored the co-occurrences of MK and CT; and

Phase 3. Developed case studies of assessment items and conducted a subsequent cross-case analysis to examine in what ways students' MK and MK CT co-occurrences related to their CT assessment item performance.

I used existing video data of kindergarten students engaging with CT assessment items and the students' subsequent assessment results during all three phases of the study.

Research Context for Existing Data: Coding in Kindergarten

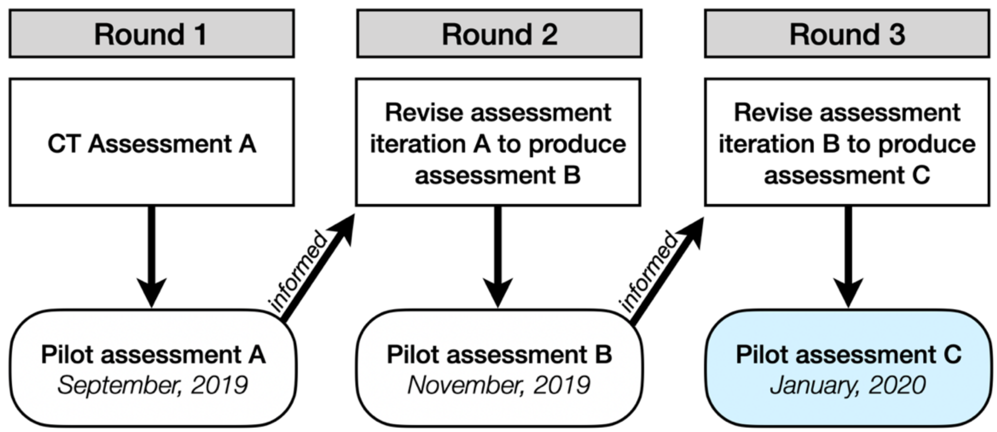
This study drew on existing data from a larger study called Coding in Kindergarten (CiK; National Science Foundation Grant #1842116). The CiK research team developed and piloted curriculum tasks and assessment items with kindergarten students at Title I elementary schools in the western U.S. to investigate early childhood computational and mathematical thinking using commercially available robot coding toys (Clarke-Midura, Shumway, et al., 2021). Coding lessons engaged small groups of students in programming tangible robot coding toys (e.g., Cubetto and Botley toys) to travel to and from various locations on large floor grids. The design-based research around these curriculum tasks served to operationalize CT in early childhood and develop an early childhood CT assessment. While the curriculum was developed to use coding toys in small-group settings in the classroom, the assessment was not connected to coding toys and was designed to assess CT more broadly using similar activities in a standardized one-on-one interview-style settings with unplugged coding materials (i.e., paper arrow codes and non-robot moveable agents). As a Graduate Research Assistant for the CiK research team, I participated in all aspects of this research: developing and testing curriculum, operationalizing CT, and designing and piloting the assessment.

One of CiK's primary objectives is to develop a CT assessment for kindergarten-aged students (Clarke-Midura, Silvis, et al., 2021), and it is this assessment that served as the context for this study. Figure 6 depicts the CT assessment development piloting process from conception until the data set used for this study. We designed the assessment using evidenced-centered design and piloted and tested using it using design-based research methods (Clarke-Midura, Silvis, et al., 2021).

I examined the video data and score sheets collected for pilot assessment C (highlighted in blue in Figure 6), the most recently piloted version available at the time the present study began. This data set was chosen as it was, at the time, the most developed version of the CT assessment. The data set included 15 hours of video data documenting 60 students and their accompanying score sheets. These sources will be described in the Data Sources section.

Figure 6

Computational Thinking Assessment Pilot Process



Justification for Use of an Existing Dataset

I selected this dataset since I had collected the data with the CiK team and, in collaboration with the CiK project's Principal Investigators, determined a reanalysis of the dataset would support the project broadly, yet be unique to the dataset's initial assessment-development purpose. While the CiK team also used this dataset for grant-related research, the research objectives and theoretical perspectives of the CiK research and the present study are distinct. According to the American Psychological Association (APA, 2019) "it is not considered duplicate publication to reanalyze already published data in light of new theories or methodologies, if the reanalysis is clearly labeled as such and provides new insights into the phenomena being studied" (p. 18). The remainder of this section will detail how the present study met the APA guidelines for reanalyzing already-published data.

This study examined students' MK and CT separately *and* together within an assessment context. Examining MK and CT within an assessment environment provided a better understanding of MK and CT in early childhood, as the interview-based assessment video data provided a rich look into students' independent interaction with CT tasks (compared with collaborative interaction during the small-group lessons with robot toys in curriculum tasks in other components of the CiK project). The assessment video data also permitted me to document how the students interacted with CT tasks, thus moving beyond the correct/incorrect assessment outcomes to witness how students indicated MK and CT.

My use of existent data is also different from the CiK project's use as I interpreted the data using a unique theoretical lens (enactivism) and generated new insights into the mathematical aspects of students' engagement in the CT assessment. Analyzing this dataset with a joint embodied cognition and enactivist lens expanded the CiK team's use of embodied interpretations while an enactivist lens provided a unique perspective of how students interacted with the assessment as a tool for which students had no previous instruction. An enactivist lens contributes to the field's theoretical understanding of how young children indicate MK and CT knowledge through gesture, language, and interaction with the environment through actions on objects.

Whereas our CiK team used the assessment video dataset for assessment-development purposes using an Evidence-Based Design approach (Clarke-Midura, Silvis, et al., 2021), I used the video data in this study to examine the content connections between MK and CT. This study builds on the CiK team's previous work with a curriculum video dataset that investigated students' application of MK and CT in a small-group collaborative problem-solving context (Shumway et al., 2021). My study extends this work by focusing on how students' MK and CT knowledge are connected when students engage with CT tasks independently. This further appropriated using the intended dataset as outlined in this study.

Research Design with Existing Data

To examine how kindergarten students indicated MK and CT and examine how their MK and MK CT co-occurrences might relate to their CT assessment item performance, I used a sequential, three-phase qualitative design in which the findings of research question 1 informed the subsequent analysis of research question 2. A qualitative design was appropriate as I conducted an inductive exploration to build a greater understanding of MK and CT connections. The three analysis phases will be discussed in more detail in the subsequent sections.

The focus of this design was to observe students' interactions with the assessment items as well as students' assessment outcomes. The CT assessment's highly interactive design permitted me to observe how individual students indicated specific MK and CT. This study's units of analysis included the student being assessed and the CT assessment items. Data analysis at the student level addressed research question 1, while analysis of data at the assessment item level addressed research question 2. The following sections describe the participants and setting, data sources, acquisition of the dataset, and data collection procedures before describing the data analysis.

Participants and Setting

The assessment video data for this study included 60 kindergarten students (33 females, 27 males) from four classes located at one Title I public school in a semi-rural community in the western U.S. The majority of students (33) were Caucasian. Other ethnicities represented include Asian/Pacific Islander (2), Hispanic/Latinx (3), Native American (1), and mixed/other ethnicities (2). Nineteen of the parents chose not to report their child's ethnicity. These demographics were similar to the school's demographics, as 71% of the school's student population identified as white and 29% identified as from a minoritized group, 77% of whom were Hispanic/Latinx.

The students attended one of four classes at this school. One teacher taught half-time to an afternoon group, another taught two half-day kindergarten sessions, and the third teacher taught a full-day kindergarten class. The students at this school participated in the three rounds of assessment, not the curriculum portion of CiK's research. The classroom teachers, however, taught CT on their own using unplugged activities such as the Turtle Robots board game and used coding robot toy activities similar to those in our curriculum research.

We worked closely with all three kindergarten teachers to coordinate three pilot testing iterations, each spaced about two months apart. We administered the assessment to the same students for each round. We did not provide performance feedback to students either during or after the assessment. A few of the assessment items were similar across assessment iterations.

Data Sources and Collection Instruments

This study's two data sources were collected concurrently and included audio/video footage of kindergarten students engaging with the CT assessment (Pilot

Assessment C in Figure 6, $N = 60$) and the assessment results from the same students ($N = 60$). Each video was about 15 minutes long and depicted a CiK researcher administering the CT assessment to a kindergarten student. Both the administrator and student sat at a table as the student manipulated assessment materials (i.e., arrow cards, a movable agent, pencil) in response to each assessment item.

Materials used for data collection included video cameras and assessment scoring sheets (see Appendix B). Additional items for data collection included the assessment materials (Clarke-Midura, Silvis, et al., 2021), tripods, backup batteries, battery chargers, memory cards, and extension cords.

The two data collection instruments for this study were a pilot CT assessment and its accompanying score sheets. The data sources (audio/video footage, assessment results) and collection instruments (CT assessment, assessment scoring sheets) were interconnected in such a way that they will be described in tandem below.

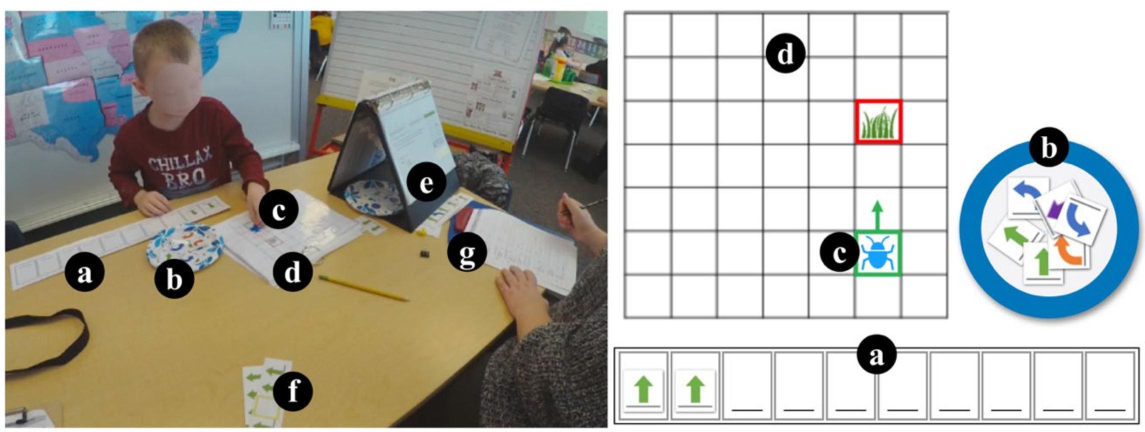
Computational Thinking Assessment

The assessment consisted of 27 items assessing the CT skills of spatial reasoning, algorithmic thinking, debugging, and decomposition (see Appendix C for Pilot Assessment C; Clarke-Midura, Silvis, et al., 2021). Our CiK research team constructed the assessment items for the purpose of examining young children's computational thinking. We made assumptions that spatial reasoning and other mathematics knowledge and practices were implicit in the assessment tasks due to the spatial nature of the environment (i.e., moving an agent within a grid space) and the need to count movements and codes. Our CiK team developed the CT assessment while simultaneously operationalizing CT, however the assessment items I examined measure the CT subcomponents described in the Data Analysis Section.

Materials used for assessment administration are pictured in Figure 7. The CT assessment materials include a program organizer for students to build programs (Figure 7, a), programing codes represented as arrows (b), a movable agent (c), paper grids specific to each task (d), an administration script (e), program sequences for specific items (f), and a scoring sheet (g).

Figure 7

Assessment Materials

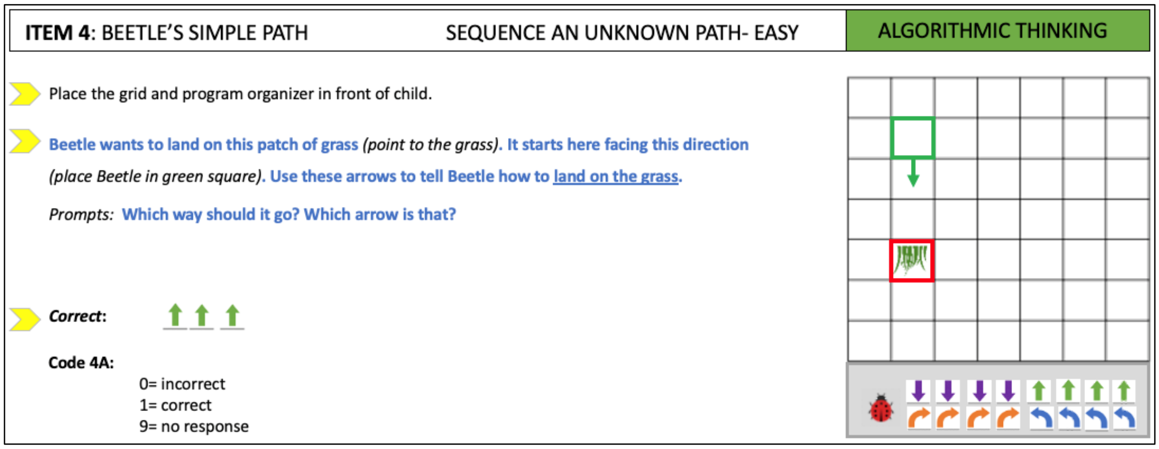


Note: A researcher administering the CT pilot assessment with the labeled assessment materials. Image adapted from Clarke-Midura, Silvis, et al., 2021, Figure 3.

Figure 7 contains a screenshot of actual assessment administration on the left and a larger, pictorial representation of specific assessment materials on the right. Figure 8 shows a sample item script used by the assessor (a description of the administration is in the Data Collection Section).

Figure 8

Sample Administration Item

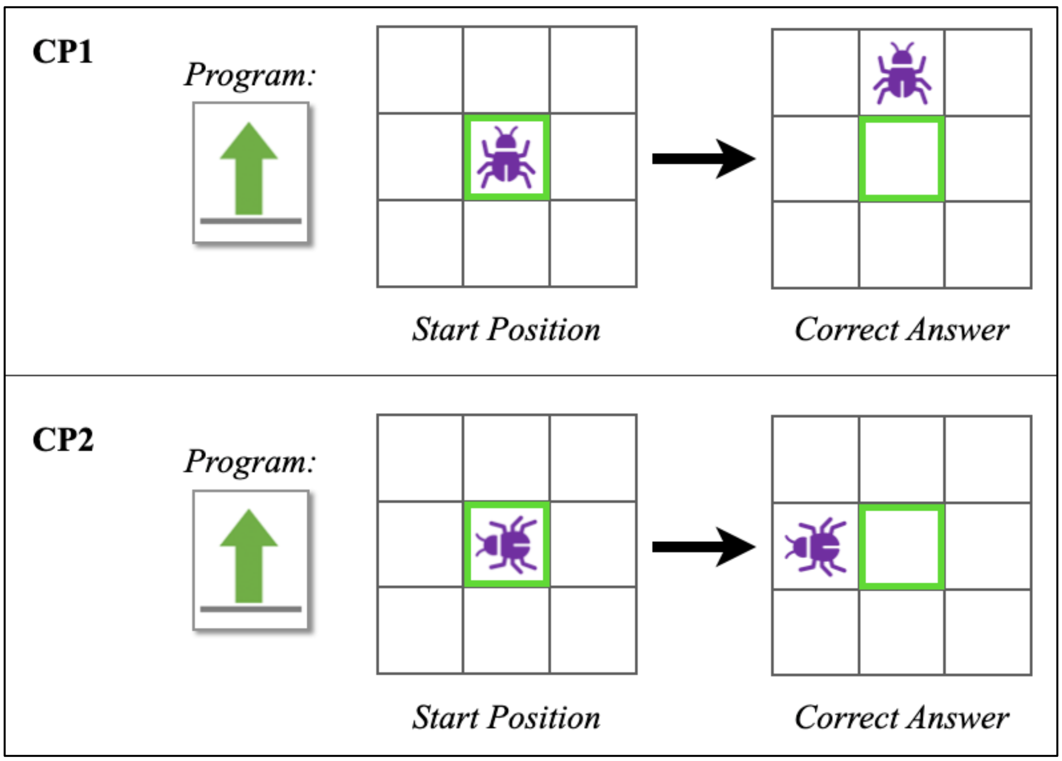


The grid pictured on the right side of Figure 8 shows the assessment item's associated grid from the administrator's perspective with an arrow indicating which direction the movable agent should be facing at the start of the task. Additional student materials required for the item are pictured below the grid. The first yellow arrow on the figure's left provides setup and administration instructions. The second yellow arrow indicates the administration script. The administrator is meant to read the blue, bolded prompt while enacting the italicized directions. The prompt provides verbiage to prompt the student to engage with the assessment item, if needed. The final yellow arrow indicates the anticipated answer and assessment coding for the score sheet.

Of the original 27-item assessment, I analyzed 14 items—four contextual proficiency (CP) items and 10 CT items. I selected only 14 items as that the remaining 13 items had been dropped or flagged for redesign by the CiK research team. The CP items were the first four assessment items and were intended to measure students' ability to manipulate an agent in the grid space in response to a given program code and the agent's starting orientation. For example, assessment item CP1 asked the student to move the agent as if the agent began facing the same direction as the student and was programmed with one forward code (see Figure 9, CP1).

Figure 9

Sample Context Proficiency Items



CP2 asked students to move the agent as if the agent were programmed with a forward code, this time when the agent started facing the child's left (Figure 9, CP2). Items CP3 and CP4 reflect CP1 and CP2's design, but with rotation codes instead of forward codes. Analyzing these CP assessment items provided a baseline for students' context proficiencies.

The 10 CT items that I analyzed were the assessment items retained in CiK's final assessment. Concentrating my analysis on these items allowed me to analyze patterns more deeply among and within these items. In all, the 14 items (4 context proficiency items + 10 CT items) constituted about eight of the 15 hours of video across 60 students.

Assessment Scoring Sheet

The scoring sheet (see Appendix B) organization reflected the assessment item progression with one row for each item. Each item row included the item number, the correct response, a section to code student responses (correct, incorrect, no response), and general notes. To score the assessment, a point was awarded for each item the student answered correctly. Each item was worth one point, so the highest score possible for the 27-item assessment was 27. Since this study focused on 14 of the 27 items, the highest possible score was 14.

Acquiring the Data Set

I requested to use this dataset from the CiK primary investigators (Drs. Jody Clarke-Midura, Jessica Shumway, Victor Lee). All primary investigators (two of whom are on my dissertation committee) approved my request. Additionally, Utah State University's Institutional Review Board approved the use of CiK's NSF-funded project data for this dissertation research (see Appendix A). After successfully defending my dissertation proposal, I submitted amendments – which were subsequently approved – to the existing CiK Institutional Review Board certification to use this dataset for my dissertation research.

Data Collection Procedures in the Larger Project

We, the CiK research team, recruited classes from the Title 1 school and then collected informed consent forms from the teachers and assent from the student participants in English and Spanish. The assessment administration for this dataset spanned six days over the course of two weeks in January 2020. All researchers administrating the assessment had prior teaching experience and advanced degrees in education or other related fields.

We set up as many as three assessment stations at a time in the students' classrooms and distanced each from distractions as much as feasibly possible. Each station included a table to organize materials, chairs positioned so that the student faced the administrator and away from other potential distractions, the assessment materials, an audio/video camera positioned to capture students' actions and expressions, and a researcher acting as assessment administrator. We collected audio/video data for future analysis and to free the assessor to focus on assessment administration rather than taking detailed field notes (Blikstad-Balas, 2017). As schedules permitted, additional researchers attended the assessment to serve as second exam coders. The students were accustomed to interventions and assessments set away from regular classroom activities, so students were rarely hesitant to leave classroom activities during the assessment. When such concerns arose, the researchers rescheduled the student's assessment for a later time.

The assessment administrators followed standardized facilitation procedures and used the items' scripted prompts with fidelity (see Appendix C). To administer the assessment, the assessor read the student an introduction script to explain that the student would answer questions about a "robot" (the agent) moving around on a grid. The administrator introduced all assessment materials and demonstrated how each program code instructed the agent to move. If the student needed assessment item instructions repeated, the administrator was permitted to repeat the instructions once. Once the student completed an item, the administrator transitioned to the next item without providing performance feedback. This was often done by the administrator saying, "Thank you."

Many students completed the assessment within 15 minutes while a few took up to 18 minutes. This time frame is consistent with other interview-based, early childhood assessments (Hojnoski et al., 2009; Jordan et al., 2010). Fifty-three of the 60 students completed the entire assessment.

We uploaded the video memory card data after each assessment day to an external hard drive and Box, an institutionally funded and secure cloud-storage platform. The video memory card was wiped immediately after we confirmed the file transfer. The hard drive was password protected and housed in a locked office. Box is an encrypted, cloud-based storage platform. Access to the CiK files on Box are limited to researchers listed on the CiK institutional review board protocol. We named each video with a standardized file-naming format to anonymize the student and for future reference:

> (1-digit location #)(1-digit teacher #)(2-digit student #)_(assessor initials)_(four-digit year)(2-digit month)(2-digit day)_(Uploader's initials)_(assessment version)
>
> *Example:*
>
> 1101_AI_20200101_UI_Version3

Data Analysis

Table 2 summarizes the present study's research questions, data sources, and analysis methods. I will first explain how I cleaned and organized the data and then describe the three sequential phases of data analysis: (1) identification of MK and CT, (2) analysis of ways MK and CT co-occur, and (3) case study and cross-case analysis.

Table 2

Research Questions, Data Sources, and Data Analysis

Questions	Data sources	Analyses
1. How are kindergarten students' MK and CT operationalized during a CT assessment? In what ways, if any, do MK and CT co-occur?	Video of CT assessment administration Co-occurrence frequency report Qualitative memos	A priori coding and open coding (Saldaña, 2021) Memos (Saldaña, 2021) Frequency tables (Christopher, 2017)
2. How do students' MK and MK CT co-occurrences relate to their performance on individual CT assessment items?	Video of CT assessment administration CT assessment item scores	Visualizations (Christopher, 2017) Multiple-case studies (Yin, 2018)

Preparing and Cleaning the Data

I prepared the video data for analysis in MAXQDA 2020, a video analysis software, by isolating each assessment item's occurrence within the video data and indicated if the student answered the item correctly or incorrectly (see Table 3).

Table 3 depicts each assessment item code with its corresponding meaning. These codes served two organizational purposes. The first purpose was to isolate the occurrence of students' engagement with the assessment items to indicate where in the video each

Table 3

Assessment Item Codes

Item code	Code meaning
Contextual Proficiency (CP) Items	
CP1	Contextual Proficiency Question 1 – answered correctly
-CP1	Contextual Proficiency Question 1 – answered incorrectly
CP2	Contextual Proficiency Question 2 – answered correctly
-CP2	Contextual Proficiency Question 2 – answered incorrectly
CP3	Contextual Proficiency Question 3 – answered correctly
-CP3	Contextual Proficiency Question 3 – answered incorrectly
CP4	Contextual Proficiency Question 4 – answered correctly
-CP4	Contextual Proficiency Question 4 – answered incorrectly
Computational Thinking (CT) Items	
CT1	CT Question 1 – answer correctly
-CT1	CT Question 1 – answer incorrectly
CT2	CT Question 2 – answer correctly
-CT2	CT Question 2 – answer incorrectly
CT3	CT Question 3 – answer correctly
-CT3	CT Question 3 – answer incorrectly
CT4	CT Question 4 – answer correctly
-CT4	CT Question 4 – answer incorrectly
CT5	CT Question 5 – answer correctly
-CT5	CT Question 5 – answer incorrectly
CT6	CT Question 6 – answer correctly
-CT6	CT Question 6 – answer incorrectly
CT7	CT Question 7 – answer correctly
-CT7	CT Question 7 – answer incorrectly
CT8	CT Question 8 – answer correctly
-CT8	CT Question 8 – answer incorrectly
CT9	CT Question 9 – answer correctly
-CT9	CT Question 9 – answer incorrectly
CT10	CT Question 10 – answer correctly
-CT10	CT Question 10 – answer incorrectly

item began and ended. Each video had 14 occurrences or—as was the case for seven of the 60 students—as many items as the students completed. The second purpose was to label each engagement as the student answering correctly or incorrectly. Organizing the data to include the students' correct/incorrect responses supported the Phase 3 analysis, when I examined how students' MK and MK CT co-occurrences might relate to their performance on individual CT assessment items. This will be discussed further in the Phase 3 analysis section.

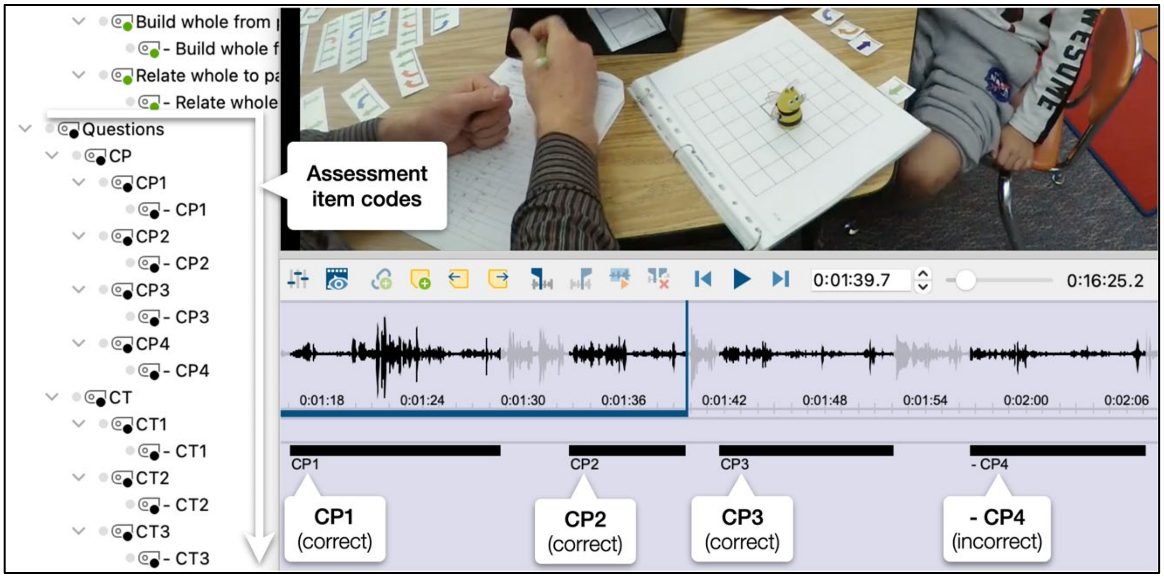
I cleaned the data by isolating the students' interactions with each assessment item across all videos by attaching the timestamps indicating students' interaction with each assessment item with a corresponding assessment item code to indicate which assessment item is being administered and if the student answered correctly or incorrectly (see Figure 10). Correct and incorrect responses were drawn from assessment item scores, which were previously double coded by CiK researchers. Examples of these codes are shown on the left side of Figure 10 under the heading "Questions."

In the example pictured in Figure 10, the student correctly answered CP1, CP2, and CP3, so these items were coded with the assessment item codes *CP1*, *CP2*, and *CP3*, respectively. The student answered CP4 incorrectly, so the item was coded as *-CP4*.

The assessment item codes isolated the unit of analysis for this study's research question 1. Further identifying if the student responded correctly or incorrectly to the assessment item supported question 2's analysis, which examined how students' MK and MK CT co-occurrences relate to students' CT assessment performance on individual assessment items.

Figure 10

Data Preparation Example



Note. Screenshot depicting how I prepared the video data by isolating units of analysis in MAXQDA 2020 with assessment item codes.

Phase 1 Analysis

Question 1: How are kindergarten students' MK and CT operationalized during a CT assessment? In what ways, if any, do MK and CT co-occur?

To answer question 1, I coded the video data using MK, CT, and MK CT co-occurrence a priori codes and open coding (Saldaña, 2021). MK a priori codes originated from a study exploring the co-occurrences of MK and CT skills demonstrated by kindergarten students in small-group coding activities with coding robot toys (Shumway et al., 2021). Shumway and colleagues observed and developed codes for three MK categories: *Spatial*, *Measurement*, and *Number* (see Table 4). MK codes were developed in the context of curriculum tasks; hence, I also applied open coding to account for other MK categories that might have emerged in this study's CT assessment context.

Table 4

Mathematical Knowledge A Priori Codes Adapted from Shumway et al. (2021)

MK	Definition
Spatial	
Spatial orientation	Understands and operates on relationships between different positions in space *(e.g., rotating an agent to face the direction of travel)*
Spatial visualization	Understands and performs imagined transformations of objects *(e.g., mental images of movement in space such as a 90-degree rotation)*
Spatial language	Describes movement of an agent using spatial language accurately (e.g., forward, backward, rotate left) or intuitively *(e.g., straight, down, turn*) or position of the agent relative to locations or objects *(e.g., next to)*
Spatial knowledge in codes	Connects spatial orientation, spatial movement, and spatial language to a representational system *(e.g., codes for the program represented as arrows)*
Measurement	
Units of measure	Understands and operates with a unit of measure, usually one linear forward movement or one 90-degree rotation
Distance measurement	Understands that distance can be measured by units of linear movement either by counting the units of measure or describing or showing a distance between two points. Additional applications include comparing actuated and intended distances with number and codes *(e.g., expressing a distance of linear movement with the total number of linear codes)*
Number	
Counting	Counts movements, objects, or codes
Counting on	Counts on from a space on the grid *(e.g., agent's starting point)* which involves understanding that we are counting the movements of the agent, not the squares on the grid
Coordinating counts	Coordinates the totals of two quantities and/or matches 1-to-1 counting with movements or codes
Operations	Uses addition or subtraction to operate on quantities, including operative language *(e.g., one more, one less)*

Note. This table reflects minor modifications in the examples to Shumway et al.'s original code descriptions. This was done to better describe the codes' application to an assessment context.

Each of the three MK categories (Spatial, Measurement, Number) in Table 4 is broken into more specific categories, listed in the table's left column. The table's right column characterizes each category in relation to a grid-based CT assessment item.

Consistent with embodied cognition theory, I anticipated that the indicators for each MK code would be evident through students' gestures, language, and actions on objects, as this theory affords an interpretation of students' actions/language/interactions as evidence of existing knowledge. My interpretation of these codes was grounded in an enactivist perspective in that the students' engagement with the assessment materials was novel, meaning that the students had not previously engaged with the assessment materials. While the students had participated in classroom CT coding activities, the coding activities used different materials than the assessment. Interacting with the assessment materials required the students to respond to assessment tasks using novel assessment materials, thus demonstrating "learning in action" (Khan et al., 2015, p. 272). Further, embodied cognition informed my interpretation of students' gesture, language, and actions as evidence of students' existing MK and CT knowledge, permitting me to qualitatively code students' indicated MK and CT as existing knowledge.

The CT a priori codes were drawn from the CiK research team's Early Childhood Computational Thinking (ECCT) competency model (see Figure 3 in Chapter II; Clarke-Midura, Shumway, et al., 2021). The ECCT was informed by extensive empirical research conducted by the CiK research team and a CT domain analysis (how the field currently operationalizes CT). The ECCT describes the MK (Spatial reasoning, Sequencing, Linear units, Rotation on a point, Counting on) and CT (Debugging, Algorithmic Thinking, Decomposition) that kindergarten students demonstrate when engaging with CT tasks using screen-free robot coding toys. I used the ECCT's CT subcomponents as a priori codes to identify the CT that kindergarten students indicated during the CT assessment. These subcomponents (defined in Table 5) operationalize how students use algorithmic thinking, debugging, and decomposition.

Table 5

Computational Thinking A Priori Codes and Definitions

CT Subcomponents	Definition
Algorithmic thinking	
Sequence codes	Ordering and arranging codes based on knowledge of syntax and semantics
Plan program	Determining instructions required to successfully reach goal
Read/enact program	Interpreting (reading) and executing (enacting) sequence of codes
Debugging	
Recognize bug	Noticing that instructions do not work as expected, or anticipating a problem before executing the program (i.e., knowing that there is a bug)
Locate bug	Finding the part in the program that caused the problem (i.e., knowing where the bug is)
Propose solution	Making a plan or suggestion for how the program could change (i.e., knowing how to fix it)
Fix bug	Implementing a successful repair strategy (i.e., resolving the bug)
Decomposition	
Break whole into parts	Recognize how whole programs can be broken down into units or segments of code to simplify the problem
Build whole from parts	Writing program by combining chunks or sequencing codes one-by-one
Relate parts to whole	Coordinating units or segments of code with one another as well as with whole program

Each CT component is italicized in Table 5 (Algorithmic Thinking, Debugging, Decomposition). The subcomponents (listed below each component) describe how the components are operationalized within a coding toy context. The subcomponents were developed via empirical, curriculum-based research and informed the CT assessment item development.

Based on the MK CT section of the literature review in Chapter II, I identified additional MK CT codes that might have emerged in open coding (see Table 6). These codes were explored as MK CT codes in Phase 2 of the analysis if they appeared to co-occur as MK and CT.

Table 6

Possible Mathematical Knowledge Computational Thinking Codes and Definitions

MK and CT Code	Definition
Abstraction	Abstraction in CT: "to conceptualize and then represent an idea or a process in more general terms by foregrounding the important aspects of the idea while backgrounding less important features" (Weintrop et al., 2016). Abstraction in Math: a "process of reducing complexity by ignoring irrelevant details in order to focus attention on important elements of a problem, situation, or phenomenon" (Rich et al., 2019, p. 269).
Decomposition	Decomposition in CT: a method to simplify tasks by breaking the tasks into manageable portions and work on one portion at a time (Wang et al., 2021) and "recognizing parts in part-whole relationships, building a whole from parts, and breaking a whole into part" (Clarke-Midura, Shumway, et al., 2021, p. 20). Decomposition in Math: breaking apart quantities, space, and problems while considering part-whole relationships (Shumway et al., 2021).
Develop and apply algorithms	Algorithms in CT and Math: "creating a set of ordered steps…and then performing them in a particular order to accomplish a task in a way that could be repeated by others" (Lavigne et al., 2020, p. 63).
Patterning	Patterning in CT and Math: a set of objects, rules, or actions determined by an underlying structure so that the set's structure may be accurately generalized and extended upon
Sequencing	Sequencing in Math: Attending to the order of objects, actions, and/or processes. Sequencing in CT: "Knowing how to sort and arrange codes and also how to use number sequencing to determine how many movements or codes are needed" (Clarke-Midura, Shumway, et al., 2021, p. 19).
Spatial reasoning	Spatial reasoning in CT and Math: "the ability to recognize and (mentally) manipulate the spatial properties of objects and the spatial relations among objects," (Bruce et al., 2017, p. 146).

Each MK or CT code is listed in Table 6's left column, while each code's definition from existing literature is listed on the right. As these codes emerge, I memoed about how each coded event exhibited MK CT co-occurrences. These memos supported Phases 2 and 3 of the analysis and this study's second research question, which inquired how students' MK and MK CT co-occurrences relate to their performance on individual CT assessment items.

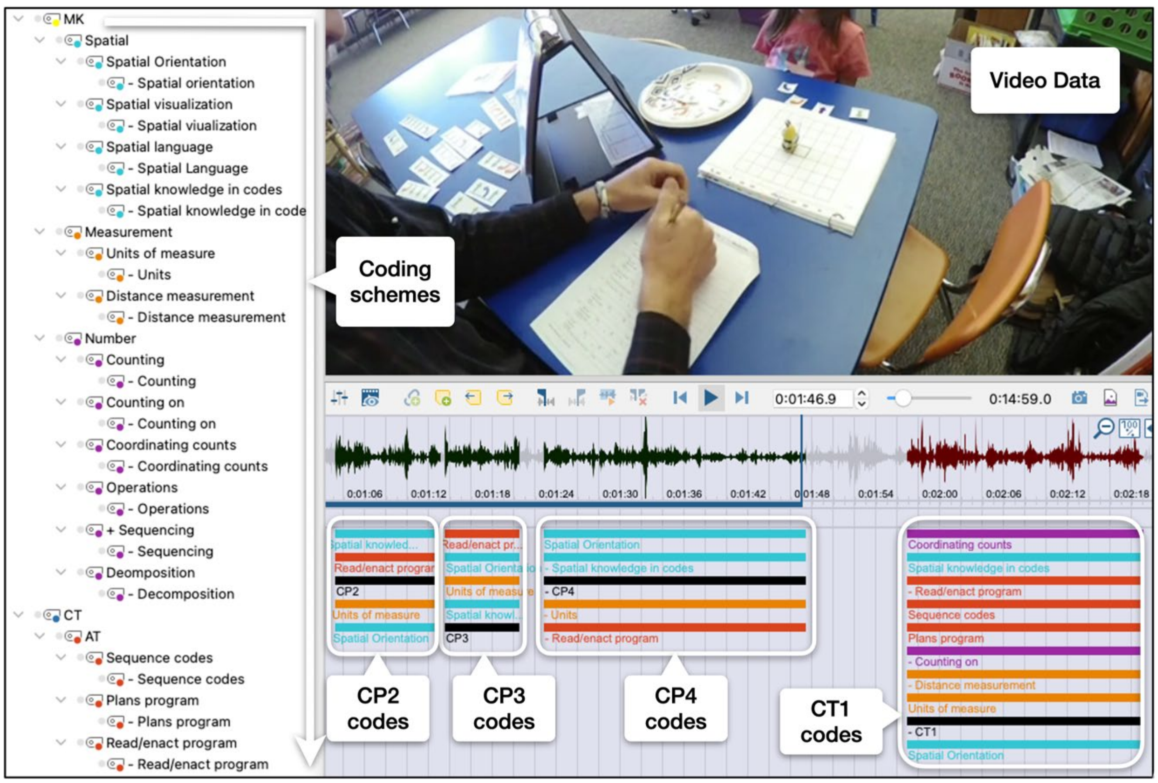
I coded each assessment item occurrence (which was isolated during data preparation) with the a priori codes and open codes. The MK a priori codes represent MK that, according to curriculum standards (CCSSI, 2010) and learning trajectories (Clements & Sarama, 2021), kindergarten-aged students may still be developing. To account for the likely possibility that students may have indicated MK with varying levels of understanding, I coded each a priori code as either indicated knowledge or developing knowledge. Students' gestures (e.g., hand movements, pointing, head bobs), language (e.g., verbal counting, verbal responses, sound-making), and actions on the assessment material (e.g., moving the agent, manipulating coding cards) informed this coding and the results of this coding based on these behaviors are discussed in Chapter IV of this dissertation. When students indicated developing knowledge, I attached a memo to the coded instance describing how the student's language, gesture, and/or actions on the assessment material(s) indicated the student's knowledge as developing. I also memoed specific language and gestures that students used to indicate MK and CT.

An example of a student who indicated counting knowledge (coded as *Counting*) used a correct counting sequence to identify the total number of codes in a program.

Conversely, a student who indicated developing counting (coded as *-C* to distinguish it from the indicated code *Counting*) also counted all the program codes in a program but counted one of the codes twice. Figure 11 illustrates how the final codes appeared when coded in MAXQDA.

Figure 11

Phase 1 Coding Example



Note. Screenshot coding in MAXQDA 2020. The codes for developing knowledge are indicated by a "-". For example, the code *Counting on* was attached to instances as indicated knowledge and -*Counting on* was attached to instances of developing knowledge. For ease of reading this document, developing knowledge codes will be abbreviated to further distinguish them from their indicated counterparts.

As shown in Figure 11, I input the coding schemes in MAXQDA (left column) and attached corresponding codes to the student's engagement with each assessment item. During the data preparation stage, each assessment item was isolated using assessment item codes (indicated by black bars below the video data). MK, CT, and MK CT codes were then attached to each item, as highlighted in Figure 11.

For example, Figure 12 isolates, more specifically, how CP3 and CP4 were coded for one student's engagement with the items.

Figure 12

CP3 and CP4 Isolated Coding in MAXQDA

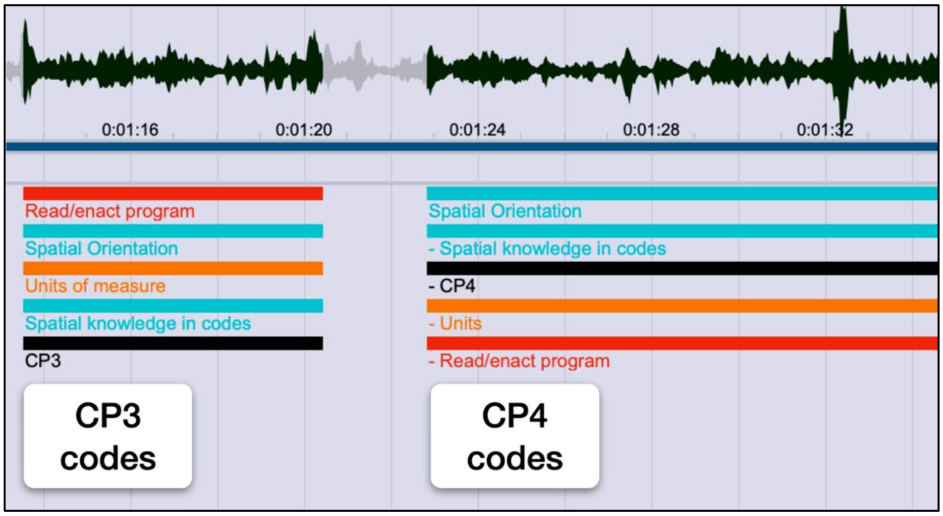
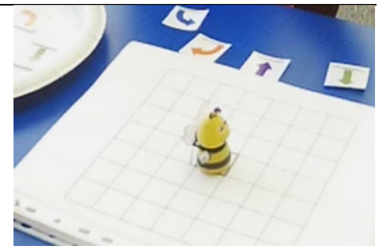
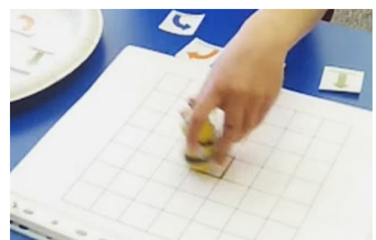
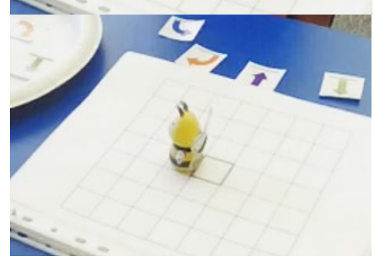


Figure 12 shows that the student answered CP3 correctly and indicated knowledge of spatial orientation, units of measure, spatial knowledge in codes, and reading/enacting the program. Conversely, the student answered CP4 incorrectly and indicated spatial orientation knowledge but also a developing knowledge of spatial knowledge in codes, units, and enacting the program. To further illustrate this process, Table 7 describes specifically how each code was attached to a student's engagement with CP4.

Table 7

Example of Coding for CP4

Transcript	Codes and justification	Image
Teacher: With bumblebee facing this way *((points to agent/bee))*, could you tell me what this code *((points to right-turn arrow))* would make it do?		
Student: *((rotates the agent counter-clockwise 180 degrees, then briefly pauses))*	**Spatial orientation**: Student matches the agent's orientation with the arrow's direction. **- SK**: The right-turn arrow requires one 90-degree rotation to the agent's right	
Student: *((moves the agent one square to the right))*	**- SK**: The right-turn arrow does not require linear movement **- UM**: The student modeled two distinct units (one 180-degree rotation, one linear movement) for one code **- REP**: The student incorrectly enacted the program (one right turn)	 

Note. The student is positioned at the top of each picture, sitting directly across from the teacher. All directions are written to the child's perspective, so the right-turn arrow appears to the student as such, whereas in the teacher's perspective the right-turn arrow appears to be pointing left. -SK = Developing Spatial knowledge in codes; -UM = Developing Units of measure; -REP = Developing Read/enact program.

In Table 7, the first column contains the transcript of teacher and student language, gesture, and movement during the administration of CP4. The middle column lists the attached codes with justification for their application and are examples of how these items were memoed. The right column includes screenshots for further illustration.

The output for Phase 1 was the coding frequencies for the MK and CT demonstrated by all students in the data set. To respond to the first part of research question 1's objective (identifying how MK and CT are operationalized during a CT assessment) I used a joint embodied and enactivist lens to operationalize students' language, gesture, and actions on objects within the three MK categories (spatial, measurement, and number) and the three CT components (algorithmic thinking, debugging, and decomposition). I also attached memos to instances when students indicated developing MK and CT knowledge to describe how students' indications of accurate and developing MK were interpreted.

Using language, gestures, and actions on objects to interpret how students operationalize MK and CT reflected this study's embodied perspective as the students using existing knowledge. An enactivist perspective further supported this analysis by considering students' developing abilities and was used to describe how students' actions were indicative of knowledge development, or knowledge in transition.

Phase 2 Analysis

Question 1: How are kindergarten students' MK and CT operationalized during a CT assessment? In what ways, if any, do MK and CT co-occur?

The second objective for question 1 was to identify if, and if so what ways, MK and CT co-occur during a CT assessment. To do this, I generated a MAXQDA co-occurrence frequency report (see Appendix E). A portion of this report is shown in Table 8 for illustrative purposes.

Table 8

Mathematical Knowledge Computational Thinking Co-Occurrence Frequency Report Excerpt

	CT codes			
MK Codes	Builds an intended algorithm	- BIA	Read/enact program	- REP
Spatial visualization	10	12	9	6
- SV	0	1	0	0
Spatial language	24	37	31	30
- SL	0	3	0	3
Spatial knowledge in codes	129	5	133	4
- SK	0	154	3	135
Units of measure	60	25	133	43
- UM	0	58	1	87

The report (see Table 8) is organized in a table with the MK codes listed vertically in the left column and the CT codes listed horizontally at the top. To easily differentiate indicated from developing knowledge codes, indicated knowledge codes are spelled out (i.e., *Spatial language, Read/enact program*) and codes to indicate knowledge in development are abbreviated with a dash in front (i.e., *- SL, - REP*). For coding purposes, I retained the entire code description for all developing knowledge codes in MAXQDA. The values reported within the table indicate the frequency of specific MK CT co-occurrences across all students. For example, in Table 8 the row for "Units of measure" and column for "Read/enact program" intersect to show 133 instances of co-occurrences. This report responded to question 1's inquiry by identifying which MK and CT students demonstrate while interacting with the CT assessment items, and further operationalized how students indicate MK and CT knowledge within a CT assessment. To report on these co-occurrences, I identified the most frequent co-occurrences and analyzed video clips of students who indicated these co-occurrences and reviewed attached memos to generalize the conditions of the co-occurrences. This was done to ascertain why each co-occurrence so frequently appears.

Phase 3 Analysis

Question 2: How do students' MK and MK CT co-occurrences relate to their performance on individual CT assessment item?

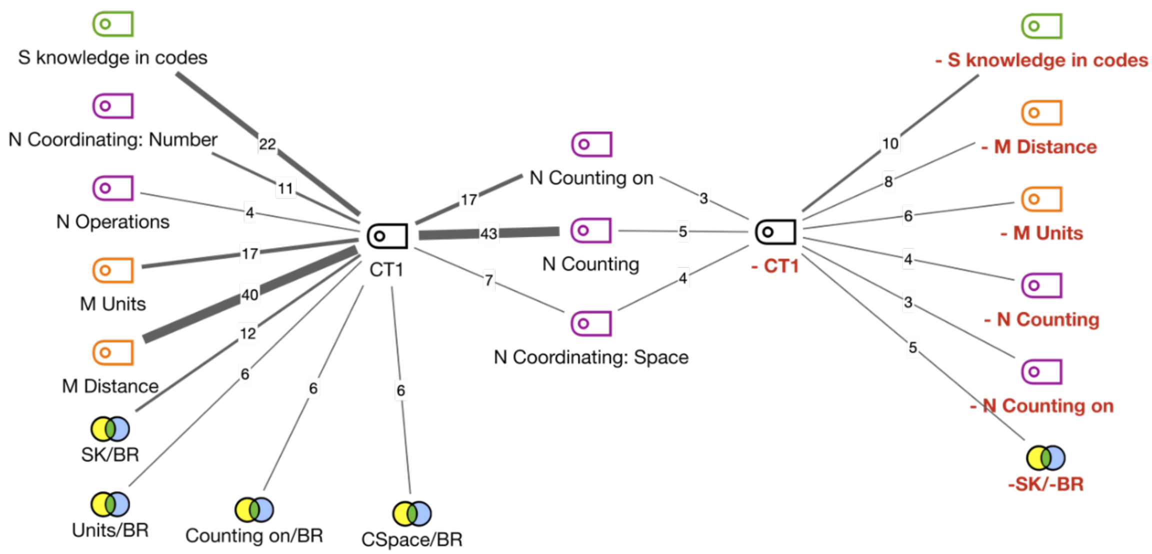
I used the results for research question 1 to develop multiple-case studies (Yin, 2018) to describe how students' MK and MK CT co-occurrences might relate to their CT assessment item performance, then synthesized the case studies in a cross-case report. These cases further described the connections and relationships between MK and CT in a CT assessment context and described how MK CT co-occurrences might relate to students' CT assessment item performance. The unit of analysis for each case study was an assessment item. I conducted four case studies, each of a different CT assessment item. Each case described how either a different MK or MK CT co-occurrence relationship manifested in students' assessment item performance and how the MK or MK CT co-occurrence might relate to students' performance on the item.

I selected case studies by drawing on findings for question 1 and generating visual case models for each CT assessment item in MAXQDA (see Appendix D). Each case model depicts the MK and MK CT co-occurrence exhibited by students who completed the respective assessment item correctly and incorrectly (see Figure 13).

For example, Figure 13 depicts the visual case model for assessment item CT1.

Figure 13

Mathematical Knowledge/Mathematical Knowledge Computational Thinking Case Model of Student Performance on Item CT1



The model's central points (CT1, -CT1) represent instances when students responded correctly (labeled CT1) and incorrectly (-CT1) to item CT1. The model's nodes connect the central points with the MK and MK CT co-occurrences students indicated when responding correctly (CT1) and incorrectly (-CT1) to the assessment item. Since the analysis codes were attached to the span of time that each student engaged with the relative assessment item, some nodes indicated a frequency less than or equal to the number of times that an assessment item was administered across the entire data set. The specifics of these diagrams and an explanation of how to read them are discussed in more detail in the *Results for Research Question 2…* section in Chapter IV.

An enactivist lens permitted me to theorize how students' gestures, movements, and language might support MK and CT development simultaneously and contribute to the field's theory of how we view MK and CT knowledge at this age and how children express their thinking and learning.

Validity and Trustworthiness

Validity and trustworthiness considerations were embedded throughout each phase of this study. For phase 1, I used a priori and open coding to operationalize students' MK and CT when interacting with the CT assessment items. Using a priori codes situated this research within existing work and extended upon prior research findings. Additionally, I helped to develop the a priori codes developed in a previous study, so I have a deep understanding of their meaning and how these codes were previously used. This strengthened the coding validity as I applied the a priori codes to data set.

For phase 2, I attended to trustworthiness by identifying MK CT co-occurrences based on code frequencies rather than anecdotal evidence. This helped me distance my personal biases by relying on the coding results rather than personal interests in specific MK.

Finally, in phase 3 I chose cases based on evidence that emerged in phase 2. This extended the trustworthiness considerations in place for research question 2 as I selected cases that accurately represent the MK and MK CT co-occurring knowledge that students indicated and their subsequent assessment item performance.

Summary

I used existing data to conduct a sequential, three-phase qualitative design to operationalize kindergarten students' MK and CT during a CT assessment and how students' MK and MK CT co-occurrences might relate to their CT assessment performance. The unit of analysis for question 1 was the student, whereas question 2's unit of analysis was individual assessment items. I addressed question 1 by using a priori and open coding to identify how 60 students indicated MK, CT, and MK CT co-occurrences during a CT assessment, then used a subsequent frequency report to identify MK and CT co-occurrences. I coded the dataset using an embodied cognition and enactivist lens to interpret students' language, gestures, and actions on objects as knowledge in action and knowledge as action respectively. Question 2 was answered using multiple-case studies with an embodied cognition and enactivist lens to further understand the role of MK and MK CT co-occurring knowledge in kindergarten students' CT assessment item performance. This analysis contributes empirically to the field by describing the deep connections between MK and CT and how they might relate to each other, as well as theoretically by describing how students' language, gesture, and actions indicate their MK, CT, and MK CT co-occurring knowledge within a CT assessment context.

CHAPTER IV

RESULTS

The purpose of this study was to operationalize how kindergarten students indicated MK and CT while solving CT assessment items and to understand how the MK and MK CT co-occurrences might relate to the students' assessment performance. The research questions guiding this study include the following.

1. How are kindergarten students' MK and CT operationalized during a CT assessment? In what ways, if any, do MK and CT co-occur?
2. How do students' MK and MK CT co-occurrences relate to their performance on individual CT assessment items?

The findings presented in this chapter are a result of applying qualitative methods to analyze video data of 60 kindergarten students engaging with a CT assessment. First, I will present findings for research question 1 detailing the fine-grain analysis of students' demonstration of mathematical and CT knowledge and how I operationalized students' MK and CT as they engaged with a CT assessment. Next, I describe the emergent themes related to how MK and CT co-occur during kindergartners' engagement with the assessment. Finally, I present the results for research question 2 with four case studies (and subsequent cross-case conclusions) that describe the MK and MK CT co-occurring knowledge that might relate to students' performance on individual assessment items.

Results for Research Question 1

Students' Demonstration of Mathematical Knowledge and Computational Thinking During a Computational Thinking Assessment

In this section, I present the results for Research Question 1, *How are* *kindergarten students' MK and CT operationalized during a CT assessment? In what ways, if any, do MK and CT co-occur?* To answer Research Question 1, I used a priori and open coding methods to code video data of 60 kindergarten students engaging with 14 items in a pilot CT assessment. Before reporting how kindergarten students' MK and CT were operationalized during the CT assessment, I first present the overall frequencies of students' demonstration of accurate and developing mathematical and CT knowledge. Table 9 summarizes the overall frequencies of MK and CT knowledge codes.

The first column of Table 9 categorizes the codes as MK or CT. The second column labels the categories of MK (Spatial Knowledge, Measurement Knowledge, Number Knowledge) and CT (Algorithmic Thinking, Debugging, Decomposition). The third column lists the codes used to evidence knowledge in their respective category as either indicators of knowledge or indicators of developing knowledge. For example, coded indicators of the MK category Measurement include units of measure and measurement distance. When a student displayed indicators of units of measure knowledge such as moving the assessment agent in linear units consistent with the assessment's rules, the *Units of measure* code was applied. However, when a student indicated developing units of measure knowledge, such as when a student moved the agent in linear units of different lengths, the -*UM* code was used. As described in Chapter III, indicated knowledge codes are spelled out (i.e., *Units of measure, Decomposition*) and the codes indicating knowledge in development are abbreviated with a dash in front (i.e., - *UM,* - *DC*). The fourth column in Table 9 lists the frequencies of each code's occurrence across all 60 students' engagement with 14 assessment items. Finally, the fifth

Table 9

Mathematical Knowledge and Computational Thinking Code Frequencies

MK/ CT	Knowledge category	Knowledge skill code	Frequency	Ratio of developing knowledge to total indicators of knowledge
MK	Spatial	Spatial visualization	35	0.03
		- SV	1	
		Spatial language	121	0.03
		- SL	4	
		Spatial knowledge in codes	449	0.43
		- SK	344	
	Measurement	Units of measure	393	0.30
		- UM	167	
		Distance	41	0.16
		-D	8	
	Number	Coordinating counts: Space	453	0.13
		-CC:S	65	
		Coordinating counts: Number	34	0.08
		-CC:N	3	
		Counting	91	0.05
		-C	5	
		Counting on	416	0.02
		-CO	8	
		Operations	22	0.00
		-O	0	
CT	Algorithmic Thinking	Plans program	4	0.00
		-PP	0	
		Builds intended algorithm	129	0.55
		-BIA	160	
		Read/enact program	138	0.50
		-REP	138	
	Debugging	Recognize bug	31	0.30
		-RB	13	
		Fix bug	26	0.26
		-FB	9	
	Decomposition	Decomposition	53	0.15
		-DC	9	

column presents the ratio of indicators of developing knowledge for each skill. To use units of measure knowledge as an example, the ratio of developing units of measure knowledge (-*UM*; N = 167) to all indicators of units of measure knowledge (N = 560) is 0.30. Stated otherwise, of the 560 instances that students indicated units of measure knowledge or units of measure developing knowledge, 30% of the instances indicated developing knowledge.

Table 9 shows *Spatial knowledge in codes* as the most frequently coded skill. *Spatial knowledge in codes*' high frequency suggests it is an important skill in students' CT assessment performance. *Spatial knowledge in codes* also has the highest ratio of developing knowledge suggesting that it may be the more difficult mathematics skill in the CT assessment. *Operations* has the lowest MK frequency, which is consistent with the CT assessment items' design. The assessment items do not require students to conduct numerical operations, so operation knowledge and developing knowledge indicators would likely have been student motivated. Notably, knowledge from all MK categories is represented, suggesting that CT assessment performance requires students to draw on varied MK skills.

Among the CT knowledge, *Builds intended algorithm* and *Read/enact program* were the most frequently coded. Additionally, *Builds intended algorithm* and *Read/enact program* are the only knowledge types with a ratio of developing knowledge to indicated knowledge equal to or greater than 0.5. This may be the case because students are being introduced to a new process. When considered alongside the ratio for MK's *Spatial knowledge in codes*, the related component is that the assessment items ask students to operate a new coding system by using a set of codes (*Spatial knowledge in codes*) to build (*Builds intended programs*) and enact or describe (*Read/enact programs*) programs. These three CT codes represent the assessment's primary tasks as well as the novel knowledge that students encounter while engaging with each item.

Operationalizing Mathematical Knowledge and Computational Thinking Through Gestures, Language, and Action on Objects

As shown in Table 9, students who engaged in the CT assessment exhibited knowledge indicators and developing knowledge indicators of MK and CT. An embodied cognition lens allowed me to interpret knowledge and developing knowledge indicators as existing knowledge, whereas enactivism provided a lens to interpret knowledge and developing knowledge indicators as evidence of evolving knowledge. Students indicated knowledge and developing knowledge while engaging in the CT assessment items through gesture, language, and actions on objects. These behaviors as indicators align with enactivism's perspective of knowledge as fluid and housed jointly in the mind-body-environment. All three indicators (gestures, language, actions on objects) engage the student's thoughts (mind) and biological self (body) within the assessment context (environment). This section describes how MK and CT knowledge and developing knowledge were operationalized through students' multimodal and specific gestures, language, and actions on objects (i.e., interactions with materials). While the following sections describe how students demonstrated knowledge using each modality in isolation, the three modalities regularly co-occurred.

In accordance with this study's methodology (described in Chapter III), I coded the data to identify when students indicated knowledge or developing knowledge for each code. This was done to account for students' varying levels of MK and CT understandings. Distinguishing students' developed and developing knowledge allowed me to operationalize how students indicated their MK and CT at varying levels of development. Operationalizing students' developing knowledge provides a more detailed perspective of how students utilize MK and CT in this context. While a thorough examination of finite knowledge levels is outside the scope of the present study, operationalizing students' knowledge as indicated or developing is within this study's scope and could inform related work. First, students' indicators of spatial knowledge will be presented, followed by measurement knowledge, number knowledge, and, lastly, CT knowledge.

How Students Demonstrated Spatial Thinking

Students' spatial knowledge was observed in students' evidence of spatial visualization, spatial language, and spatial knowledge in codes. Table 10 summarizes the ways students indicated spatial knowledge and developing spatial knowledge. All indicators presented in this table, and subsequent MK and CT indicator tables, were generalized from memos taken during the qualitative coding analysis. The first column in Table 10 lists the codes associated with spatial knowledge, the second column lists the knowledge indicators used to operationalize each respective code, and the final column lists the indicators of developing knowledge for each code. Similarly organized tables will be presented later for measurement knowledge, number knowledge, and CT knowledge.

Table 10

Spatial Knowledge Indicators and Indicators of Developing Knowledge

MK	Indicators	Indicators of developing knowledge
Spatial visualization	Traces finger along grid path Points with finger to indicate linear or rotational direction Gestures while describing movement Taps/touches grid squares Compares robot's perspective compared with rotation options Holds a turn code up to grid	Only outward indicators evidencing students engaging with spatial visualization were recorded, so there were no observable behaviors indicating students' developing visualization knowledge.
Spatial language	Says: *Here/Over here/Right here* *There/Over there* *Over* *This way/That way* *Turn/Turn around* *Straight* *Forward/Up* *Go back/Backward* *Down* *Forward* *Left* *Right* *Rotate* *Diagonal*	Incorrectly uses the terms right and left Names codes with incorrect spatial language (i.e., stating "forward" for an R code)
Spatial knowledge in codes	Builds a program correctly Successfully fixes a programming error Enacts code units correctly Correctly names a code Correctly states what a code does	Builds a program incorrectly States that a code does something other than it is designed to do Enacts a code with the wrong unit (i.e., F as two forwards, R/L=180° rotation, R/L= rotate and move) Enacts a code incorrectly

Note. Program codes are abbreviated within this table and throughout the chapter as follows: F = forward, B = backward, R = rotate 90° right, L = rotate 90° left.

As shown in Table 10, students generally indicated spatial visualization knowledge with gestures (i.e., taps, points), spatial language knowledge with language (i.e., down, diagonal, up), and spatial knowledge in codes with actions on objects (i.e., building programs, enacting codes/programs). While these specific spatial knowledge types can be generalized by modality, they are not exclusive. For example, students who indicated spatial visualization by gesturing on the grid to describe an agent's movement combined gesture and language modalities. While students indicated knowledge multimodally, presenting knowledge indicators by modality provides insight into how students' knowledge is embodied.

Gesture indicating spatial knowledge. Students indicated spatial knowledge when they gestured with their hands and arms directly in front of them or over the grid. Some students gestured directly in front of them by pointing their fingers in various directions, which evidenced spatial visualization. Students also gestured over the grid space by tracing or touching grid squares along the agent's path or anticipated path. Students sometimes combined gestures over the grid space with spatial language by describing the movements that each gesture represents (see the *Language Indicating Spatial Knowledge* section). Gestures indicating developing spatial knowledge were not observed. This may be because most gestures indicated spatial visualization and students' spatial visualization behaviors were broad and did not require precision.

Language indicating spatial knowledge. Only assessment item CT1 required students to verbally respond, which asks students how many forwards the agent needs to reach a destination. Students' language use outside of assessment item CT1 was unprompted. For CT1 students responded either verbally by stating a quantity, nonverbally with fingers or program codes, or a combination of verbal and nonverbal response methods.

Three categories of spatial language types emerged from the code memos. I termed these categories as referential language, general directional language, and precise directional language. Students used *referential language* (this way, that way, here, there) when indicating a location (here, there) and to indicate a direction or movement specific to the student's referent such as the agent or position on the grid (this way, that way). *General directional language* (turn, straight, up, down) included directional language not specific to the assessment, whereas *precise directional language* included language specific to the assessment and the programming codes (forward, backward, left, right, rotate). Students used general directional language to describe agent movements on the grid from the student's perspective. For example, students used "up" or "straight" to indicate the robot moving north on the grid, whereas students used "down" or "straight" to indicate the robot moving south on the grid. Students also used straight to indicate the agent's linear movement east or west. The precise directional language for up, down, or straight would have been "forward." Students used "turn" to describe the agent rotating in place on the grid or rotating and moving to a new location on the grid. Precise directional language includes spatial language specific to the assessment and attends to the agent's orientation. For example, students described the agent's movements as "forward" and "backward" with respect to the agent's orientation, as well as "left" and "right" rotations. When using the word "rotate," the students demonstrated the agent rotating in place without traveling to another square. These spatial language categories were also observed in students' indicators of number language and are discussed in the subsequent *Language Indicating Number Knowledge* section.

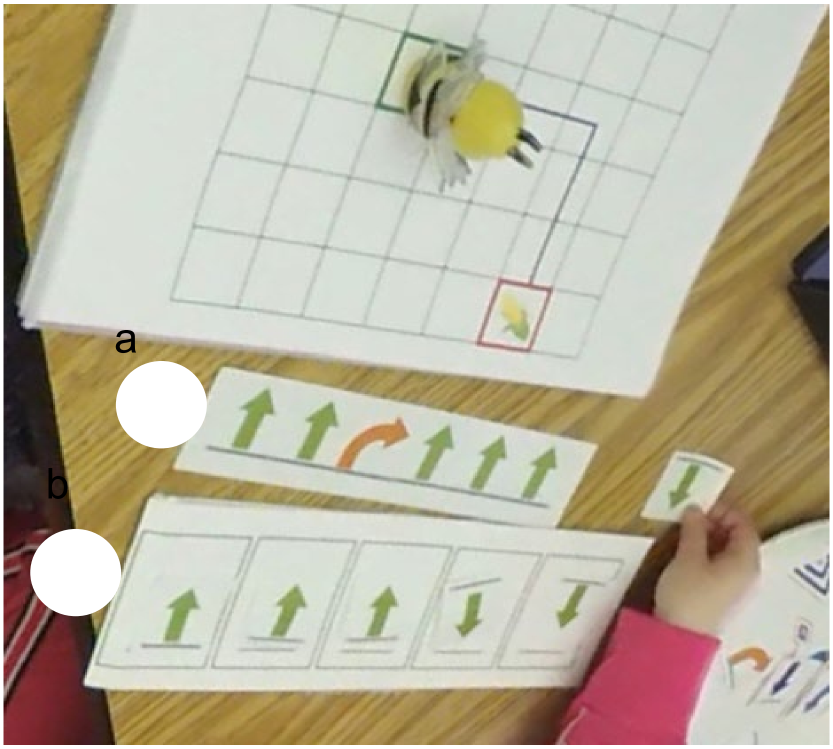
Students used language from each category in isolation or in combination. For example, one student used referential language while enacting the program FFF for assessment item CT2. He moved the agent one linear length at a time, stating “this way, this way, this way.” Another student engaged with assessment item CT9 (see Figure 14) by first using precise language, then a combination of precise and general spatial language. The student first enacted the program FFRFFF by naming each code as “forward, forward, forward, rotate right, forward, forward, forward” respectively while enacting the program (Figure 14a). Next, the student combined precise directional language with general directional language while attempting to fix the given program by building the program FFFFFF, naming the first three F codes as forwards and the final three F codes as downs (Figure 14b).

The student in Figure 14 indicated spatial language knowledge by using general and precise directional language while moving the agent and building a program. Although the student used the F program codes to indicate both forward and “down” directions, the spatial language used indicates that the student can attach correct spatial language to her intent for the agent to move “down” (south) on the grid according to her perspective.

Language indicators of developing spatial knowledge. Indicators of developing spatial knowledge emerged when students used the vocabulary “right” and “left” incorrectly or attached precise language incorrectly to program codes. For example, one student built the program RRRBBB to debug item CT9. The student read the program as “forward, forward, forward, down, down, down.” This student indicated developing spatial knowledge via language by attaching the precise directional word “forward” to the R program code. The student’s incorrect use of language indicated that they recognized

Figure 14

A Student Engaging with Assessment Task CT 9



Note. The student enacted the given program (a) and attached each action with precise directional language. Next, the student coded a debugging solution (b) and placed each program code while using a combination of precise and general directional language.

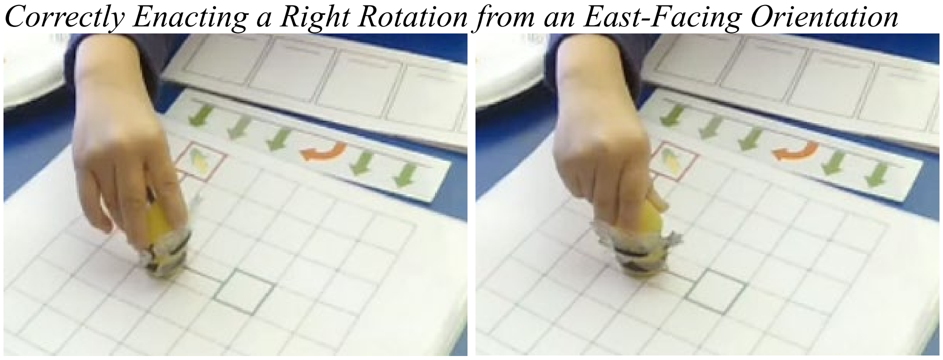
that each program code represented a directional movement, but that the student was still learning each program code's precise language and meaning.

Actions on objects indicating spatial knowledge. This section describes how students indicated spatial knowledge while manipulating the assessment agent and program codes. Students used the assessment agent to model program code meanings in assessment items CP1–CP4 then used the agent to enact given and written programs for the remainder of the assessment. Knowledge indicators of spatial knowledge of program code meanings occurred when students used the agent to correctly enact program codes as incremental linear or rotational movements while attending to the agent's orientation. For example, assessment item CT2 asked students to enact the program FFF with the agent beginning in an east-facing orientation on the grid. Students who correctly answered assessment item CT2 demonstrated an understanding that each F program code is enacted as one unit of forward linear movement with respect to the agent's starting orientation.

Students also attended to the agent's starting orientation when indicating spatial knowledge of the R and L rotation program codes. For example, assessment item CT9 requires students to debug a given program (FFRFFF; correct program FFFRFFF) with the agent starting in an east-facing direction on the grid (see Figure 15).

Figure 15

Correctly Enacting a Right Rotation from an East-Facing Orientation



Note. A student demonstrates spatial knowledge in codes by enacting a given program's right rotation as a 90°, clockwise rotation. This image is from the assessor's perspective.

As shown in Figure 15, students who correctly enacted the program's right rotation rotated the agent 90° clockwise so that the agent, which initially faced east on the grid, rotated in place to face south on the grid. In doing so, the student demonstrated attention to the program code's incremental nature as a 90° turn with respect to the robot's beginning orientation.

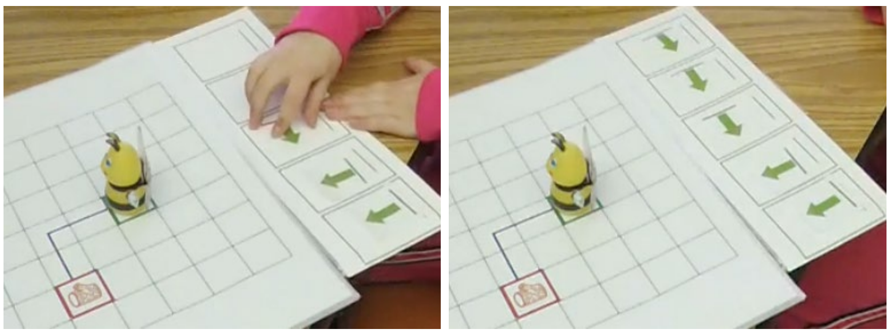
Indicators of spatial knowledge also occurred when students manipulated the assessment program codes. Students interacted with the program codes to sequence programs and to identify which program code is required for a task. The two ways students sequenced programs with program codes was in a horizontal row below the grid and along the agent's path on the grid. To build a program, students laid program codes in a desired sequence from left to right below the grid or from start to end along the agent's path on the grid. Students sometimes held a program code up to the grid to identify if the program code was needed. This most often occurred when a student was determining if a right rotation or left rotation was required. For example, assessment item CT3 provides students with an incomplete program and asks students to fill in the missing program code (missing code is L). Students who recognized that the missing code was a rotation sometimes held up a R or L program code to the grid space before either selecting the chosen program code as the missing code or swapping it for the opposite rotation code.

Students also combined their actions on the agent and program codes by placing a program code down then enacting the program code with the agent, then repeated the process until an entire program was built. This method of building a program by adding a program code and testing it indicated students' knowledge of spatial code meanings as students enacted each code on the grid.

Another way students used actions on objects to indicate spatial knowledge was by orienting program codes in the cardinal direction that the student intended the agent to travel. For example, one student built a program for CT7 (correct response FFLFF) by first sequencing FF, then adding three more F codes positioned to face west on program organizer (see Figure 16).

Figure 16

Student Cardinally Orienting Program Codes



Note. A student demonstrating spatial knowledge by positioning arrows in the cardinal direction of the agent's travel. The student builds a program by orienting the first two F codes north on the program organizer and the final three F codes west.

By orientating the program codes in the cardinal direction of the agent's intended movement along the path, the student pictured in Figure 16 indicated a spatial understanding of how the agent should move in space. The first two codes (FF) point north to indicate that the agent moves north according to the agent and student perspective. By positioning the final F codes facing west, the student indicated an understanding that the agent should travel in a left direction according to the student's perspective. This indicates that the student understands how the agent should move in space to reach the destination, but that the student is favoring their own perspective rather than adopting the agent's perspective. If the student had adopted the agent's perspective, the student would have sequenced all F codes facing north. Positioning all F codes facing north would have indicated that the student adopted the agent's perspective and recognized that the code's cardinal position does not impact the agent's travel, as "forward" will translate as forward movement in the direction the agent is facing.

Actions on objects indicators of developing spatial knowledge. Indicators of spatial knowledge in development emerged in instances of spatial knowledge in codes and spatial language. Similar to gestures indicators, I categorized all instances of spatial visualization as indicators of knowledge since outward indicators only evidenced students engaging with spatial visualization with no distinction of students' visualization in development. For example, students' gestures while describing movements indicated outwardly that the students are visualizing how the agent will move in space, however it is unclear what the student was specifically visualizing. As such, it was not possible to distinguish spatial visualization indicators as evidence of indicated knowledge or developing knowledge.

Students' actions with the assessment agent indicated developing spatial knowledge when they enacted program codes incorrectly or reoriented the agent to face a program code's cardinal direction regardless of the agent's original orientation. Students incorrectly enacted the F program code in a variety of ways (see Table 11). By enacting the F program code incorrectly, these students indicated developing spatial knowledge in codes knowledge and their interpretation of the program codes at the time of enactment. Identifying how students interpreted the program codes furthers this study's objective of operationalizing students' MK of spatial knowledge when applied to a CT context.

Table 11 describes the correct and incorrect ways that students enacted the F and L codes with the assessment agent. The first column categorizes each corresponding row

Table 11

Students' Correct and Incorrect Enactions of Program Codes in Relation to the Agent's Starting Orientation

Code enacted as correct/incorrect	Agent's starting orientation			
	North	East	West	South
Forward code enacted correctly (Spatial knowledge in codes)	One linear forward unit north	One linear forward unit east	One linear forward unit west	One linear forward unit south
Forward code enacted incorrectly (- SK)	Forward two or more units Forward to the end of the grid	Forward two or more units Sidestep one unit to the north so that the agent remains facing east Rotate 90° counterclockwise to face north Rotate 90° counterclockwise to face north and forward one unit	Sidestep one unit to the north so that the agent remains facing west Rotate 90° clockwise to face north Rotate 90° clockwise to face north and forward one unit	Rotate 180° to face north Rotate 180° to face north and forward one unit
Left rotation code enacted correctly (- SK)	One 90° counterclockwise rotation to face west	One 90° counterclockwise rotation to face north	One 90° counterclockwise rotation to face south	One 90° counterclockwise rotation to face east
Left rotation code enacted incorrectly (- SK)	Rotate 90° counterclockwise to face west and forward one unit	Forward one unit Rotate 180° to face west Rotate 180° to face west and forward one unit	Forward one unit	Rotate 90° clockwise to face west Rotate 90° clockwise to face west and forward one unit

Note. Right rotations have the same indicators as left rotations with opposite rotation directions.

as descriptors of students' enaction of a specific program code (F, L) and if the descriptors reflect a correct or incorrect enactment of the program code. The remaining four columns describe how students demonstrated the F and L program codes correctly and incorrectly with the assessment agent when the agent's beginning orientation faced to the assessment grid's north, south, east, or west.

When the agent began with an east-, west-, or south-facing orientation, students who displayed indicators of developing spatial knowledge enacted the program code F by moving the agent forward two or more spaces, sidestepping the agent north while maintaining the agent's perspective, or rotating the agent to face north. By rotating the robot to face north, students aligned the agent with the program code's cardinal direction. Students similarly attended to the cardinal directions of the program codes R and L when enacting R and L with the agent (see Table 11).

When enacting either the R or L program codes, students who indicated developing spatial knowledge enacted the rotation by facing the agent in the program code's cardinal direction with or without a forward linear unit. The students who enacted program codes R or L as a forward linear unit in the opposite direction of the arrows cardinal direction did so only near the beginning of the assessment for items CP2 and CP4. Observed through an enactivist lens, students enacted R and L in opposite directions of the arrows' cardinal direction near the assessment's beginning and learned over the course of the assessment to enact R and L correctly or in the program code's cardinal direction.

How Students Demonstrated Measurement Knowledge

Measurement knowledge was operationalized through students' indicators of units of measure and distance measurement knowledge. The assessment's F and B program code units are one linear movement horizontally or vertically from the center of one square to an adjacent square. The R and L rotation program code units are 90° rotations in the code's respective direction (R=clockwise, L=counterclockwise). Table 12 summarizes the analysis memos that described how students indicated measurement knowledge in the

Table 12

Measurement Knowledge Indictors and Indicators of Development

MK	Indicators	Indicators of development
Units of measure	Enacts agent movements incrementally Names an action or makes a sound for each agent movement Counts individual agent movements to determine number of F arrows needed Moves the agent one unit at a time or coordinates a code with one unit at a time	Enacts the agent with units different than indicated in the assessment Enacts one code as a continuous linear movement Enacts one code with two or more movements (e.g., R = turn and move) Enacts a rotation as 180° Moves the agent diagonally Includes the start space when enacting a program Includes start space when counting the number of F codes needed
Distance measurement	States the correct number of F codes needed to travel from one point to another	States how many forward codes are needed but enacts a different number States the incorrect number of F codes needed

assessment through indicators of units of measure and distance measurement knowledge.

As shown in Table 12, students indicated units of measure (abbreviated as UnitsM) knowledge and developing knowledge in multiple ways while fewer distance measurement indicators emerged. UnitsM knowledge was coded for more frequently than distance measurement knowledge (see Table 9). Also, students indicated UnitsM knowledge in a greater variety of ways than other measurement knowledge. The higher occurrence of UnitsM knowledge in the data might explain why more indicators of this knowledge type manifested than distance measurement indicators. Students indicated measurement knowledge most frequently through language (e.g., “states the correct number of F codes…”) and actions on objects (e.g., “enacts agent movement incrementally”) while gestures indicators were not frequently observed. The following sections will discuss how students used these modalities to indicate measurement knowledge and developing knowledge.

Gesture indicating measurement knowledge and developing measurement knowledge. As described in this chapter's spatial knowledge section, students tapped grid spaces to indicate the agent's path and movement, however these gestures only indicated where students expected the agent to travel. These gestures did not indicate precisely measured units nor distinguish how the students perceived the agent's path from the agent's units. To indicate measurement knowledge and developing knowledge with gesture, the student would have needed to demonstrate surrogate embodiment of the agent by tracing their finger along the path with noticeable pauses between units. Alternatively, students would have needed to combine gesture with language to show the distance between two points and describe it as a total length or gesture over the grid to compare the length between two distances.

Language indicating measurement knowledge. Language indicators of UnitsM knowledge emerged when students moved an agent and named each movement unit or made a sound for each movement. For example, a student enacted the program FFLFF (item CT7) while naming each movement as "straight, straight, turn, straight, straight." Another student enacted FFLFF while making a "beep" sound for each movement. Associating sounds and language with each unit movement was an auditory indicator that the student associated each movement as a singular unit.

Distance measurement knowledge indicators arose when students stated the number of forward codes needed to direct the agent from one location to another either by counting the number of movements the agent requires and stating the total, or simply stated the number of forward codes needed. One student verbally compared the distance of the program for item CT4 with the program for CT5 by stating that, “it’s three again, they’re both three!”

Language indicators of developing measurement knowledge. As described in Table 12, students indicated developing measurement knowledge with language when stating the incorrect number of forward codes needed and in their multi-modal use of language with actions on objects (language was incongruous with agent enactions) and language with gestures (counting grid squares to identify the number of forward units needed). For instance, some students answering item CT1 stated that the agent required 1 or 3 forward units to travel to a destination (correct answer 2) but moved the agent two incremental units to the destination without changing their initial answer or appearing to rethink their answer.

Some students also indicated developing measurement knowledge by counting the number of squares along a linear path to determine the number of forward arrows required by the agent to reach a destination. For example, one student counted the number squares along a path to answer item CT5. The student included the agent’s start square so the written program was FFFF, whereas the correct program was FFF. This indicated that the student considered each grid square as the unit rather than the forward linear movement from the middle of one square to the next. These instances were also coded with the number knowledge code -Counting on, which will be discussed in the upcoming *How Students Demonstrate Number Knowledge* section.

Actions on objects indicating measurement knowledge. Units of measure knowledge emerged as a necessary component of students' spatial knowledge in codes when students enacted program codes with the assessment agent. This is because students must attend the program code's unit to correctly enact a program code. The spatial knowledge indicators described in the *Actions on Objects Indicating Spatial Knowledge* section in which students manipulated the assessment agent also describe measurement knowledge indicators because spatial knowledge in codes indicators also indicated units of measure knowledge when students enacted program codes (action on objects).

The distinction between unit measurement knowledge indicators and spatial knowledge in codes indicators is that unit measurement knowledge indicators occurred when students used assessment-specified units to enact programs, regardless of if the student attached the unit to a program code correctly. For example, assessment item CT5 asks students to build a program for the agent to reach a patch of grass located three squares east of the agent's start position. One student built the program RRF (correct program is FFF). Next, they enacted the program by moving the agent forward three distinct units. While this student demonstrated developing spatial knowledge by incorrectly enacting their built program, the student demonstrated measurement knowledge indicators by enacting each code with the assessment-designated forward unit.

Actions on objects indicators of developing measurement knowledge. Indicators of developing measurement knowledge are similar to those described in the *Actions on Objects Indicating Developing Spatial Knowledge* section for the same reasons described in the *Actions on Objects Indicating Measurement Knowledge* section.

Students indicated developing measurement knowledge by enacting single program codes as a combination of two or more unit movements, which I will refer to as compound units. For example, if the agent began facing west (CP4), some students enacted a R program code by compounding two right rotations (RR) to turn the agent 180° to face east.

One student used action on objects to move the agent while counting the total number of forward codes needed, but when asked how many total codes, the student repeated the process of moving the agent while counting each movement. In this way, the student indicated an understanding of the agent's units, but a developing understanding of distance by counting each unit rather than stating the total number of unit iterations required.

How Students Demonstrated Number Knowledge

Table 13 summarizes how students indicated number knowledge and developing knowledge through their gestures (e.g., "uses number words while tapping…"), language (e.g., "uses spatial language…"), and actions on objects (e.g., "physically adds or subtracts from a quantity of codes"). Notice that many of these indicators have overlapping modes for demonstrating knowledge (e.g., "Uses spatial language while enacting each associated movement"), especially the coordinating knowledges. Coordinating counts manifested in two ways; (1) when students coordinated counting with gestures or material use (coded as *Coordinating counts: Number*), and (2) when students moved the agent or gestured in a one-to-one relationship with program codes (coded as *Coordinating counts: Space*).

Table 13

Number Knowledge Indictors and Indicators of Development

MK	Indicators	Indicators of development
Counting	Counts objects or an agent's movements Names the correct number of codes or agent's movements Uses number words while tapping or pointing along on the grid or moving the agent	Mismatches counts with codes/objects/ movements
Counting on	Counts on from starting space when enacting a program or counting spaces Enacts the first program code by moving the agent forward one or more space Enacts the first program code by rotating the agent then moving the agent forward one space	Includes the starting square when enacting a program or counting spaces
Coordinating counts: Number	Connects the number of movements along a path with the number of program codes needed to build the path (through a verbal explanation or gestures) Connects the number of movements of the agent with number of codes (through a verbal explanation or gestures) Uses number words (e.g., three forwards) instead of spatial language (e.g., forward, forward, forward) for counting movements or codes	Does not coordinate counts with codes/objects/movements
Coordinating counts: Space	Uses spatial language (e.g., forward, forward, forward) while enacting each associated movement Enacts a program with one movement unit per code Connects each agent movement with an associated code (through a verbal explanation or gestures)	Does not coordinate movements with codes or objects
Operations	Physically adds or subtracts from a quantity of codes Verbalizes the need to add or subtract from a quantity of movements (e.g., "we need one more F")	

As shown in Table 13, there were more indicators of number knowledge than developing number knowledge. Similar to measurement knowledge, students indicated number knowledge most frequently with language and actions on objects than with gestures. The following sections will describe how students used indicated knowledge and developing knowledge through gesture, language, and actions on objects.

Gesture indicating number knowledge and developing number knowledge. The most common gesture students used to demonstrate number knowledge was tapping grid squares along an intended path while counting. This indicated students' counting and coordinating counts with number knowledge. By coordinating each tap with a counting number, the students indicated an understanding that each tap represented a discrete unit (coordinating counts: number knowledge) and that the collection of units could be quantitized by counting (counting knowledge). Students also used gestures to indicate developing knowledge of counting on, for example, students included the start square when they tapped and counted grid squares along an intended path. The tapping gesture allowed me to see that these students were still developing a knowledge of the distinction between counting on to identify units and counting discrete quantities.

Language indicating number knowledge. Students used number words, counting sequences, words indicating operations (which I termed operational language) as they enacted programs with the assessment items. Students also coordinated language with gesture, material use, and counts. *Operational language* indicated students' ability to apply addition knowledge to codes, movements, or spaces (e.g., "another one," "one more," "this and this," "three more"). For example, one student built a program by placing three F program codes on the program organizer saying "straight, straight, one more straight." By stating that the program needs "one more" code, the student indicated operations knowledge of adding. Another student used operational language to indicate repeated addition knowledge by tapping a F program code and stating "again, this one" before moving the agent.

Use of referential language (e.g., “here,” “there”; described in this chapter’s *Language Indicating Spatial Knowledge* section) and general/precise directional language (e.g., “straight,” “right,” “up”; described in this chapter’s *Language Indicating Spatial Knowledge* section) indicated coordinating counts with space knowledge when the students used language one-to-one while enacting a program and/or coding a program. For example, one student built the program FFF moved the agent forward one space while saying “one,” then added a F code to the program, then moved the agent forward another space saying “here,” then added a F code to the program. Finally, the student moved the agent one final forward movement while saying “then here” and placed the final F code on the program. This student demonstrated coordination between number, space, and materials. Additionally, the student indicated sequencing knowledge by using the word “then” between each movement.

Language indicators of developing number knowledge. Language indicators of developing knowledge occurred when students incorrectly quantitized codes, such as for item CT1 when students incorrectly stated the number of F program codes required for the agent to reach a destination (correct answer 2). When students stated that the agent required 3 F program codes, they indicated developing knowledge of counting on since 3 would have included the start square. Students who responded that the agent required 1 F program may have considered only the number of squares between the start and end locations rather than the number of unit movements required to land on the destination.

Actions on objects indicating number knowledge. Students demonstrated number knowledge of operations, counting on, and coordinating counts with space and number while engaging with the assessment's movable agent and program codes. Students indicated operations knowledge when they added or removed program codes to fix a program. One student demonstrated subtraction knowledge during item CT4 by sequencing all the available forward program codes on the program organizer to build FFFFF (correct answer FFF) then checked the robot's intended path and removed three forward codes leaving his answer FF. Other times, students used the agent to inform their program sequencing such as when they sequenced program codes then tested the program with the agent. These students demonstrated coordinating counts by associating one movement with each program code while enacting the agent. If the sequenced program sent the agent too far or not far enough, students adjusted the program accordingly by adding or removing a program code, most commonly a forward.

Students indicated counting on knowledge when they enacted a program's first forward code by moving the agent forward from the start space. For example, one student correctly enacted the intended path for assessment item CT4 by moving the agent FFF. To enact it again, she placed the agent back on the start space with a brief pause before enacting FFF. The brief pause seemed to "reset" the agent at the start from which she could then re-enact the intended path.

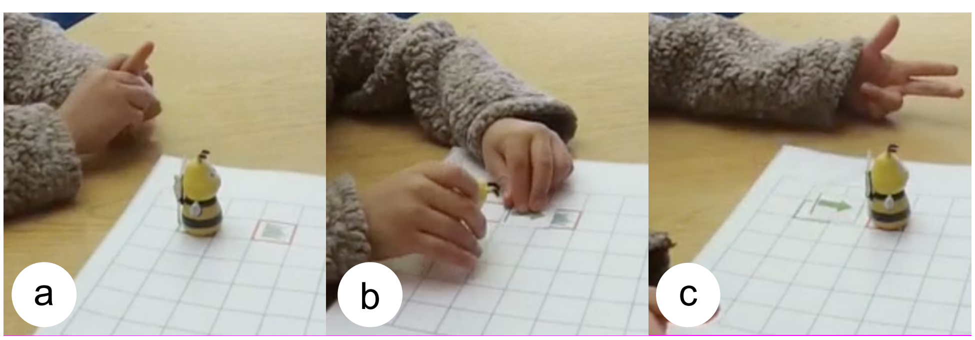
Actions on objects indicators of developing number knowledge. As shown in Table 13, a majority of developing number knowledge indicators emerged in students' actions on objects. One way that students' actions on objects to indicated developing number knowledge was when they did not coordinate actions or codes with movements. For example, one student built the program RRRRB for item CT9 (correct response

FFFRFFF). After building the program, she enacted it as FFRFFF. Although the program she built consisted of five codes, she enacted it with six movements. This indicated that she did not coordinate the number of codes in her program with the number of movements she enacted with the agent.

While the previous example highlighted an indicator of coordinating counts in space, other students indicated a developing knowledge of coordinating counts with number when they stated a quantity but modeled a different quantity without changing their answer. For example, one student who engaged with item CT1 stated that the agent required one “up,” or forward code, to reach the destination (correct answer FF; see Figure 17a). Next, the student (Figure 17b) placed a forward program code on the agent’s path between the start and end spaces and hopped the agent from the start space to the program code and then to the end space (Figure 17c) while making a noise for each movement.

Figure 17

Stating One Forward Needed but Models Three



Note. Student indicates the agent needs one forward code (a), places one forward program code on the agent’s path, (b) then hops the agent three times to the end of the path (c).

The student pictured in Figure 17 indicated knowledge of language/space coordination by coordinating each hop with a sound, however the student demonstrated a developing understanding of coordinating movements with program codes by stating and programing one forward arrow but enacting two forward commands.

How Students Demonstrated Computational Thinking

This section will describe how students' gesture, language, and actions on objects indicated their CT knowledge and developing knowledge. I operationalized CT knowledge by coding instances when students exhibited one or more of the early childhood CT components, which are algorithmic thinking, debugging, and decomposition. Algorithmic thinking was the most prevalent CT component and occurred almost exclusively when students built and read/enacted programs (see Table 9). Students indicated debugging knowledge when they recognized and fixed program errors and decomposition knowledge when they built wholes from parts and related parts to wholes. Possible wholes considered in the assessment include the agent's intended path and a sequenced program, while the parts included unit movements along a path and program codes. Table 14 summarizes the CT knowledge and developing knowledge indicators. Like the MK knowledge tables, Table 14's indicators were generalized from memos taken during the qualitative coding process. Table 14 follows a similar structure to the MK indicators and indicators of development tables (see Tables 10, 12, 13) and shows how students indicated knowledge of CT via the knowledge categories algorithmic thinking, debugging, and decomposition. The CT knowledge categories are listed in the table's first column. Category knowledge indicator codes for algorithmic thinking (*Plans program, Builds intended program, Read/enact program*) and debugging (*Recognize bug, Fix bug*)

Table 14

Computational Thinking Knowledge Indictors and Indicators of Development

CT	Indicators	Indicators of development
Algorithmic thinking		
Plan program	Gestures on grid, describes or indicates a path Uses terms such as starts, ends, travels, goes Taps codes to indicate order	Says, “It needs to go there" (pointing to destination) without indicating a path or sequence of steps Plans a path but may not show understanding of spatial ordering
Build an Intended Program	Builds programs left to right Uses language such as first, next, after, last Orders codes to accomplish a goal Applies codes correctly Builds/taps codes to show order	Builds an incorrect program Sequences a program’s correct codes out of order (e.g., “FFR” instead of “FRF”) Places codes without reason, indicating a developing understanding of how codes are sequenced or which codes to use
Read/enact Program	Enacts a program in the proper sequence Alternatingly taps a program code and enacts the code with the agent	Enacts program code(s) incorrectly (i.e., incorrect units, not from agent’s perspective) Enacts program incorrectly Enacts agent in a way that is not perceptibly related to the program codes
Debugging		
Recognize bug	Verbalizes or otherwise indicates that there is an error Stops an agent on an incorrect location and recognizes it as incorrect	Does not recognize an error in a student-built program Does not recognize an error in a presented program
Fix Bug	Fixes a bug by adjusting or rebuilding the program	Adjusts a program in a way that does not fix the error Uses a strategy that does not fix the error
Decomposition		
	Takes multiple codes at once and places them in program Codes and tests a program unit by unit Reuses a chunk of code (two or more codes) Enacts a portion of a program to code the next part Enacts a program then checks where the agent is in relation to goal; makes change when the program doesn’t work Describes a path or program with sequencing or spatial language Traces a path while describing it with sequencing or spatial language Describes how each code/chunk of code relate to the whole after building/enacting a program	Uses a F arrow to indicate continuous linear movement Identifies the whole as the number of path squares, not movements Places codes out of order or without reason, not indicating an understanding what the whole is or how codes contribute to the whole Enacts a program without reference to program codes Moves an agent twice but states that it only requires one code

are listed in the first column below each category. Although decomposition was originally broken into specific knowledge indicator codes, it was difficult to distinguish instances as primarily one form of decomposition or another. For this reason, general decomposition indicators are reported in Table 14 and are not attached to specific knowledge codes. Further research would be required to analyze instances of students' decomposition to disaggregate the forms of decomposition that students indicate.

As shown in Table 14, many of the algorithmic thinking indicators related to sequencing (e.g., "Builds/taps codes…," "places codes…", "orders…"). Considering also that algorithmic thinking was also the most frequently coded CT knowledge category (see Table 9), sequencing appears, transitively, to be an important factor in students' CT knowledge. Table 14 also shows that students used gesture primarily to indicate algorithmic thinking knowledge with one exception in decomposition ("Traces a path…").

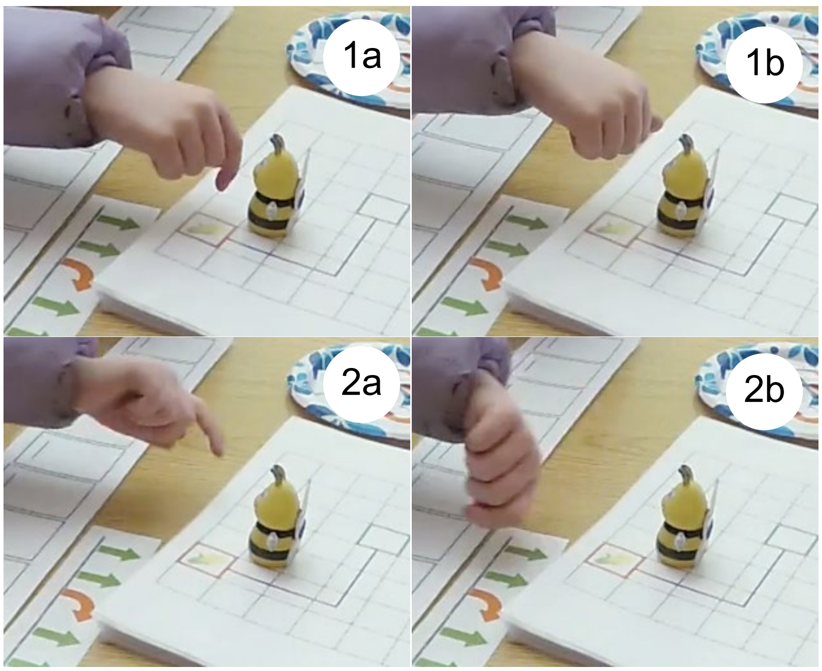
Students indicated CT knowledge and developing knowledge via all three modalities (gesture, language, actions on objects). Like students' MK knowledge, students' CT knowledge indicators overlapped among the three modalities, so each knowledge indicator is not exclusive to a specific modality. Operationalizing how students indicate CT knowledge via each modality, however, provides insight into how students embodied CT knowledge. For this reason, I will first describe how students indicated CT knowledge and developing knowledge through gestures and language. Next, I will describe how students indicated CT knowledge through actions on objects. To account for the higher instances of students indicating CT through actions on objects, the actions on objects section will be disaggregated by CT category to describe how students indicated algorithmic thinking, debugging, and decomposition knowledge and developing knowledge.

Gesture indicators of computational thinking knowledge. Overall, students gestured by tapping, sliding, and sweeping their fingers on paths and programs. These forms of gesture indicated algorithmic thinking knowledge when students tapped grid squares, slid their finger along the agent's intended path, tapped each code in a program before enacting the code, or while building a program. These knowledge behaviors related to decomposition when students used them to indicate portions of a path at a time. For example, in item CT9, students were given a program (FFRFFF, correct answer FFFRFFF) and asked to fix the program so that the agent follows a marked path to the destination. To solve the task, students' gestures indicated algorithmic thinking and decomposition knowledge by tapping or sweeping the squares along the first part of the path, then sequencing the corresponding program codes (FFF), before building the reminder of the program using various strategies. By gesturing along the first part of the path, students decomposed the path at the rotation and built the program's first string of linear movement (FFF) before tackling the remainder of the program (RFFF).

Gesture indicators of developing computational thinking knowledge. Gestures indicating developing knowledge were only observed when combined with language and/or actions on objects. For example, one student combined language and gesture in item CT9 to describe how the agent might reach the destination without following the indicated path. She stated that "if [the agent] wanted to go off the path he could go one more forward and one more right turn" while gesturing her finger south over the grid then east over the grid (see Figure 18).

Figure 18

Student Using Gesture and Language to Describe an Alternate Path



Note. Student suggests and gestures that if the agent were to reach the destination by going off the path it would "go one more forward" (1a, 1b) then make a "right turn" (2a, 2b).

The student in Figure 18 used a combination of gestures and language to indicate knowledge of precise programming language and the correct use of the forward code, however her sweeping gesture representing a right turn (pictured in 2a and 2b) indicated that she anticipated a right turn would direct the agent to rotate right and move forward one square. Her interpretation of a right turn indicated a developing understanding of program building. Hence, her gesture allowed me to see that her knowledge of program building was developing because her gesture indicated an incorrect application of the R program code.

Language indicators of computational thinking knowledge. I categorized the types of language students used into four categories: (1) process language (starts, ends, travels, goes), (2) sequencing language indicating algorithmic thinking (first, next, after, then, last), (3) language indicating decomposition (i.e., "two, because one, two," "three steps"), and (4) language indicating debugging (i.e., "that's not right," "this is wrong").

Process language (start, end, travels, goes) indicated knowledge of the program and path conventions such that the program and path have beginnings and ends and that the start and end are reached by the agent moving from one to the other. Process language also related to sequencing knowledge, as the sequenced programs and indicated paths contained a beginning and end.

Students used *sequencing language* while building and reading/enacting programs, both of which require the student to attend to the left-to-right sequencing convention dictated by the CT context. Sequencing language also indicated CT decomposition knowledge. Some students built programs code by code while stating the code name and separating each with the sequencing word "then," (i.e., "forward, then forward, then forward)." In this example, the sequencing word "then" separates each code from the next so that the student indicates an understanding that the whole (the program) is intentionally built by a collection of sequenced units (each program code).

When using *decomposition language*, students related how a whole program is composed of what I termed "chunks" of movements (a sequence of two or more units). For example, one student described why the L program code completes CT4's program

(FLFF). The student explained that the agent “moves” while enacting the first F code, “then turns this way” while rotating the agent L, and “then moves like this” while moving the agent FF. This student demonstrated decomposition knowledge by relating the units and chunks of the path (F, L, and FF) to the whole program and path.

Debugging knowledge language indicators arose when students correctly stated that a program contained an error, such as stating that there is something “wrong” or “not right” in a coded program.

Language indicators of developing computational thinking knowledge. Students indicated developing CT knowledge when they incorrectly stated that a program contained a bug, that a program directed the agent to a destination other than the actual destination or interchanged the terms right or left. For example, item CT10 required students identify the missing code in a program (given: F_FFLF; missing code: F). Students who responded correctly sometimes enacted the program to test it and misinterpret the L program code so that they incorrectly stated that the program contained a bug. Although these students responded correctly, their incorrect program code interpretation indicated a developing knowledge algorithmic thinking, and their incorrect assertion that the program needed to be debugged indicated a developing knowledge of debugging.

Actions on objects indicating computational thinking knowledge and developing knowledge. Actions on the assessment agent and program codes is required for assessment completion, so students frequently indicated CT knowledge and developing knowledge through their actions on objects. This section will discuss each CT knowledge domain (algorithmic thinking, debugging, and decomposition) and how the students indicated knowledge and developing knowledge through their actions with the assessment's agent and program codes.

Indicators of algorithmic thinking knowledge. Students indicated algorithmic thinking knowledge when they enacted a program correctly with the agent, correctly sequenced program codes, and attended to the agent's perspective. An example of students attending to the agent's perspective occurred in assessment item CT5. Assessment item CT5 presented the student with the agent facing east on the grid and asked the student to build a program for the agent to land on the "grass," a grid square picturing grass three spaces east of the agent (correct response FFF). Students who indicated algorithmic thinking knowledge in item CT5 recognized that the program code F—which appears to face north on the grid from the students' perspective—directs the agent to move forward according to the agent's orientation, not the cardinal direction of the program code's apparent north orientation. Students who indicated this algorithmic thinking knowledge responded to CT5 sequenced three F codes, indicating that they applied the F code to the agent's perspective.

Other students used the agent and program codes using program-building strategies which I call move-code and code-test. For the *move-code* strategy, students moved the agent one unit along the path then placed the movement's corresponding code in the program, repeating the move-code action until the program was complete. This actions-on-objects indicator showed students' algorithmic thinking knowledge of relating one program code to one movement to build a program unit by unit. By moving the agent first, the student indicated a unit movement along the path, then selecting and sequencing the associated program code revealed which code the student associated with the movement. Alternately, students would first add a code to the program then test the code by moving the agent. Students repeated this *code-test* program-building strategy until the program was complete.

In contrast to the move-code strategy, students indicated algorithmic thinking knowledge with the code-test strategy by selecting the anticipated code first, then testing the program code with the agent. This strategy allowed me to see students' interpretation of the code in the grid space. Students using this strategy would sometimes test the code and realize that the code did not have the intended outcome. Students would then reposition the agent and select a different code to test. In this way, the code-test strategy sometimes led to debugging knowledge indicators. These two different, but related, actions-on-objects strategies (move-code, code-test) were generally effective unit-by-unit programming strategies so long as students accurately interpreted the program codes.

Indicators of developing algorithmic thinking knowledge. Students indicated developing algorithmic thinking knowledge in various ways (see Table 14), however this section will highlight only two of these indicators: sequencing difficulties (i.e., losing track of what parts of the program had and had not been built), and not attending to the agent's perspective.

Sequencing difficulties emerged most often when students built longer programs (five or more codes) in items CT7 (FFLFF) and CT9 (FFFRFFF). In addition to being longer programs, items CT7 and CT9 each required a rotation mid-program. When programming CT7, students indicated developing knowledge by programming the first two forwards correctly and account for a rotation, but then incorrectly coded the final two F codes as one F. For CT9, students who debugged the program by rewriting it correctly built the length of the path up until the rotation (FFF), then appeared to have lost track of what they had already coded and either left out a rotation entirely or added an extra F, resulting in the program FFFFFF or FFFRFFFF. Students who indicated developing knowledge in this way built the program one code at a time, enacting each code with the agent before adding another code to the program.

A common indicator of developing algorithmic thinking knowledge was when students attended to the program codes' cardinal directions rather than applying the program code to the agent's perspective. To contrast an example given in the *Indicators of Algorithmic Thinking* section wherein students responded to item CT5 (correct response FFF) by building a program with F codes, students who indicated developing algorithmic thinking knowledge in CT5 built a program with R codes. The R program codes, from the student perspective, point west on the grid, which is the direction the agent needs to travel for item CT5. So instead of building FFF, students built the program RRR.

Indicators of debugging knowledge. Debugging knowledge indicated by students' actions on objects occurred when: (1) students correctly enacted a buggy program with the agent and recognized that the program was incorrect and (2) students successfully fixed a bug by sequencing program codes or adjusting a program's existing codes. Students indicated debugging knowledge even if they could not fix the bug. For example, one student successfully identified the bug in item CT9 (given: FFRFFF; debugged: FFFRFFF) by observing that the agent needed to go further forward before rotating. The student, however, did not know how to complete the program after the turn. Because the student correctly identified the error, the student indicated debugging knowledge. The student's inability to build the remainder of the program indicated developing algorithmic thinking knowledge as their difficulty was in coding the remainder of the program rather than debugging.

Indicators of developing debugging knowledge. Students indicated developing debugging knowledge primarily when they interacted with the program codes. This occurred when students adjusted a program by removing, replacing, or reorganizing a program's codes without fixing the bug or when students incorrectly identified a bug's location in a program and attempted to fix it by adjusting the codes.

Indicators of decomposition knowledge. Decomposition knowledge was indicated when students sequenced or enacted programs with chunks (two or more sequenced program codes). To build programs with chunks, students sometimes selected two or more program codes at a time before sequencing them in a program rather than selecting one code at a time and sequencing them individually. Other students enacted a program in chunks by referencing the program, enacting two or more program codes with the agent, then referencing the program again to enact the next program code individually or as part of the next chunk. For example, item CT3 asks students to complete a program by filling in the missing code (F_FF) for the agent to travel along a marked path (correct response F<u>L</u>FF). Some students indicated decomposition knowledge with the agent by enacting the program up to the missing code (F) before identifying which program code was missing. By enacting the given part of the program before selecting the missing program code, these students indicated an understanding that the program's whole can decomposed into parts. The parts in this instance were the program's beginning code (F), the missing code, and the remainder of the program. Some students who demonstrated this method for completing item CT3 finished enacting the program after placing the missing program code in the program while others did not. Students who enacted the remainder of the program used the actions-on-objects to indicate their knowledge of the program's final part and, in turn, the program's whole.

Indicators of developing decomposition knowledge. Students indicated developing decomposition knowledge when building and using wholes, which in this context include the path (by moving the agent) and the program (by manipulating the codes). Some students indicated a developing understanding of a program's whole by interacting with the number of squares along a path rather than interacting with the units of movement that the agent takes to move along the path. For instance, students responding to CT1 (correct response FF) sometimes stated that the agent required three F codes and enacted this by hopping the agent on the start, middle, and end square. These students indicated a developing understanding of the whole by decomposing it into incorrect units (each grid square) rather than correct units (units of linear movement).

Another interpretation of the whole was that it did not include the end space, only the squares between the start and end locations. These students regularly sequenced the program's codes so that the agent stopped one grid space shy of the intended destination.

In this way, the students demonstrated a different interpretation of the whole than was indicated by the assessment.

Mathematical Knowledge and Computational Thinking Co-Occurrences

In this section, I report on how students' MK and CT knowledge co-occurred as they completed CT assessment items. I ran a co-occurrence frequency report to determine which MK and CT codes co-occurred and the frequencies of each MK CT co-occurrence. According to the resulting co-occurrence report (see Appendix E), the most common MK codes to overlap with CT codes were *Spatial knowledge in codes* (SK), *Units of measure* (UnitsM), *Counting on*, and *Coordinating counts: Space* (CSpace). These MK codes most frequently co-occurred with the CT codes *Builds an intended algorithm* (Build) and *Read/enact program* (Read) (see Table 15). As indicated in Table 9, students demonstrated CT knowledge most frequently when they built and read/enacted programs, so it follows that the most frequent MK CT co-occurrences would occur with Build and Read.

Table 15 highlights MK and CT codes that co-occurred most frequently (referred to as MK CT codes) and the frequency of each co-occurrence. The first column lists the MK codes while the top row lists the CT codes. The intersections of each row (MK code) and column (CT code) quantify how many times the respective MK and CT codes co-occurred. Each MK CT co-occurrence represents one instance when a student indicated both knowledge types while completing an assessment item. For instance, Table 15 indicates that the MK code UnitsM and CT code Builds co-occurred 60 times. This means that of the 822 instances within the data set students enacted assessment items, 60 instances were coded with UnitsM and Builds.

Table 15

Frequencies of the Most Common MK CT Code Co-Occurrences

	CT codes			
MK Codes	Builds an intended algorithm	- BIA	Read/enact program	- REP
Spatial knowledge in codes	129	5	133	4
- SK	0	154	3	135
Units of measure	60	25	133	43
- UM	0	58	1	87
Counting on	60	71	135	115
- CO	0	2	0	3
Coordinating counts: Space	54	55	119	76
- CC:S	1	19	2	39

Table 15 reveals two main patterns. Firstly, students who indicated SK and UnitsM knowledge generally indicated Build and Read knowledge as well (SK/Build = 129, SK/Read = 133; UnitsM/Build = 60, UnitsM/Read = 133). Similarly, students who indicated developing SK (-SK) and Units (-UM) also indicated developing Build and Read knowledge (-SK/-BIA = 154, -SK/-REP = 135; -UM/-BIA = 58, -UM/-REP = 87). This pattern suggests that students' SK and UnitsM knowledge might relate to students' CT knowledge output, or that students' CT knowledge might relate to their MK knowledge.

Second, unlike the relationships between SK and UnitsM with Build and Read, CT knowledge output has an unclear relationship with students' Counting on and CSpace knowledge. A majority of Counting on and CSpace co-occurrences with -BIA and -REP were with indicated MK knowledge (Counting on, CSpace) rather than developing MK knowledge (-CO, -CC:S). This shows that students could demonstrate accurate CT knowledge and developing MK or could demonstrate developing CT knowledge even if they were using accurate MK. In total, indicated Counting on co-occurred with Build/-BIA and Read/-REP 381 times (60+71+135+115; see Table 15), compared with only five co-occurrences of -CO with Build and Read (0+2+0+3; Table 15). The same pattern emerged with CSpace, wherein students' CSpace knowledge co-occurred with Build/Read and -BIA/-REP. This pattern suggests that – in contrast with SK and UnitM's apparent relationship with students' CT knowledge output – Counting on and CSpace knowledge have an unclear relationship with students' CT knowledge output when building and reading/enacting programs.

While building and reading/enacting programs are different practices, they are similar in that students frequently engaged in both by manipulating the agent and program codes. Both practices also require students to use similar skills by interpreting program codes in a grid-based context. For these reasons, and to speak more broadly to CT, Build and Read will be discussed in tandem with the indicated MK co-occurrences.

To generalize the nature of these themes, I reviewed video clips of students exhibiting these MK CT co-occurrences and associated memos. Findings for each of the major MK CT co-occurrences will be discussed next: SK and Build/Read, UnitsM and Build/Read, Counting on and /Build/Read, and CSpace and Build/Read.

Spatial Knowledge in Codes with Building and Reading/Enacting Programs

One of the conditions in which students' SK knowledge co-occurred with Build and Read was when they used the program building strategies code-test and move-code. These strategies (discussed in the *Indicators of Algorithmic Thinking Knowledge* section) made students' program code interpretations visible as students paired agent movements with program codes. Students who correctly interpreted the program codes indicated SK knowledge and most frequently correctly built and read/enacted programs. Students who incorrectly or inconsistently interpreted the program codes indicated developing SK knowledge and most frequently built and read/enacted programs incorrectly.

Units of Measure with Building and Reading/ Enacting Programs

UnitsM frequently co-occurred with Build/Read and was observed when students enacted a program code with the agent. Students indicated UnitsM knowledge by moving the agent in unit movements specified by the assessment (see the *Actions on Objects Indicating Measurement Knowledge* section of this chapter). Assessors who allowed more time for students to build programs early on set the standard that students can build then test their programs, providing opportunities for students to demonstrate UnitsM knowledge. Assessors also elicited students' UnitsM knowledge when they reminded the student that students could move the agent if needed. Students also indicated UnitsM knowledge with -BIA and -REP, although with less frequency than with Build and Read knowledge. For example, a student might move the agent one unit at a time, but not correctly attend to the program codes' directionality. These types of occurrences might explain the frequency counts of UnitsM/-BIA (25) and UnitsM/-REP (42) in Table 15.

Counting On with Building and Reading/ Enacting Programs

Students exhibited Counting on when enacting the first program code with an agent. In indicating Counting on knowledge, students used the starting point on the grid as “zero” and counted on from the starting point. This occurred when students enacted, built, and tested programs. Counting on likely co-occurred frequently with Build and Read because students were instructed throughout the assessment to manipulate the agent in the grid space. All assessment items ask students to consider the agent’s movement in space, and all assessment items but two (CP3, CP4) require at least one forward movement from a starting point. Any time students enact a F program code from an agent’s start space they indicate knowledge, or developing knowledge, of Counting on.

Unlike SK and UnitsM, Counting on had an unclear relationship with students’ CT knowledge. Students demonstrated counting on knowledge 416 times in the data. Of the 416 instances, students indicated -CO only eight times (see Table 9). Of the 379 instances in which Counting on and -CO co-occurred with indicated and developing Read/Enact, only five co-occurred with -CO (see Table 15). The remaining 374 co-occurrences with Counting on were nearly evenly divided among Build/Read (N = 189) and -BIA/-REP (N = 185). This indicates an unclear relationship between students’ counting on knowledge and their CT knowledge output.

Coordinating Counts: Space and Building and Reading/Enacting Programs

Coordinating counts most frequently occurred when students built programs using the code-test and move-code strategies and when students coordinated language with each enacted movement. When using these strategies, students indicated their knowledge (or developing knowledge) of the one-to-one relationship between a movement in space and a program code or sound. There is an unclear relationship between students' CSpace knowledge and their program code interpretations. This may explain why there is not as drastic of a difference between the number of CSpace co-occurrences with CT's Build/Read and -BIA/-REP. Whereas SK is required for students to build programs, CSpace is not.

The CSpace and Build/Read co-occurrence is likely frequent because the assessment is performance-based, so students are instructed and expected to interact with the agent and program codes. The assessment's interactive design allowed students to demonstrate knowledge of coordinating relationships between program codes and an agent's movement in space while building and reading/enacting programs.

Like counting on, CSpace had an unclear relationship with students' CT knowledge output. As depicted in Table 15, students can indicate CSpace knowledge with Build/Read and -BIA/-REP. Co-occurrences with -CC:S knowledge, however, more often occur with -BIA/-REP, which indicates that – while there is an unclear relationship between indicated CSpace knowledge and CT knowledge output – a relationship exists between CSpace knowledge (-CC:S) and students' CT knowledge output. Additional investigation from a larger data set would be required to explore this further and verify the presence of such a relationship.

Deleted Codes in the Operationalization of Mathematical Knowledge and Mathematical Knowledge Computational Thinking Co-Occurrences

Chapter IV described the a priori coding-scheme used for this study. This section details why some of the a priori codes were dropped. The codes deleted from the analysis include *Spatial orientation*, *Abstraction*, *Decomposition (MK), Develop and apply algorithms*, *Patterning, Sequencing*, and *Spatial reasoning*. *Spatial orientation* knowledge was originally coded as part of this analysis; however, it was coded so frequently that it became superfluous, rendering its use white noise. Further research specifically exploring spatial knowledge would be useful to disaggregate how spatial orientation plays a part in other spatial knowledge categories. Other codes were also coded excessively and subsequently dropped as they appeared as overarching concepts to other codes that described the ideas more specifically (see Table 16). Table 16's first column lists a priori codes dropped because they were represented more specifically by other related codes. The related codes for each of the dropped a priori codes is listed in the table's second column.

For example, Table 16 indicates that the original a priori code *Develop and apply algorithms* was dropped since other codes (*Builds intended algorithm*, *Read/enact program*) described coded instances more specifically. As another example, abstraction was originally coded every time students abstracted meaning from a code by reading/enacting a program, stating a code's name or function, or building a program. Abstraction was also applied when students abstracted quantities by counting or using

Table 16

A Priori Codes Represented More Specifically by Other Codes

Original code	Related codes used instead
Abstraction	Building intended algorithm, Read/enact program, Counting, Operations, Distance, Coordinating counts: Space, Coordinating counts: Number
Develop and apply algorithms	Builds intended algorithm, Read/enact program
Sequencing	Builds intended algorithm, Read/enact program, Counting, Coordinating counts: Number, Coordinating counts: Space
Spatial reasoning	Spatial visualization, Spatial language, Spatial knowledge in codes

number words. In this way, while abstraction appeared as a salient aspect of students' engagement with the assessment, other codes were better at dissecting how students abstracted in the assessment.

Patterning and Decomposition as MK were also original a priori codes. *Patterning* was dropped since the assessment did not assess the coding system's underlying structure in a way that was visible in students' responses. *Decomposition* as MK was dropped since it was difficult to distinguish students' decomposition strategies as evidence of MK or CT knowledge. Additional analysis would be required to further distinguish how students indicate forms of MK and CT decomposition knowledge.

Summary of Research Question 1's Results

The results for Research Question 1 reveal that students demonstrate MK and CT knowledge and developing knowledge multi-modally through gestures, language, and actions on objects. Operationalizing MK and CT revealed that indicators of knowledge and developing knowledge provide insights into the mathematical skills and thinking that students engage with during a CT assessment. Indicated and developing knowledge were not exclusive, meaning that students sometimes indicated MK or CT knowledge and developing knowledge at the same time (e.g., *Units measurement* and *-D*). Results for Research Question 1 also revealed that the MK spatial knowledge, units, counting on, and coordinating counts with space most frequently co-occurred with the CT knowledge build and read/enact programs. Further, this section reported that some MK (SK, UnitsM) might relate to students' CT knowledge output, while other MK have an unclear relationship with CT knowledge output (Counting on, CSpace).

Results for Research Question 2

In this section, I present the results for Research Question 2, *How do students' MK and MK CT co-occurrences relate to their performance on individual CT assessment items?* To answer this question, I provide descriptive case studies of four assessment items describing how MK and MK CT co-occurrences might relate to assessment item performance. Question 1's analysis and the co-occurrence models informed my final case study selection. Case studies will be presented for assessment items CT1, CT2, CT5, and CT6. These cases were selected based on the mathematics made visible in students' enaction of each assessment item. For each case study I first describe the assessment item and why it was selected as a case, then how an MK or MK CT co-occurrence manifested in students' performance of the item and how the MK or MK CT co-occurrence might relate to students' performance of the item. Following the four case studies, I draw cross-case conclusions to generalize how students' MK and MK CT co-occurrences might relate to students' performance on individual CT assessment items.

The images in this section include case models that are described in Chapter III's *Phase 3 Analysis* section. Each case model was generated from code frequencies in MAXQDA to represent the MK and MK CT co-occurrences that students exhibited when they engaged with each CT assessment item (see Figure 19). Each case model includes the MK and MK CT co-occurrence codes that occurred three or more times in each assessment item.

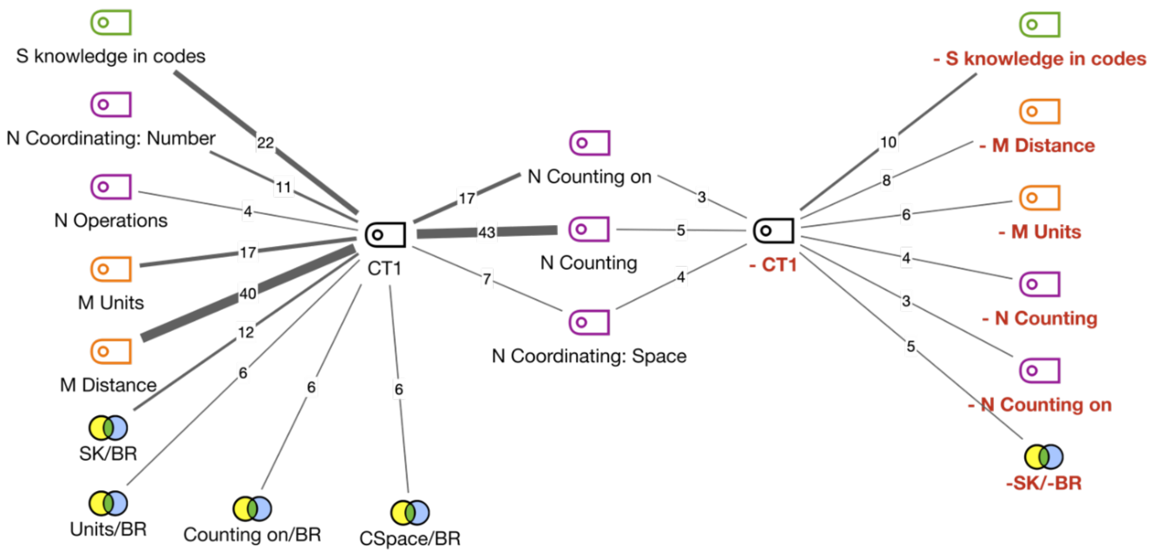
For example, Figure 19 depicts the visual case model for assessment item CT1. The model's two central points are CT1 and -CT1, which represent instances when students responded to CT1 correctly (labeled CT1) and incorrectly (-CT1). The nodes extending from CT1 and -CT1 connect with MK and MK CT indicated by students who answered correctly (CT1) and incorrectly (-CT1). For example, the top left node extending from CT1 is "S knowledge in codes" and indicates the code's frequency (22). The described node indicates that of all the instances that students responded to CT1 correctly, 22 of the instances indicated S knowledge in codes. The N Counting on code in Figure CT1 (center, top), however, has two nodes connecting it to both CT1 and -CT1. The nodes connecting N Counting with CT1 and -CT1 show that there 17 instances wherein students used Counting on and answered correctly while in three instances students indicated Counting on answered incorrectly. Four CT assessment item case models are depicted in this section with the remaining models available in Appendix D.

Assessment Item CT1 Case Study: Distance Measurement

The grid on the left of Figure 20 depicts the CT1 assessment grid. The grid square

Figure 19

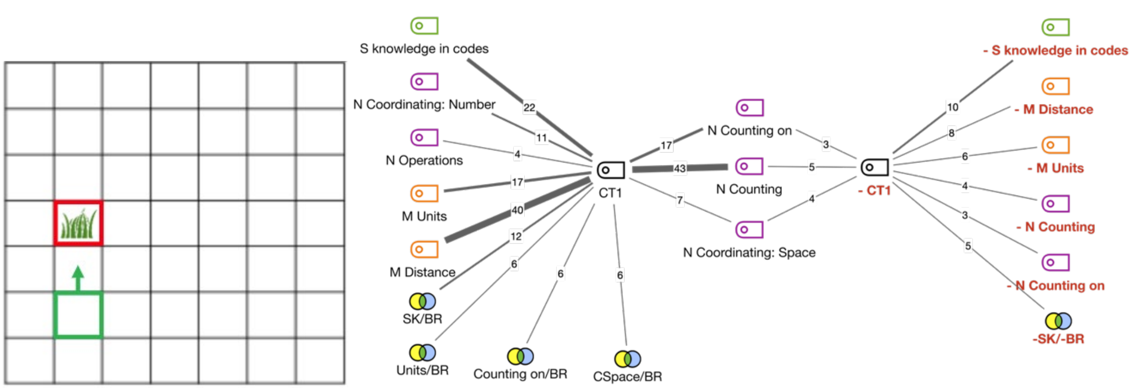
Case Model Example



Note. The case model icons resembling Venn Diagrams (colored yellow, green, and blue) indicate MK CT co-occurrences and all other icons except those labeled as CT1 and -CT1 indicate MK codes. The abbreviations used in this diagram are as follows: S (Spatial; colored green), N (Number; colored purple), M (Measurement; colored orange), S knowledge in codes (SK), Counting on: Space (CSpace), BR (Build and read/enact programs). The labels with red text indicate developing-knowledge codes (i.e., -M Distance) and an incorrect response (-CT1).

Figure 20

CT1 MK/MK CT Case Model and Assessment Materials



outlined in green indicates the start location with the agent starting orientation marked with a green arrow. The end/destination square is outlined in red with a picture of grass. To administer CT1, facilitators placed the agent on the starting square facing north and asked students how many forward arrows the agent would need to reach the destination (correct response “2” or “FF”). This item assessed students’ algorithmic thinking, in particular their knowledge of codes and planning codes for a program. Students could use the agent or program codes to respond, but using the materials was not necessary.

The case model on the right of Figure 20 represents students’ CT1 responses and corresponding codes. Item CT1 was unique in that it elicited students’ demonstration of distance measurement knowledge (M Distance in Figure 20), a knowledge that was uncommonly indicated by students in other assessment items. All but one instance of Distance knowledge and all -D occurred during CT1, so this assessment item was selected as a case study. Of the 60 students who responded to this item, 47 responded correctly (78%) and 13 responded incorrectly (22%). Students who responded to CT1 correctly indicated distance knowledge 40 times (see Figure 20) by stating that the agent required two forward codes, whereas students who responded incorrectly to CT1 indicated -D knowledge 8 times. In instances of -D, students stated the incorrect distance (one or three). Two students who answered incorrectly stated that the agent required one, but then moved the agent forward twice to reach the destination.

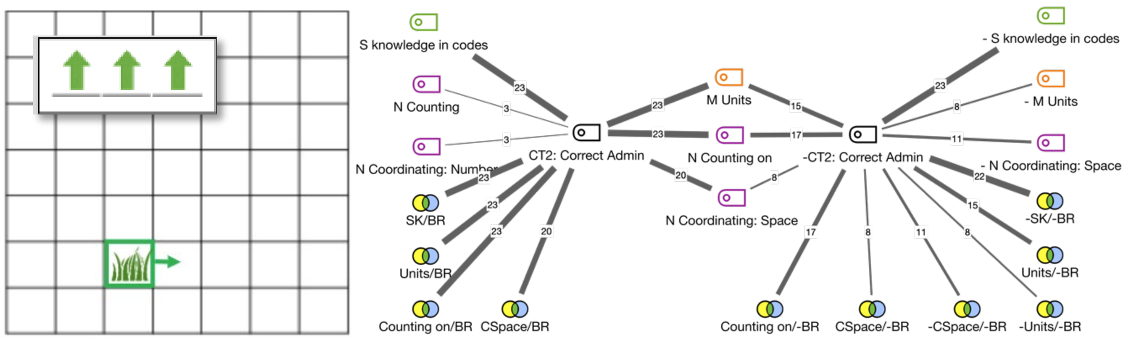
Students who indicated distance knowledge identified the total number of units (i.e., two F program codes) the agent needed to travel from a start to an endpoint. This necessitated that students consider how many linear unit iterations were required in the space and state the total number of iterations, or distance. By stating the distance as a quantity of forward units, the students also indicated spatial visualization knowledge, as they would have had to visualize how many unit iterations. Students who answered “one” but moved the agent twice indicated that they understood how the agent moved in unit movements. However, they did not yet connect this knowledge with the concept of distance being a sum of the unit iterations required to reach the destination. In this way, distance measurement knowledge related to students’ performance on item CT1, which assessed students’ algorithmic thinking.

Assessment Item CT2 Case Study: Units of Measure

For item CT2 (see Figure 21), the assessor positioned the agent on the grass facing east, gave the students a program strip with the program FFF, and instructed the student to move the agent according to the program. This item assessed students’ algorithmic thinking, in particular, their reading and enacting of a given program. Item CT2 was the only assessment item administered a portion of the time (12 of 60 instances) incorrectly. When administered incorrectly, the assessor incorrectly positioned the agent in a north-facing direction rather than an east-facing direction. The purpose of the east-facing start orientation was to assess students’ knowledge of forward codes when the agent was oriented in a direction that did not match the students’ orientation (i.e., agent’s east-facing versus the student’s north-facing), which involves the CT skill of interpreting codes as well as the MK knowledge of spatial orientation. Item CT2 was selected as a case study to understand better the assertation made in this chapter’s co-occurrence section that UnitsM knowledge might relate to students’ CT knowledge output in this assessment context.

Figure 21

CT2 MK/MK CT Case Model and Assessment Materials



Additionally, UnitsM knowledge (M Units in Figure 21) was frequently coded in item CT2 as 46 of the 48 students who responded to this assessment item indicated UnitsM knowledge or developing knowledge. The case model on the right of Figure 21 represents the 48 times item CT2 was administered correctly. Of the 48 students, 23 students responded to the item correctly (48%) and 25 students responded incorrectly (52%).

As represented in the case model in Figure 21, all students who responded to CT2 correctly (*N* = 23) indicated UnitsM knowledge (M Units), SK (S knowledge in codes), and counting on (N Counting on). This means that UnitsM co-occurred with the MK codes for SK and Counting on. The students with accurate responses to CT2 also indicated coordinating counts knowledge in either space (*N* = 20; N Coordinating: Space) or number (*N* = 3; N Coordinating: Number). Likewise, the MK CT co-occurrence codes (shown as Venn Diagrams) indicate MK co-occurrences with the CT knowledge

Build/Read (SK/BR, UnitsM/BR, Counting on/BR, and CSpace/BR). This means that the students who indicated units, space, counting on, and coordinating counts with number knowledge also indicated CT building and reading knowledge. Students who responded to CT2 manifested UnitsM when they moved the agent according to the assessment-established units. Correct units included linear movements from the center of one square to another (for F and B codes) and 90-degree rotational movements in place (for R and L codes). I coded UnitsM regardless of whether the demonstrated unit coincided correctly with an enacted program code's meaning. For example, when students enacted CT2's given program (FFF) as LFF, I coded the instance as UnitsM and -SK. Coding UnitsM knowledge separate from spatial knowledge in codes allowed me to see students' unit and spatial knowledge interplay. As described earlier, spatial orientation was dropped as a code; however, CT2 provides an example of how it became difficult to account for students' unit knowledge without understanding how their spatial orientation knowledge affected their program code interpretation. It was difficult to identify whether students enacted their interpretation of program codes were simply reorienting the agent to face the F program code's cardinal direction. This difficulty brings to question how (and if) students' spatial orientation knowledge affects their UnitsM knowledge application, if their UnitsM knowledge affects their spatial orientation knowledge, or if there is no relationship between the two and they are co-occurring knowledge.

Of the 23 students represented in the node connecting -CT2 with -SK, 15 indicated UnitsM knowledge (M Units) and 8 indicated developing UnitsM knowledge (-M Units). This suggests that students may have developed unit knowledge but that their spatial knowledge is still developing.

The students who indicated developing UnitsM knowledge (-M Units) used a combination of correct and imprecise linear unit lengths (1.5 squares, two or more squares), rotated the agent 180 degrees, or counted a hop on the start square as a forward unit length. A lack of unit length precision caused the students to end the agent on the wrong square so that the intended and actual end points did not match. Hence, it appears that students' unit length precision was important for correctly responding to item CT2.

As shown in the model in Figure 21, 15 students indicated UnitsM knowledge but answered item CT2 incorrectly. Students who answered incorrectly most often rotated the agent left 90 degrees to face north, matching the program arrow's cardinal-facing direction. After rotating the agent, most students moved the agent forward twice for a completed enaction of LFF or three times for a completed enaction of LFFF. Students who moved the agent LFF indicated three units of movement, the same number of movements indicated by the given program (FFF), suggesting that these students interpreted the program code F two different ways. The students interpreted the first F as a left rotation to face north and the final two F codes as F. In other words, the students who enacted LFF maintained a one-to-one coordinating relationship between the program codes and agent movements, whereas those who enacted LFFF did not. Students who enacted LFF indicated a coordination knowledge of program code and movement, which is a contextual CT and mathematical coordination skill.

However, students who enacted CT2 as LFFF used four-unit movements to enact a three-code program. Students who enacted LFFF did not indicate the same coordination knowledge, though it is unclear if these students:

- interpreted the first F code as a left-forward movement,
- did not consider the first rotation unit part of the program enactment and reorientated the agent, or
- reoriented the agent for another reason, such as thinking that the administrator incorrectly placed the agent.

If students interpreted the first F program code as a left-forward movement, this would indicate that they interpreted the F program code in more than one way and that their unit knowledge is developing. If a student did not consider the first rotation unit part of the program enactment, then the student likely oriented the agent north to face the F program code's cardinal direction. The student may have unintentionally applied a rotation unit to reposition the agent in this case. A similar conclusion might be drawn for the third LFFF rotation interpretation that the administrator incorrectly placed the agent. By rotating the agent to face north before enacting the F program codes, students attended to the cardinal direction of the program arrows rather than applying program codes to the agent's perspective. Conversely, one student demonstrated his ability to apply program codes to the agent's orientation by rotating the program strip 90-degres right so that the F program codes matched the agent's east-facing perspective. When students demonstrated this type of action on the objects, it showed their ability to translate the program codes to the agent's spatial orientation. Translating program codes according to the agent's orientation is another MK spatial skill that might relate to students' CT item performance and will be discussed further in a case study of item CT5.

Two students enacted FFF by keeping the agent in an east-facing orientation and moving the agent north on the grid three units, essentially side-stepping the agent FFF.

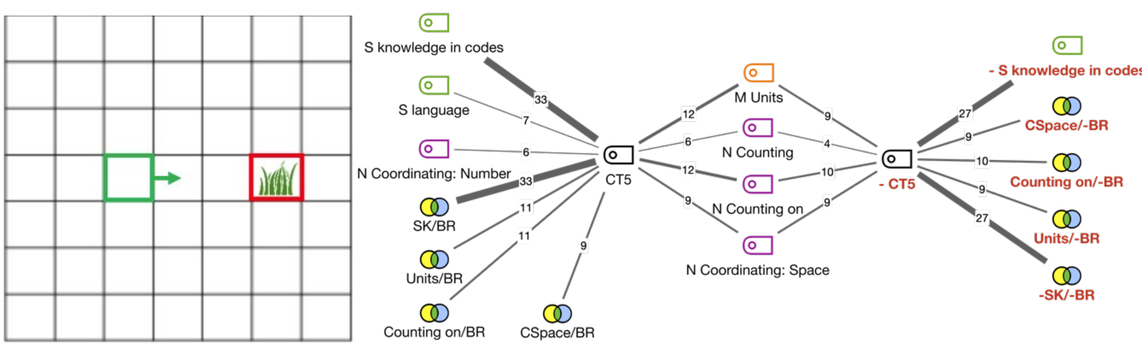
While these students indicated unit knowledge, they did not attend to the agent's orientation when applying the units, so their responses were incorrect.

Assessment Item CT5 Case Study: Spatial Knowledge in Codes with Build and Read/ Enact Program

Assessment item CT5 asked students to build a program. The assessor placed the agent on the start square facing east on the assessment grid and asked the student to build a program so that the agent would land on the grass (see the grid in Figure 22). Students who responded correctly built the program FFF. Item CT5 differed from CT2 in that students built the program with codes rather than read and enact a given program. Of the 60 students who responded to CT5, 33 answered correctly (55%) and 27 answered incorrectly (45%). This item was selected as a case study to understand better the co-occurrences between spatial knowledge in codes and building and reading/enacting programs.

Figure 22

CT5 MK/MK CT Case Model and Assessment Materials



As shown in the CT5 case model (see Figure 22), all students who responded correctly to CT5 indicated co-occurring spatial knowledge and building/reading knowledge (SK/BR; *N* = 33), while all students who responded incorrectly indicated -SK/-BR (*N* = 27). Of the students who responded incorrectly, 13 built the program RRR and four constructed other three-code programs (FFB, FRF, RRF, LLL) with the program codes positioned to face the agent's direction of travel. All but three of these students enacted their written programs, and those that enacted their program enacted it as FFF. Although none of these students built the correct program, they indicated counting on (N Counting on), coordinating counts in space (N Coordinating: Space), and units (M Units) knowledge when enacting their programs. By enacting the programs, these students also indicated their interpretations of each program code. The commonality of these students' programs was that all the program codes were positioned so that they pointed in the agent's direction of travel. These students showed that they were still developing knowledge of spatial orientation because they were not yet able to spatially apply a program code's meaning to the agent's perspective. This developing spatial orientation knowledge impacted their building and reading/enacting CT performance.

Conversely, three students who responded correctly built the program FFF but positioned the F program codes to face the agent's direction of travel. By orienting the program codes to face the direction of travel, these students indicated an understanding of the program codes' meaning applied to the agent's spatial orientation, but a developing CT understanding that the agent would enact the program code regardless of the code's orientation.

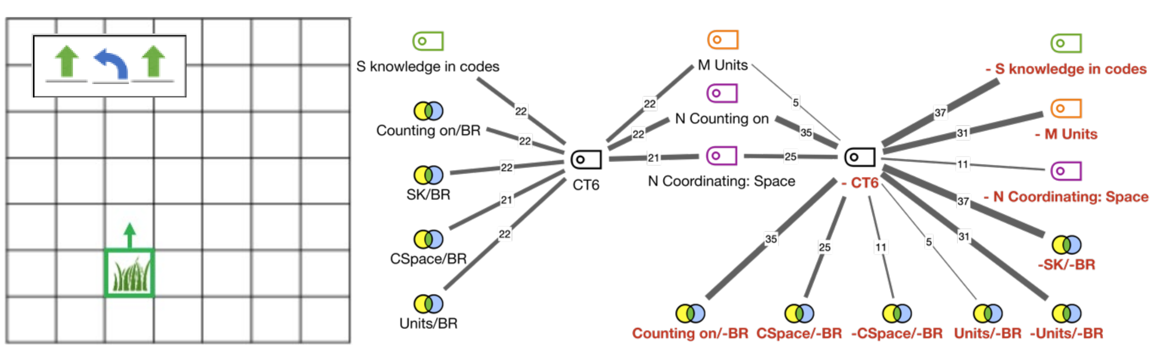
Assessment Item CT6 Case Study: Counting On

Assessment item CT6 asked students to read and enact the program FLF. The assessor placed the agent on the start square facing north on the assessment grid and asked the student to move the agent according to the given program (see the grid in Figure 23). Of the 59 students who responded to CT6, 22 answered correctly (37%). This item meant to assess students' ability to read and enact a program with a rotation, so the agent's beginning orientation matched the student's orientation. This item was selected as a case study to examine the unclear relationship between counting on knowledge and students' correct and incorrect responses to this item.

Students indicated counting on knowledge when they moved the agent away from the start space to enact the first code. The case model in Figure 23 shows that all students who responded to CT6 correctly indicated counting on knowledge (N Counting on; *N* = 22), as did 35 students who responded incorrectly. This means that, although counting on emerged as an accessible knowledge type for the students, it has an unclear relationship with students' CT knowledge. This is to say that counting on was necessary for students

Figure 23

CT6 MK/MK CT Case Model and Assessment Materials



to accurately respond to this item (as indicated by its use by all students who responded correctly), it is unclear if counting on relates to CT knowledge output. For example, students who responded incorrectly to CT6 counted on correctly, but often made an error when enacting the L program code by enacting it as a compound LF movement. In this way, students enacted the F program code correctly, but the L program code incorrectly for an incorrect response. The unclear relationship between students' counting on knowledge and their assessment responses reflects the findings reported in the co-occurrence section of this chapter that reported the co-occurrence of counting on and building/reading programs (Counting on/BR) as having an unclear relationship with students' CT assessment output.

Cross-Case Conclusions of Case Studies for Items CT1, CT2, CT5, and CT6

The case studies presented in this section reported on how students used types of measurement (case study CT1, CT2), co-occurring spatial building/reading programs (CT5), and number (CT6) knowledge to indicate CT knowledge and how these MK might relate to students' assessment performance. The models and analysis showed that some MK and MK CT co-occurring knowledge might relate to students' CT knowledge. In contrast, other MK and MK CT co-occurring knowledge suggest an unclear relationship with students' CT knowledge. The MK and MK CT co-occurring knowledge that might relate to students' CT knowledge is present when students correctly respond to a CT assessment item. Case studies CT1 and CT5 illustrated this relationship. Distance measurement knowledge was necessary for correctly responding with a total number of unit iterations in CT1. As shown in case study CT5, the co-occurring knowledge SK with Build/Read was likewise present when students correctly built the FFF program when the agent's orientation did not match the student's orientation.

On the other hand, case studies CT2 and CT6 described how other knowledge types have an unclear relationship with students' CT knowledge output. These knowledge types appear necessary for students to respond correctly to CT assessment items; however, students' CT knowledge is not contingent upon these types of MK knowledge. In these cases, the MK was important but already well developed (i.e., not co-occurring with the CT knowledge) or not as intricately co-occurring with the CT knowledge as knowledge types that might relate to students' CT knowledge. For instance, in CT6, counting on was a well-developed skill among the students, and students responded incorrectly to CT6 for reasons other than counting on knowledge.

The case models and analysis also showed interconnections between and within MK and CT knowledge, the most prominent interconnections being (1) units of measure knowledge as foundational to successful coding, and (2) spatial concepts as foundational to reading, interpreting, and enacting program codes. These themes emerged across all case studies, as shown in Table 17.

Table 17 cites evidence from the case studies supporting this chapter's themes. The table's first column lists the case studies. The second and third columns list the evidence presented in each case that supports each corresponding theme. For example, one of the supporting evidences of *Units as a Foundation to Early Coding* from case study CT5 is that "Most students wrote a three-code program, which reflects the required

Table 17

Case Study Evidence Supporting Overarching Themes

Case study	Themes: Units as foundational to early coding	Themes: Spatial concepts in program codes and agent enactments
CT1	Units of measure knowledge necessary to identify distance; Enacted units correctly when counting on from the start; Units represented parts of the whole (distance); Some indicated developing knowledge of distance as the sum of unit iterations	Visualized units to state total distance without enacting
CT2	Units co-occurred with spatial knowledge in codes, coordinating counts, and counting on; Unit length precision was necessary for correct responses; Some students attached unit enactment one-to-one with program codes, others enacted with more units than program codes	Attended to program code's cardinal direction rather than agent orientation; Did not attend to agent's perspective when side-stepping agent; Students' spatial orientation knowledge was not immediately visible, making it difficult to draw decisive conclusions
CT5	Most students wrote a three-code program, which reflects the required number of units (FFF); Most students enacted the correct units regardless of the program that they wrote	Spatial knowledge in codes was necessary to correct program enactment; Students positioned program codes to face the agent's direction of travel; Coordinating spatial knowledge in codes with agent's east-facing perspective was more difficult than when the agent faced north
CT6	Enacted rotation incorrectly as a compound unit (L and F units)	Spatial knowledge in codes (a dependent knowledge type) impacted students' response accuracy rather than counting on knowledge

number of units." One piece of evidence from CT5 of the theme *Spatial Concepts in Program Codes and Agent Enactments* is that "Students positioned program codes to face the agent's direction of travel."

While distance measurement was the focus of the item CT1 case study and units the focus of the item CT2 case study, the concept of a unit of measure was implicit and foundational in this analysis. It also appeared important for understanding codes in item

CT5 and as an underlying concept in counting on to successfully enact a program in CT6. Spatial concepts as an important interconnection were not surprising due to the assessment's three-dimensional design (i.e., the grid, agent movements), but the nature of spatial knowledge–in particular, representing space in the form of an arrow code and understanding one's orientation in relation to the robot and the start/stop points on the grid–was salient across the case studies. Spatial knowledge overlapped within other MK, specifically measurement units and counting on, because of the spatial nature of these mathematics concepts in this grid- and movement-oriented context. The particulars of spatial orientation emerged as a key idea in analyzing the case studies and, although not coded for in this study, will be discussed further as an area for future research in Chapter V.

Summary of Research Question 2 Results

The results for research question 2 indicate that MK and MK CT co-occurrences primarily occur when students build and read/enact programs. The MK and MK CT co-occurrences were identified as relating to or having an unclear relationship with students' CT knowledge output, or assessment item performance. Additionally, a cross-case analysis revealed that the connections between MK and CT were more prominent in the ways that units of measure knowledge and spatial knowledge might relate to CT outcomes.

Summary

The results reported in this chapter operationalize how kindergarten students indicated MK and CT knowledge and developing knowledge during a CT assessment through multi-modal means, more specifically via their gestures, language, and actions on objects. This chapter reports on specific indicators that students exhibited to indicate their knowledge and developing knowledge. These results reveal that students indicate various knowledge types via the three modalities and that some knowledge category indicators were more prevalent in certain modalities than others.

The results also indicate that meaningful MK CT co-occurrences exist, primarily MK with building and reading/enacting programs. Further, these co-occurrences revealed patterns that suggest that certain MK might relate to students' CT assessment item performance and other MK has an unclear relationship with students' CT assessment item performance. The MK that might relate to students' CT assessment item performance emerge as indicated knowledge when students respond correctly to assessment items (Spatial knowledge in codes, Units of measure). MK with an unclear relationship with CT assessment item performance may be present as indicated or developing knowledge whether a student responds correctly or incorrectly to a CT assessment item.

Finally, the case studies presented in this chapter generalize specific MK and MK CT co-occurrences that might relate to students' CT assessment performance on specific assessment items. These case studies also highlighted specifically how measurement, space, and number might relate to students' item responses. Two themes emerged across the case studies: (1) units of measure knowledge as foundational to successful coding, and (2) spatial concepts as foundational to reading, interpreting, and enacting program codes.

CHAPTER V

DISCUSSION

The purpose of this study was to operationalize how kindergarten students indicated MK and CT while engaging with CT assessment items, to understand how students' MK and CT co-occur, and to describe how students' MK and MK CT co-occurrences might relate to their performance on individual CT assessment items. Results showed that students used gesture, language, and actions with the assessment materials to indicate measurement, spatial, and number knowledge while solving CT assessment items and that specific MK might relate to students' CT item responses. This study's results also indicate that deep interconnections exist between (and within) MK and CT knowledge. Two primary interconnections (referred to hereafter as themes) arose as particularly prominent in students' engagement with the CT assessment and their assessment responses. These themes include (1) units as foundational to early coding and (2) spatial concepts in program codes and agent enactment.

This chapter is organized into five sections. In the first two sections, I will discuss the themes within the context of related literature and describe recommendations for future research. The final three sections of this chapter contain this study's implications, limitations, and a subsequent conclusion.

Units as Foundational to Early Coding

One important theme from this study's results is that units of measure knowledge is foundational to early coding skills. This theme was made particularly apparent by this study’s identification of units of measure knowledge as having as having a possible relationship with students’ CT knowledge output. Stated otherwise, unit knowledge was related to students’ accurate CT assessment item responses. This means that in this assessment context, students’ knowledge of the agent’s units of movement through space appears to be related to their performance on – and engagement with – the assessment. Based on this study’s results, students’ units of measure knowledge in this CT context necessitated three distinct understandings.

- Linear and rotational units are distinct with fixed magnitudes.
- Units are continuous and dynamic in nature.
- Units are iterated.

For example, correctly enacting the program for assessment item CT5 (forward, rotate left, forward; FLF) necessitated the knowledge that (1) the linear unit length of one forward program code and the rotational magnitude of one L program code, (2) each unit is a dynamic continuous movement rather than a discrete quantity, and (3) each unit is iterated by enacting one after the other continuing from where the agent began and where the previous unit left off. This section will discuss this theme by describing these distinct understandings within the context of related literature.

Linear and Rotation Units Are Unique and Distinct with Fixed Magnitudes

A foundational element of measurement knowledge is that units are identical copies of a measurable attribute used to measure the same attribute of a larger object (Clements & Sarama, 2021; Smith & Barrett, 2019). This foundation of measurement knowledge translates to this study’s context via length and angle units. Length and angle measurement knowledge particularly apply to this study's context as the assessment's program code units are linear (forward, *F*; backward, *B*) and rotational (rotate right, *R*; rotate left, *L*) within a three-dimensional grid space.

This study's joint embodied and enactivist perspectives allowed me to observe how students' gestures and actions on objects indicated their existing knowledge and knowledge-in-development of linear and rotational units. Students in this study indicated rotational knowledge when they enacted rotations with the agent and gestured with their hands. Clements et al. (1996) similarly found that students used embodiment to make sense of rotations by rotating their trunks to face different directions to assign a quantity to a rotation. Relatedly, Shumway et al. (2021) found in a study of kindergarteners engaging with tangible robot toys that students made sense of rotations by physically rotating the robot (actions on objects) and embodying rotations via body movement (i.e., rotating their heads and bodies) and hand gestures.

One of the difficulties students encountered when making sense of the assessment's program codes to build and enact programs was interpreting the left (L) and right (R) rotation program codes. Students in this study attended to length units more intuitively than rotational units, which Clements et al. (1996) also found in a case study examining four 9-year-old students engaging with digital programming activities. Students in the present study indicated developing knowledge of rotational units when they enacted the R and L program codes as linear units with varying rotational magnitudes (45, 90, or 180 degrees) and as compound rotation+linear movement units. By enacting the R and L program codes with differing magnitudes, students indicated their developing understanding of the coding system and a burgeoning understanding of the rotational unit's fixed magnitude and that linear and rotational units are distinct units of movement and unique attributes. While U.S. students do not precisely measure angles until fourth grade (CCSSI, 2010), kindergarten students are expected to compare measurable attributes (CCSSI, 2010) and can implicitly use angle concepts (Clements & Sarama, 2021). Understanding that length and rotation are unique attributes could support students' ability to differentiate linear and rotational codes from one another. This means that students' varying precision with linear and rotation units relates to their performance on CT assessment items. This early unit measurement knowledge overlapped with CT when students used actions on objects to build programs with codes and enacted programs with the agent.

Units are Continuous and Dynamic

Another foundational understanding of units that is necessary for students to respond correctly to the assessment items within this context was that the program code units were continuous and dynamic; continuous meaning that the units can be infinitely divided (Thompson & Carlson, 2017), and dynamic in that each unit movement is enacted rather than existing as a static representation of magnitude (Welch et al., 2022). Understanding that the linear and rotational units represented by program codes are continuous *and* dynamic was necessary for students to build and read/enact programs accurately. For example, students indicated developing knowledge of the program code's units' continuous properties when they counted discrete grid squares to plan and build programs. This strategy was unsuccessful when students included the starting square in their count (indicating developing counting on knowledge) or when a program required a rotation, as a rotation unit was to be enacted in place without the agent traveling to another square. In this way, students with a developing knowledge of the units' continuous properties inaccurately associated each unit as a discrete quantity (in this case, a grid square) rather than a continuous, dynamic movement within space. These findings resemble Shumway et al.'s (2021) findings, which described how students in a small group setting indicated knowledge of a robot's units as continuous and dynamic by gesturing along a grid path with their hands and intentionally pausing between each unit. The findings of the present study and Shumway et al.'s study both indicate how students' gestures in a grid space indicate their knowledge of the respective agents' continuous and dynamic properties.

The continuous properties of linear and rotational units used by students in the study are closely related to linear measurement, number lines, and research exploring children's readiness to work with discrete and continuous units (Boyer & Levine, 2015; Friso-van den Bos et al., 2018; Solomon et al., 2015). Number lines and measurements are composed of continuous units. Within this study's context, students who enacted programs with the agent by counting on from the start square with linear units (indicating the units' continuous properties) are similar to students' actions on number lines. For example, research indicates that students frequently include the start number on a number line rather than counting on from the start number (Baroody & Purpura, 2017) and similarly include the start number on the ruler by counting the hash marks to identify a length. In a study examining students' linear measurement using rulers and broken rulers, Solomon et al. (2015) observed that students incorrectly counted hash marks rather than the spaces between, which could have been interpreted as the students interpreting hash marks as discrete units of measure rather than the spaces between the hash marks as continuous units of measure. Solomon et al. posited that students did not perceive spatial intervals as countable units. Similarly, students in the present study indicated discrete unit interpretation when they counted each square along a path rather than the linear/rotational movements required of the agent to travel from one square to the next, which would have indicated a continuous unit interpretation.

A grid-based CT instructional context may be an entry point for practitioners and curriculum developers to connect the continuous attributes of rulers and number lines with an agent's continuous units. The units used in a grid-based CT context are dynamic (Welch et al., 2022), which might bridge students' understanding of the abstract idea of linear units on a ruler as spaces rather than concrete hash marks. One article describes how this might be achieved by positioning a screen-free robot coding toy along a number line and programming it with forward and backward codes to add and subtract (Welch et al., 2021). These types of interventions may also be effective since, as reported in the present study, students who engaged in these CT tasks frequently used gesture, and gesture has been shown to support students as they learn measurement concepts (Congdon et al., 2018).

While some students enter formal schooling with an understanding of rulers, not all understand how to use them correctly or understand how units are represented on rulers (Barrett et al., 2011). Future research would be required to identify if a correlation exists between students who interpreted an agent's movements as discrete or continuous and students' understanding of ruler units. Further, the effectiveness of interventions for students to develop unit understandings with coding toys in a CT context could inform the field if a CT context could support students' measurement knowledge development.

In summation, knowledge of units as continuous and dynamic was important for students to respond correctly to the CT assessment items. This understanding of the assessment contexts' linear and rotational units is closely related to length measurement and students' work with number lines and rulers, where students indicate similar developing knowledge of measurement units as discrete, rather than continuous, units. Additionally, learning tasks situated in a CT context may support students' further development of number line and measurement unit concepts.

Units are Iterated

The third understanding necessary for students to respond accurately to assessment items is that the program code units were iterated. This necessary understanding translates to mathematics and holds that all measurement units are iterated. Beginning in first grade, students are taught that units are iterated (CCSSM, 2010); however, the results of this study indicate that students in this context intuitively iterated the agent's movements when building and reading/enacting programs. This finding is supported by Clements and Sarama (2021) length measurement learning trajectory, which states that children ages 4–5 can iterate units to measure lengths. Students' intuitive unit iteration may have occurred in the present study because the units were dynamic and embodied by the student with the agent acting as surrogate. Students indicated unit iteration knowledge when they moved the agent along a path with pauses between each unit. Shumway et al. (2021) also observed students indicating unit knowledge in this way when students gestured over a grid in a small-group curricular context. In the example, Shumway's research group described how one student punctuated each gestured unit with the word "drive." Shumway et al.'s description of how the student indicated linear unit knowledge mirrors how, in this study, students used actions on the agent to enact programs by sliding or hoping the agent along a path with pauses between units while sometimes punctuating each unit movement with language (i.e., directional language, sounds).

The present study also observed this punctuation when students coordinated words or sounds with each enacted unit movement. Angeli and Valanides (2020) likewise observed students exhibiting unit-coordination practices when programming screen-free coding toys. According to Angeli and Valanides, students built and tested programs unit by unit. Students used this same program-building strategy in the present study, elaborated in Chapter IV, and identified these strategies as code-test or move-code strategies. For example, students used the move-code strategy by moving the agent one unit forward, then sequencing the related program code in a program. The student would repeat this move-then-code strategy until the agent reached the destination and the program was built. This method indicated unit iteration knowledge as students paused between each new enactment and associated one code with one enacted movement. Students who used the code-test or move-code strategies indicated knowledge of the one-to-one relationship between movement and program code, which is an important component of CT knowledge (Clarke-Midura, Shumway, et al., 2021).

Students who indicated unit iteration knowledge also indicated decomposition knowledge at the finite level of a unit. Other studies also observed students coding unit by unit while coding screen-free coding toys in CT tasks (Angeli & Valanides, 2020; Sung et al., 2017). Whereas Angeli and Valanides primarily interpreted decomposition from a CT perspective (decomposing a task into small pieces), the present study examined students' decomposition knowledge indicators from both an MK and CT perspective. Although the present study categorized decomposition indicators as CT, students mathematically decomposed whenever they exhibited understandings of spatial (continuous quantities) or numerical parts and wholes. When students paused between each unit enactment, they indicated an understanding that each unit was a part. Each unit iteration represented a part of the whole, the whole being the entire path or program being enacted. Students indicated the whole by continuing to iterate units to the end of a given path. Identifying parts and wholes is both an MK and CT skill. From a mathematical measurement perspective, units of measure are used to represent measurable attributes quantitatively. In this mathematical context, each unit is a part and the attribute of a specific object is the whole. In this study's CT context, the whole being considered was a path in the grid space or a given program and the parts were the program codes and the linear and rotational movements that the program codes represent. The ability to decompose mathematically (as linear and rotational unit movements) was necessary for students to represent the contextual whole (a program or path represented with program codes).

There is also a literature base describing children's unit construction (e.g., Olive, 2000; Ulrich, 2015, 2016) that may also relate to students' unit construction in coding. Future research should explore possible connections between students' arithmetical unit construction and units of measure within a CT coding context.

These findings are relevant for assessment and curricular designers seeking to integrate specific mathematics into CT assessments and curricula. These designers should consider the MK they seek to integrate and how the MK components would translate to a CT context. This section described how this study's assessment context highlighted the iterative nature of units, which could be connected to early math learning in number and measurement. As another example, in the CT1 case study presented in Chapter IV, the CT1 item design elicited students' understanding of distance as a set of unit iterations represented as a total quantity.

Spatial Concepts in Program Codes and Agent Enactment

The second theme that emerged from this study's results is that students' spatial knowledge was related to their assessment item performance, specifically when they built and enacted programs. The relationship between the MK and CT in this theme can be described as a useful or hindering co-occurrence of spatial knowledge and building/reading programs. This means that students who correctly interpreted the program codes and most frequently correctly built and read/enacted programs also indicated spatial knowledge, while students who incorrectly or inconsistently interpreted the program codes most frequently built and read/enacted programs incorrectly and indicated developing spatial knowledge. One skill related to this MK CT co-occurrence was symbolizing spatial knowledge (via an arrow or code). This might be a key skill in being able to engage in Building and Reading programs, much like learning to symbolize quantities with a numeral (Levine et al., 2010; Lipton & Spelke, 2006). Another skill related to this MK CT co-occurrence of students' spatial knowledge and building and reading/enacting programs could mean that they were able to correctly interpret program codes and this was necessary to predictably build and read/enact programs correctly on the assessment. This section will describe how students indicated spatial knowledge and how these indicators related to students' CT item performance through orienting program codes while building programs and selecting and enacting program codes.

Spatial Knowledge When Orienting Program Codes

The theme of spatial concepts in program codes speaks to students' spatial orientation knowledge. While this study attempted to fully describe students' MK while engaging with CT assessment tasks, coding for spatial orientation became meaningless as the Spatial orientation code was applicable to all instances of students building and enacting programs. While this section discusses spatial orientation, additional research would be required to describe more fully how students' spatial orientation knowledge is operationalized in a CT assessment context and how specific spatial orientation knowledge relates to students' CT assessment performance. However, what I describe in this section is specific to students' orientation of program codes and what their actions on these objects showed about their spatial orientation knowledge and how it related to building programs.

Some students indicated developing spatial knowledge in codes by orienting program codes in the cardinal direction that the student intended the agent to travel. Positioning the program codes to face the intended direction of travel indicated that the student attended to their own perspective rather than the agent's. The Common Core State Standards of Mathematics (CCSSM) for kindergarten addresses spatial orientation solely from student perspectives, although research shows that children begin developing mental rotation capabilities as early as 3 years old (Krüger, 2018). According to the CCSSM standards, kindergarten students are only required to respond from their perspective, not take up the perspective of another person or object. By operating from their own perspective, students attend to the MK requirements set by the CCSSM; however, to have responded correctly to most of the CT assessment items, students needed to adopt the agent's perspective through visualization or by repositioning themselves to take the agent's perspective (Clarke-Midura, Kozlowski, et al., 2021). As such, students who correctly oriented the program codes indicated spatial orientation knowledge addressed in the CCSSM standards and spatial orientation knowledge required for the CT task.

Orienting program codes to match the student's perspective rather than the agent's did not always relate to the students' assessment performance as the correct codes could be used, just oriented differently. Recording students' code orientation is an important insight, however, into the students' spatial orientation knowledge. For example, the CT5 case study presented in Chapter IV described how some students built the program with the correct program codes (FFF); however, they positioned the program codes to face east in alignment with the agent's beginning orientation. These students applied CCSSM spatial orientation knowledge by pointing the program code in the agent's relative direction of travel, but not the CT spatial orientation knowledge that the agent will enact the F program code from its perspective, regardless of the program code's orientation. Students who indicated both MK and CT spatial orientation knowledge interpreted the F program code to indicate that the agent moves one square forward from the agent's perspective.

Early childhood research shows that students have difficulties interpreting rotational arrows (Cuneo, 1985), which may explain why students repositioned rotational arrows. Students may have oriented rotational arrows to scaffold their understanding of the arrows' meaning. Another reason students may choose the correct program codes but orient them according to their own perspective is that research suggests that students' cognitive development is a factor in their spatial reasoning abilities (Strawhacker & Bers, 2019). I hypothesized in this study that spatial reasoning would impact students' performance since kindergarten is a highly developmental period. This study confirmed that students' spatial knowledge is related to their performance, and that mathematical and CT spatial knowledge emerged in different ways, as described in this section.

Knowing that students' ability to apply the agent's perspective relates to their interpretation of program codes has implications for CT curriculum designers. Early childhood CT curriculum designers should be aware of students' developing ability to apply an agent's perspective to program code selection and develop tasks to build this skill. This finding also suggests to mathematics curriculum designers the value of expanding spatial orientation learning opportunities from first-person and relational perspectives to the perspectives of others (Clarke-Midura, Kozlowski, et al., 2021). Since spatial skills are malleable (Uttal et al., 2013) and a fundamental element of mathematics (Clements & Sarama, 2011; Davis & the Spatial Reasoning Study Group, 2015; Mix et al., 2016), purposefully integrating spatial skill-developing opportunities into mathematics instruction could support students' overall mathematics learning.

Spatial Knowledge When Selecting and Enacting Program Codes

One of the ways students' spatial knowledge related to their assessment item performance was when they sequenced programs with codes that, when oriented correctly in the program organizer, faced the agent's intended direction of travel from the student's perspective. This placement of program codes is unique to when students positioned program codes to match the student's perspective, since students selected program codes based on the program code's cardinal arrow direction rather than the meaning of the code. For example, the CT5 case study described that, of the 60 students who responded to CT5, 13 (22%) students incorrectly built the program RRR instead of FFF but enacted their built program as FFF. This may be because, as observed by Rijke and colleagues (2018), who examined students from ages 6–12 engaging with CT robot tasks, younger children are not as readily able to abstract CT symbols as older children. The students' program code selection in the present study may be related to their developmentally-influenced abstraction skills. To use students' CT5 responses for context, the students who built the program RRR abstracted at a level in line with the CCSSM spatial knowledge standards by abstracting the program code arrows to mean directional movements from the students' perspective.

In contrast, students who correctly abstracted the program code arrows movement from the agent's perspective applied CCSSM and CT spatial knowledge. This finding aligns with Moore et al.'s (2020) observation that second-grade students had difficulty with spatial orientation when programming coding robot toys on a grid. The students in their study frequently changed their position around the grid to select the required program codes. Another study (Clarke-Midura, Kozlowski, et al., 2021) also observed students shifting positions to adopt the agent's perspective. Clarke-Midura, Kozlowski, et al. explored children's reference frames and perspective-taking while interacting with a coding robot toy on the floor. Similar to Moore et al.'s (2020) findings, Clarke-Midura, Kozlowski's research team also observed that students aligned their perspective with the robot's by moving around the floor. The robot used in Clarke-Midura, Kozlowski's study was positioned on a large grid that students could easily walk along. Students took advantage of the grid's size a position on the floor to embody the robot's movements by walking, scooting, and crawling along the grid. In contrast, students in the present study were seated at a table, and none of them attempted to stand to change their orientation, which may have scaffolded their ability to select program codes from the agent's perspective. Assessment designers might consider administering the assessment on the floor or otherwise to encourage student movement around the grid, thus allowing the student to scaffold the task by aligning their perspective with the agent's.

The agent's starting orientation for CT5 brought to light students' difficulty in applying program code meaning to the agent's perspective when compared to assessment item CT4. Item CT4 also asked students to build a linear program (correct response FFF), the difference being the agent's starting orientation as north-facing to match the student's orientation. Students responded correctly to CT4 48 times, 15 more correct responses than CT5. The difference in correct responses between CT4 and CT5 suggests that the agent's east-facing staring orientation for CT5 increased the item's difficulty and showcased students' difficulty in applying program code meanings when an agent's orientation is different from their own. Further, the difference in correct responses between CT4 and CT5 suggests that students' SK is not enough; students must coordinate SK with their own orientation, the agent's orientation in space, and the item's start/stop positions. While this was shown in the model as spatial knowledge in codes, a closer analysis indicated the importance of students' developing knowledge of spatial orientation.

Students also indicated spatial abstraction knowledge of the program code when they enacted it with the assessment agent. For example, the CT2 case study in Chapter IV described how some students enacted the program FFF by first rotating the agent left from its east-facing starting orientation to face the agent north on the grid (final enaction: LFFF). Other students might have considered the first F code in the program as a left rotation and finished by enacting FF (final enaction: LFF). Students' enaction on the agent embodied their interpretation of abstract symbols. While embodying program code interpretations through agent enactment was required for this assessment, other studies reported that embodying program codes with external agents supported students' sense-making (Moore et al., 2020; Shumway et al., 2021). Teachers and curriculum developers should design CT experiences to allow students to embody agent movements to scaffold students' spatial orientation knowledge as they develop an abstract understanding of CT program codes.

Learning to abstract is foundational to mathematics learning (Benis-Sinaceur, 2014; Clements & Sarama, 2021) and future achievement (Geary et al., 2018). Abstraction also has roots in CT (Papert, 1972), so it is not surprising that abstraction arose in this study in the context of Spatial knowledge in codes, as expected, and relates to students' CT assessment performance. Awareness of abstraction as having shared importance for young children's developing MK and CT knowledge is important for teachers and curriculum developers when designing, assessing, and carrying out CT activities. As described in this section, spatial knowledge and development is related to students' program code abstractions. Teachers and curriculum developers should not assume that students are able to immediately abstract program codes and symbols based on a cursory introduction and be aware that different students will demonstrate varying levels of abstraction. To further inform the field's understanding of students' MK and CT abstraction knowledge, additional research should be conducted to determine if students' ability to abstract CT directional program codes (i.e., arrows) relates to their ability to abstract quantities and age-appropriate mathematical symbols (i.e., +, -, =).

Implications

This study's results have implications related to CT curriculum and assessment design, CT assessment practices, mathematics curriculum design, and theory. This section will discuss each of these respectively.

The plethora of MK and MK CT co-occurring knowledge that students indicated with the CT assessment items evidences the intricate and married connections of MK within CT, grid-based tasks in early childhood. These connections have implications for CT curriculum and assessment design. Designers should be aware of the MK connections with CT and plan for them to develop curriculum and assessments appropriate for the target ages. For example, the study showed that many students' spatial language precision was still developing. Being aware of this developing knowledge informs designers of the language students are expected to use to indicate directionality.

This study also has implications for CT assessment practices in early childhood to consider which specific skills the assessment is designed to measure. For example, assessment designers and administrators should be aware of students' varied abilities to adopt an agent's perspective. Coupled with related literature's findings that students might reorient themselves to adopt an agent's perspective, the findings of the present study suggest that grid-based assessments might be administered on the floor, so students feel free to move around if the assessments are not specifically measuring students' ability to adopt an agent's perspective.

The present study's documentation of the space, measurement, and number concepts knowledge the students use when engaging with CT tasks also has implications for mathematics curriculum designers. Being aware of the specific mathematics knowledge that different tasks elicit can inform designers of how students can build mathematics knowledge in hands-on ways within a CT context. For example, this study found that students decomposed paths and programs in chunks and units when building and enacting programs. Mathematics curriculum designers could use this finding to guide students to connect how decomposing paths and programs is related to decomposing numbers.

Finally, this study also has theoretical implications in its use of an enactivist lens to examine assessment data. Assessment is used to measure what an individual knows, whereas a tenet of enactivism is that knowledge is dynamic and in a constant state of flux. This study, however, utilized enactivism uniquely to operationalize students' indicated and developing knowledge within an assessment context. An enactivist lens was necessary to make sense of students' knowledge while engaging with an assessment that used materials and practices different than they had used before.

Limitations

The primary limitations of this study include (a) sample size, (b) this study's narrow context, and (c) this study utilized an existing data set that was collected for different research objectives.

This data was drawn from one geographical location with a small population (N = 60) with limited demographic variability. A narrow geographic context limits the transferability of this study's results, however—as discussed in this chapter—many of this study's findings coincide with related research. Additional research is recommended to explore if the specific findings reported in this study translate to other populations with larger sample sizes. Additionally, disaggregating the findings by student demographics would further inform the field of how students show MK and MK CT co-occurrences and how this knowledge relates to their CT assessment item performance.

The context for the study was narrowly situated in an unplugged, grid-based, and interview-based assessment. While much of early-childhood CT research is conducted in grid-based contexts, this study is limited in its generalizability to other CT assessment contexts. Additional research would be required to identify if this study's results translate to other assessment contexts and mediums, such as independent and digital assessments.

Finally, this study was conducted using existing data collected for a larger, related research project. As such, the data collection procedures of the data set used for the present study were not specifically designed to address this study's research questions. Were the data collected for this study's purpose, additional protocols would have integrated interview questions for the assessor to derive students' thinking as they engaged with the assessment items. Doing so would have provided additional evidence to discern students' specific MK and CT knowledge.

Conclusion

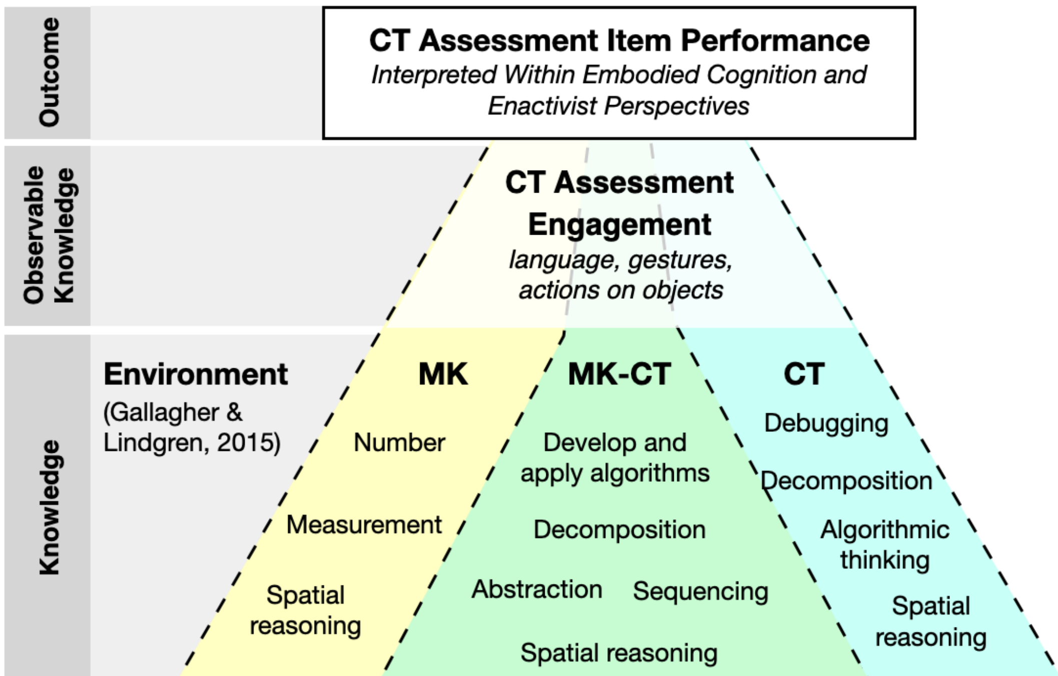
This study identified the MK, CT, and co-occurring MK CT that kindergarten students indicated during an interview-based CT assessment and described how students' MK and MK CT co-occurring knowledge related to their assessment item performance. Using the dual lens of embodied cognition and enactivism, I operationalized students' existent and developing MK and CT knowledge through students' use of language, gestures, and actions on objects while engaging with the assessment items. Results of the study showed that students' assessment performance was dependent on some MK skills (i.e., spatial knowledge, units), while other MK skills (i.e., counting on, coordinating counts) were independent of students' performance on CT assessment items. Results also indicated that most MK CT co-occurrences emerged while students built and read/ enacted programs and that students' unit and spatial knowledge were related to students' assessment item performance. Future research is required to examine if there is a directional relationship between these MK and students' CT knowledge output. These findings have implications for practitioner practices, CT assessment design, and curriculum design.

I updated CT Assessment Item Performance conceptual framework (CT-AP) in accordance with the process for obtaining the results (shown in Figure 24). The updates to the CT-AP include distinct categories of Knowledge, Observable Knowledge, and Outcomes which reflects the process of operationalizing MK, CT, and MK CT co-occurrences and ways students showed that knowledge and its connections to CT assessment performance.

Figure 24 is divided laterally into those three distinct categories (i.e., gray sections of the figure). The bottom section (labeled *Knowledge*) depicts the specific MK, CT, and co-occurring MK CT that I observed students indicate across the breadth of the CT assessment. The knowledge concepts in the bottom section are contained within the base of a triangle, indicating the knowledge is housed in the mind and body. The triangle's dashed lines, however, indicate the fluid and intermingling nature of the

Figure 24

Reconceptualized Computational Thinking Assessment Item Performance Conceptual Framework (CT-AP)



knowledge and that this knowledge is also housed in students' environment (labeled as *Environment*). The framework's middle section (labeled *Observable Knowledge*) represents the specific knowledge that I observed students indicate while engaging enacting with each assessment item through their language, gestures, and actions on objects. Finally, the top section of the framework (labeled *Outcome*) represents a student's performance on an assessment item. The story told by this framework is that a student begins a CT assessment with existing and developing MK, CT, and MK CT co-occurring knowledge (section *Knowledge*), makes their knowledge visible by engaging with an assessment item (section *Observable Knowledge*), to which their enaction of knowledge is reflected in their assessment item performance (section *Outcome*).

The revised CT-AP framework reflects this study's results within an embodied cognition perspective in that it represents knowledge as housed in students' mind, body, and environment. The framework further reflects a joint embodied and enactivist perspective by indicating that students' knowledge is observable in their language, gestures, and actions on objects.

REFERENCES

American Educational Research Association, American Psychological Association, & National Council on Measurement in Education [AERA, APA, NCME]. (2014). *Standards for educational and psychological testing*. American Psychological Association.

American Psychological Association (APA). (2019). *Publication manual of the American Psychological Association*. American Psychological Association.

Angeli, C., & Valanides, N. (2020). Developing young children's computational thinking with educational robotics: An interaction effect between gender and scaffolding strategy. *Computers in Human Behavior*, *105*, Article 105954. https://doi.org/10.1016/j.chb.2019.03.018

Arnoux, P., & Soto-Andrade, J. (2019). From concrete to abstract and back: Metaphor and Representation. In U. T. Jankvist, M. van den Heuvel-Panhuizen, & M. Veldhuis (Eds.), *Eleventh Congress of the European Society for Research in Mathematics Education* (Vol. TWG24, Issue 2). Freudenthal Group. https://hal.archives-ouvertes.fr/hal-02435190

Baroody, A. J., Feil, Y., & Johnson, A. R. (2007). An alternative reconceptualization of procedural and conceptual knowledge. *Journal for Research in Mathematics Education*, *38*(2), 115–131. https://doi.org/10.2307/30034952

Baroody, A. J., & Purpura, D. J. (2017). Early number and operations: Whole numbers. In Cai, J. (Ed.), *Compendium for research in mathematics education* (pp. 308–354). National Council of Teachers of Mathematics.

Bateson, G. (1972). *Steps to an ecology of mind*. Ballentine Books.

Barrett, J. E., Cullen, C., Sarama, J., Clements, D. H., Klanderman, D., Miller, A. L., & Rumsey, C. (2011). Children's unit concepts in measurement: A teaching experiment spanning grades 2 through 5. *ZDM*, *43*(5), 637–650. https://doi.org/10.1007/s11858-011-0368-8

Battista, M. T., Winer, M. L., & Frazee, L. M. (2017). How spatial reasoning and numerical reasoning are related in geometric measurement. In E. Galindo & J. Newton (Eds.), *Proceedings of the 39th annual meeting of the North American Chapter of the International Group for the Psychology of Mathematics Education* (pp. 355–362). Indianapolis, IN: Hoosier Association of Mathematics Teacher Educators.

Benis-Sinaceur, H. (2014). Facets and levels of mathematical abstraction. *Philosophia Scientae*, *18*(1), 81–112. https://doi.org/10.4000/philosophiascientiae.914

Bers, M. U., González-González, C., & Armas-Torres, M. B. (2019). Coding as a playground: Promoting positive learning experiences in childhood classrooms. *Computers & Education*, 138, 130–145. https://doi.org/10.1016/j.compedu.2019.04.013

Blikstad-Balas, M. (2017). Key challenges of using video when investigating social practices in education: Contextualization, magnification, and representation. *International Journal of Research & Method in Education*, *40*(5), 511–523. https://doi.org/10.1080/1743727X.2016.1181162

Booth, J. L., & Siegler, R. S. (2008). Numerical magnitude representations influence arithmetic learning. *Child Development*, *79*(4), 1016–1031. https://doi.org/10/c93ctp

Boticki, I., Pivalica, D., & Seow, P. (2018). The use of computational thinking concepts in early primary school. In S. C. Kong, D. Andone, G. Biswas, T. Crick, H. U. Hoppe, T. C. Hsu, R. H. Huang, K. Y. Li, C. K. Looi, M. Milrad, J. Sheldon, J. L. Shih, K. F. Sin, M. Tissenbaum, & J. Vahrenhold (Eds.), *Proceedings of the International Conference on Computational Thinking Education 2018* (pp. 8-13). The Education University of Hong Kong.

Boyer, T. W., & Levine, S. C. (2015). Prompting children to reason proportionally: Processing discrete units as continuous amounts. *Developmental Psychology*, *51*(5), 615–620.

Brown, L., & Coles, A. (2011). Developing expertise: How enactivism re-frames mathematics teacher development. *ZDM*, *43*, 861–873. https://doi.org/10.1007/s11858-011-0343-4

Bruce, C. D., Davis, B., Sinclair, N., McGarvey, L., Hallowell, D., Drefs, M., Francis, K., Hawes, Z., Moss, J., Mulligan, J., Okamoto, Y., Whiteley, W., & Woolcott, G. (2017). Understanding gaps in research networks: Using "spatial reasoning" as a window into the importance of networked educational research. *Educational Studies in Mathematics*, *95*(2), 143–161. https://doi.org/10.1007/s10649-016-9743-2

Bruner, J. S. (1961). The act of discovery. *Harvard Educational Review*, *31*(1), 21–32.

Campbell, S. (2010). Embodied minds and dancing brains: New opportunities for research in mathematics education. In B. Sriraman & L. English (Eds.), *Theories of mathematics education: Seeking new frontiers* (pp. 309–331). Springer.

Cetin, I., & Dubinsky, E. (2017). Reflective abstraction in computational thinking. *The Journal of Mathematical Behavior*, *47*, 70–80. https://doi.org/10.1016/j.jmathb.2017.06.004

Christopher, A. N. (2017). Describing data with frequency distributions and visual displays. In *Interpreting and using statistics in psychological research* (pp. 53–88). SAGE Publications. https://doi.org/10.4135/9781506304144

Città, G., Gentile, M., Allegra, M., Arrigo, M., Conti, D., Ottaviano, S., Reale, F., & Sciortino, M. (2019). The effects of mental rotation on computational thinking. *Computers & Education*, *141*, Article 103613. https://doi.org/10.1016/j.compedu.2019.103613

Clarke-Midura, J., Kozlowski, J. S., Shumway, J. F., & Lee, V. R. (2021). How young children engage in and shift between reference frames when playing with coding toys. *International Journal of Child-Computer Interaction*, *28*, 100250. https://doi.org/10.1016/j.ijcci.2021.100250

Clarke-Midura, J., Shumway, J., Lee, V. F., Silvis, D., Welch, L., Kozlowski, J., Peterson, R., Lewis, S., Bullock, K., & Childers, K. (2021). *Coding in Kindergarten (CiK) project report 2019-2020: Empirical foundations of CT model development* [White paper]. Utah State University.

Clarke-Midura, J., Silvis, D., Shumway, J. F., Lee, V. R., & Kozlowski, J. S. (2021). Developing a kindergarten computational thinking assessment using evidence-centered design: The case of algorithmic thinking. *Computer Science Education*, *31*(2), 117–140. https://doi-org.dist.lib.usu.edu/10.1080/08993408.2021.1877988

Clements, D. H., Battista, M. T., Sarama, J., & Swaminathan, S. (1996). Development of turn and turn measurement concepts in a computer-based instructional unit. *Educational Studies in Mathematics*, *30*(4), 313–337. https://doi.org/10.1007/BF00570828

Clements, D. H., & Sarama, J. (1997). Research on Logo: A decade of progress. *Computers in the Schools*, *14*(1–2), 9–46. https://doi.org/10/cgzmks

Clements, D. H., & Sarama, J. (2009). *Learning and teaching early math: The learning trajectories approach* (Vol. 2). Routledge.

Clements, D. H., & Sarama, J. (2011). Early childhood teacher education: The case of geometry. *Journal of Mathematics Teacher Education*, *14*(2), 133–148. https://doi.org/10.1007/s10857-011-9173-0

Clements, D. H., & Sarama, J. (2021). *Learning and teaching early math: The learning trajectories approach* (Vol. 3). Routledge.

Clements, D. H., Sarama, J., & Wolfe, C. B. (2011). *TEAM—Tools for Early Assessment in Mathematics.* McGraw-Hill Education.

Clements, D. H., Sarama, J. Wolfe, C. B., Day-Hess, C. A. (2019). *REMA—Research-based Early Mathematics Assessment*. Kennedy Institute, University of Denver.

Common Core State Standards Initiative (CCSSI). (2010). *Common core state standards for mathematics*. http://www.corestandards.org/Math/

Computer Science Teachers Association (CSTA). (2017). *K-12 computer science standards*. Creative Commons Attributions.

Congdon, E. L., Kwon, M.-K., & Levine, S. C. (2018). Learning to measure through action and gesture: Children's prior knowledge matters. *Cognition*, *180*, 182–190. https://doi.org/10.1016/j.cognition.2018.07.002

Cooperrider, K., Gentner, D., & Goldin-Meadow, S. (2016). Spatial analogies pervade complex relational reasoning: Evidence from spontaneous gestures. *Cognitive Research: Principles and Implications*, *1*(28), 1–17. https://doi.org/10.1186/s41235-016-0024-5

Crollen, V., & Noël, M.-P. (2015). The role of fingers in the development of counting and arithmetic skills. *Acta Psychologica*, *156*, 37–44. https://doi.org/10.1016/j.actpsy.2015.01.007

Crooks, N. M., & Alibali, M. W. (2014). Defining and measuring conceptual knowledge in mathematics. *Developmental Review*, *34*(4), 344–377. https://doi.org/10.1016/j.dr.2014.10.001

Cuneo, D. O. (1985, April 25). *Young children and turtle graphics programming: Understanding turtle commands* [paper presentation]. Biennial Meeting of the Society for Research in Child Development, Toronto, Ontario, Canada.

Davis, B. (1996). *Teaching mathematics: Towards a sound alternative*. Garland Publishing Inc.

Davis, B., & Francis, K. (2020). Embodied cognition. *Discourses on Learning in Education*. https://learningdiscourses.com/discourse/embodied-cognition/

Davis, B., & the Spatial Reasoning Study Group (SRSG). (2015). *Spatial reasoning in the early years: Principles, assertions, and speculations*. Routledge.

Davydov, V. V. (1975). Logical and psychological problems of elementary mathematics as an academic subject. In L. P. Steffe (Ed.), *Children's capacity for learning mathematics. Soviet studies in the psychology of learning and teaching mathematics* (Vol. 7, pp. 55–107). Chicago: University of Chicago.

De Smedt, B., Verschaffel, L., & Ghesquière, P. (2009). The predictive value of numerical magnitude comparison for individual differences in mathematics achievement. *Journal of Experimental Child Psychology*, *103*(4), 469–479. https://doi.org/10/fdq4c5

Dickes, A. C., Farris, A. V., & Sengupta, P. (2020). Sociomathematical norms for integrating coding and modeling with elementary science: A dialogical approach. *Journal of Science Education and Technology*, *29*(1), 35–52. https://doi.org/10.1007/s10956-019-09795-7

Doidge, N. (2007). *The brain that changes itself: Stories of personal triumph from the frontiers of brain science*. Penguin Books.

Dougherty, B. (2008). Measure up: A quantitative view of early algebra. In J. J. Kaput, D. W. Carraher, & M. L. Blanton (Eds.), *Algebra in the early grades* (pp. 389–412). Routledge. https://doi.org/10.4324/9781315097435-18

Ehsan, H., & Cardella, M. (2017). Capturing the computational thinking of families with young children in out-of-school environments. *2017 ASEE Annual Conference & Exposition Proceedings*, Paper 19605. https://doi.org/10.18260/1-2--28010

Ferrara, K., Hirsh-Pasek, K., Newcombe, N. S., Golinkoff, R. M., & Lam, W. S. (2011). Block talk: Spatial language during block play. *Mind, Brain, and Education*, *5*(3), 143–151. https://doi.org/10.1111/j.1751-228X.2011.01122.x

Francis, K., Khan, S., & Davis, B. (2016). Enactivism, spatial reasoning and coding. *Digital Experiences in Mathematics Education*, *2*(1), 1–20. https://doi.org/10.1007/s40751-015-0010-4

Friso-van den Bos, I., Kroesbergen, E. H., & Van Luit, J. E. H. (2018). Counting and number line trainings in kindergarten: Effects on arithmetic performance and number sense. *Frontiers in Psychology*, *9*, Article 975. https://doi.org/10.3389/fpsyg. 2018.00975

Fuchs, L. S., Fuchs, D., & Zumeta, R. O. (2008). A curricular-sampling approach to progress monitoring: Mathematics concepts and applications. *Assessment for Effective Intervention*, *33*(4), 225–233. https://doi.org/10/d2t2x4

Gallagher, S. (2017). *Enactivist interventions: Rethinking the mind*. Oxford University Press.

Gallagher, S., & Lindgren, R. (2015). Enactive metaphors: Learning through full-body engagement. *Educational Psychology Review*, *27*(3), 391–404. https://doi.org/10.1007/s10648-015-9327-1

García-Valcárcel-Muñoz-Repiso, A., & Caballero-González, Y.-A. (2019). Robotics to develop computational thinking in early childhood education. *Comunicar*, *27*(59), 63–72. https://doi.org/10.3916/C59-2019-06

Geary, D. C., vanMarle, K., Chu, F. W., Rouder, J., Hoard, M. K., & Nugent, L. (2018). Early conceptual understanding of cardinality predicts superior school-entry number-system knowledge. *Psychological Science*, *29*(2), 191–205. https://doi.org/10/gc2tnn

Ginsburg, H., & Baroody, A. (2003). *Test of early mathematics ability* (3rd ed.). Pro-Ed.

Gunderson, E. A., Ramirez, G., Beilock, S. L., & Levine, S. C. (2012). The relation between spatial skill and early number knowledge: The role of the linear number line. *Developmental Psychology*, *48*(5), 1229–1241. https://doi.org/10.1037/a0027433

Hamilton, M., Clarke-Midura, J., Shumway, J. F., & Lee, V. R. (2020). An emerging technology report on computational toys in early childhood. *Technology, Knowledge and Learning*, *25*(1), 213–224. https://doi.org/10.1007/s10758-019-09423-8

Hickmott, D., Prieto-Rodriguez, E., & Holmes, K. (2018). A scoping review of studies on computational thinking in K–12 mathematics classrooms. *Digital Experiences in Mathematics Education*, *4*(1), 48–69. https://doi.org/10.1007/ s40751-017-0038-8

Hoffman, H., & Grialou, T. (2005). Test of early mathematics ability (3rd ed.) by Ginsburg, H. P., & Baroody, A. J. (2003). Austin, TX: PRO-ED. *Assessment for Effective Intervention*, *30*(4), 57–60. https://doi.org/10.1177/073724770503000409

Hojnoski, R. L., Silberglitt, B., & Floyd, R. G. (2009). Sensitivity to growth over time of the preschool numeracy indicators with a sample of preschoolers in Head Start. *School Psychology Review*, *38*(3), 402–418. https://doi.org/10.1080/02796015.2009.12087823

Hunting, R. (2003). Part-whole number knowledge in preschool students. *The Journal of Mathematical Behavior*, *22*(3), 217–235. http://dx.doi.org/10.1016/S0732-3123(03)00021-X

Jordan, N. C., Glutting, J., Ramineni, C., & Watkins, M. W. (2010). Validating a number sense screening tool for use in kindergarten and first grade: Prediction of mathematics proficiency in third grade. *School Psychology Review*, *39*(2), 181–195. https://doi.org/10.1080/02796015.2010.12087772

Kadijevich, D. M. (2018). Relating procedural and conceptual knowledge. *The Teaching of Mathematics*, *21*(1), 15–28. http://ipir.ipisr.org.rs/handle/123456789/298

Kazakoff, E. R., Sullivan, A., & Bers, M. U. (2013). The effect of a classroom-based intensive robotics and programming workshop on sequencing ability in early childhood. *Early Childhood Education Journal*, *41*(4), 245–255. https://doi.org/10.1007/s10643-012-0554-5

Khan, S., Francis, K., & Davis, B. (2015). Accumulation of experience in a vast number of cases: Enactivism as a fit framework for the study of spatial reasoning in mathematics education. *ZDM*, *47*(2), 269–279. https://doi.org/10.1007/s11858-014-0623-x

Kiefer, P., Giannopoulos, I., Raubal, M., & Duchowski, A. (2017). Eye tracking for spatial research: Cognition, computation, challenges. *Spatial Cognition & Computation*, *17*(1–2), 1–19. https://doi.org/10.1080/13875868.2016.1254634

Klein, A. & Starkey, P. (2006). *Child math assessment.* Berkeley: University of California

Kotsopoulos, D., Makosz, S., & Zambrzycka, J. (2017). Number knowledge and young children's ability to measure length. *Early Education and Development*, *28*(8), 925–938. http://dx.doi.org.dist.lib.usu.edu/10.1080/10409289.2015.1060801

Krüger, M. (2018). Three-year-olds solved a mental rotation task above chance level, but no linear relation concerning reaction time and angular disparity presented itself. *Frontiers in Psychology*, *9*, Article 1796. https://doi.org/10.3389/fpsyg.2018.01796

Kucian, K., Grond, U., Rotzer, S., Henzi, B., Schönmann, C., Plangger, F., Gälli, M., Martin, E., & von Aster, M. (2011). Mental number line training in children with developmental dyscalculia. *NeuroImage*, *57*(3), 782–795. https://doi.org/10.1016/j.neuroimage.2011.01.070

Lavigne, H. J., Lewis-Presser, A., & Rosenfeld, D. (2020). An exploratory approach for investigating the integration of computational thinking and mathematics for preschool children. *Journal of Digital Learning in Teacher Education*, *36*(1), 63–77. https://doi.org/10.1080/21532974.2019.1693940

Levine, S. C., Suriyakham, L. W., Rowe, M. L., Huttenlocher, J., & Gunderson, E. A. (2010). What counts in the development of young children's number knowledge? *Developmental Psychology*, *46*(5), 1309–1319. https://doi.org/10.1037/a0019671

Li, Q., Clark, B., & Winchester, I. (2010). Instructional design and technology grounded in enactivism: A paradigm shift? *British Journal of Educational Technology*, *41*(3), 403–419. https://doi.org/10.1111/j.1467-8535.2009.00954.x

Libertus, M. E., Feigenson, L., & Halberda, J. (2011). Preschool acuity of the approximate number system correlates with school math ability: Approximate number system and math abilities. *Developmental Science*, *14*(6), 1292–1300. https://doi.org/10/bjxwz6

Lipton, J. S., & Spelke, E. S. (2006). Preschool children master the logic of number word meanings. *Cognition*, *98*(3), 1–10. https://doi.org/10.1016/j.cognition.2004.09.013

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. https://doi.org/10. 1016/j.chb.2014.09.012

MacDonald, A., & Lowrie, T. (2011). Developing measurement concepts within context: Children's representations of length. *Mathematics Education Research Journal*, *23*(1), 27–42. https://doi.org/10/bvqjp9

Maheux, J.-F., & Proulx, J. (2015). Doing mathematics: Analysing data with/in an enactivist-inspired approach. *ZDM*, *47*(2), 211–221. https://doi.org/10.1007/s11858-014-0642-7

Merleau-Ponty, M. (1964). *The primacy of perception and other essays.* Northwestern University Press.

Miller, J. (2019). STEM education in the primary years to support mathematical thinking: Using coding to identify mathematical structures and patterns. *ZDM*, *51*(6), 915–927. https://doi.org/10.1007/s11858-019-01096-y

Mix, K. S., Levine, S. C., Cheng, Y.-L., Young, C., Hambrick, D. Z., Ping, R., & Konstantopoulos, S. (2016). Separate but correlated: The latent structure of space and mathematics across development. *Journal of Experimental Psychology: General*, *145*(9), 1206–1227. https://doi.org/10.1037/xge0000182

Moore, T. J., Brophy, S. P., Tank, K. M., Lopez, R. D., Johnston, A. C., Hynes, M. M., & Gajdzik, E. (2020). Multiple representations in computational thinking tasks: A clinical study of second-grade students. *Journal of Science Education and Technology*, *29*(1), 19–34. https://doi.org/10.1007/s10956-020-09812-0

Nam, K. W., Kim, H. J., & Lee, S. (2019). Connecting plans to action: The effects of a card-coded robotics curriculum and activities on Korean kindergartners. *The Asia-Pacific Education Researcher*, *28*(5), 387–397. https://doi.org/10.1007/s40299-019-00438-4

National Research Council. (1990). *Reshaping school mathematics: A philosophy and framework for curriculum*. National Academy Press. https://doi.org/10.17226/1498

National Research Council. (2009). *Mathematics learning in early childhood: Paths toward excellence and equity* (C. T. Cross, T. A. Woods, & H. A. Schweingruber, Eds.). National Academies Press.

Newcombe, N. S., & Shipley, T. F. (2015). Thinking about spatial thinking: New typology, new assessments. In J. S. Gero (Ed.), *Studying visual and spatial reasoning for design creativity* (pp. 179–192). Springer. http://dx.doi.org/10.1007/978-94-017-9297-4_10

Nguyen, T., Watts, T. W., Duncan, G. J., Clements, D. H., Sarama, J. S., Wolfe, C., & Spitler, M. E. (2016). Which preschool mathematics competencies are most predictive of fifth grade achievement? *Early Childhood Research Quarterly*, *36*, 550–560. https://doi.org/10/gf8gr7

Núñez, R. E., Edwards, L. D., & Matos, J. F. (1999). Embodied cognition as grounding for situatedness and context in mathematics education. *Educational Studies in Mathematics*, *39*(1), 45–65.

Obersteiner, A., Reiss, K., & Ufer, S. (2013). How training on exact or approximate mental representations of number can enhance first-grade students' basic number processing and arithmetic skills. *Learning and Instruction*, *23*, 125–135. https://doi.org/10.1016/j.learninstruc.2012.08.004

Olive, J. (2000). Children's number sequences: An explanation of Steffe's constructs and an extrapolation to rational numbers of arithmetic. *The Mathematics Educator, 11*(1), 4–9.

Palmér, H. (2017). Programming in preschool: With a focus on learning mathematics. *Early Childhood Education*, *8*(1), 75–87.

Papert, S. (1972). Teaching children to be mathematicians versus teaching about mathematics. *International Journal of Mathematical Education in Science and Technology*, *3*(3), 249–262. https://doi.org/10.1080/0020739700030306

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

Papert, S. (1996). Metaphorical intentions. *International Journal of Computers for Mathematical Learning*, *1*(1), 95–123.

Pasnak, R., Kidd, J. K., Gadzichowski, K. M., Gallington, D. A., Schmerold, K. L., & West, H. (2015). Abstracting sequences: Reasoning that is a key to academic achievement. *The Journal of Genetic Psychology*, *176*(3), 171–193. https://doi.org/10.1080/00221325.2015.1024198

Pérez, A. (2018). A framework for computational thinking dispositions in mathematics education. *Journal for Research in Mathematics Education, 49*(4), 424-461.

Purpura, D. J., Baroody, A. J., & Lonigan, C. J. (2013). The transition from informal to formal mathematical knowledge: Mediation by numeral knowledge. *Journal of Educational Psychology*, *105*(2), 453–464. https://doi.org/10.1037/a0031753

Reid, D. A. (2014). The coherence of enactivism and mathematics education research: A case study. *AVANT. The Journal of the Philosophical-Interdisciplinary Vanguard*, *V*(2), 137–172. https://doi.org/10.26913/50202014.0109.0007

Relkin, E., de Ruiter, L., & Bers, M. U. (2020). TechCheck: Development and validation of an unplugged assessment of computational thinking in early childhood education. *Journal of Science Education and Technology*, *29*(4), 482–498. https://doi.org/10/gj2p2m

Resnick, I., Harris, D., Logan, T., & Lowrie, T. (2020). The relation between mathematics achievement and spatial reasoning. *Mathematics Education Research Journal*, *32*(2), 171–174. https://doi.org/10.1007/s13394-020-00338-7

Rich, K. M., Strickland, C., Binkowski, T. A., Moran, C., & Franklin, D. (2017). K–8 learning trajectories derived from research literature: Sequence, repetition, conditionals. *Proceedings of the 2017 ACM Conference on International Computing Education Research*, 182–190.

Rich, K. M., Yadav, A., & Zhu, M. (2019). Levels of abstraction in students' mathematics strategies: What can applying computer science ideas about abstraction bring to elementary mathematics? *Journal of Computers in Mathematics and Science Teaching*, *38*(3), 267–298.

Rijke, W. J., Bollen, L., Eysink, T. H. S., & Tolboom, J. L. J. (2018). Computational thinking in primary school: An examination of abstraction and decomposition in different age groups. *Informatics in Education*, *17*(1), 77–92. https://doi.org/10.15388/infedu.2018.05

Rittle-Johnson, B., Schneider, M., & Star, J. R. (2015). Not a one-way street: Bidirectional relations between procedural and conceptual knowledge of mathematics. *Educational Psychology Review*, *27*(4), 587–597. https://doi.org/10.1007/s10648-015-9302-x

Saldaña, J. (2021). *The coding manual for qualitative researchers* (4th ed). SAGE Publications.

Saxena, A., Lo, C. K., Hew, K. F., & Wong, G. K. W. (2020). Designing unplugged and plugged activities to cultivate computational thinking: An exploratory study in early childhood education. *The Asia-Pacific Education Researcher*, *29*(1), 55–66. https://doi.org/10.1007/s40299-019-00478-w

Schrank, F. A., McGrew, K. S., & Mather, N. (2014). *Woodcock-Johnson IV Tests of Cognitive Abilities*. Riverside Publishing.

Shumway, J. F., Welch, L. E., Kozlowski, J., Clarke-Midura, J., & Lee., V. R. (2021). Kindergarten students' mathematics knowledge at work: The mathematics for programming robot toys. *Mathematical Thinking and Learning*. Online first. https://doi.org/10.1080/10986065.2021.1982666

Shute, V. J., Sun, C., & Asbell-Clarke, J. (2017). Demystifying computational thinking. *Educational Research Review*, *22*, 142–158. https://doi.org/10.1016/j.edurev.2017.09.003

Silvis, D., Clarke-Midura, J., Shumway, J.F. Lee, V. (2021, June 8–11). *Objects to debug with: How young children resolve errors with tangible programming toys* [Paper presentation]. Annual Meeting of International Society of Learning Sciences, Bochum, Germany.

Simms, V., Clayton, S., Cragg, L., Gilmore, C., & Johnson, S. (2016). Explaining the relationship between number line estimation and mathematical achievement: The role of visuomotor integration and visuospatial skills. *Journal of Experimental Child Psychology*, *145*, 22–33. https://doi.org/10.1016/j.jecp.2015.12.004

Smith, J. P., & Barrette, J. E. (2019). Learning and teaching measurement: Coordinating quantity and number. In J. Cai (Ed.), *Compendium for research in mathematics education* (pp. 355–385). National Council of Teachers of Mathematics.

Smith, M. (2016, January 30). *Computer science for all*. https://www.whitehouse.gov/blog/2016/01/30/computer-science-all

Solomon, T. L., Vasilyeva, M., Huttenlocher, J., & Levine, S. C. (2015). Minding the gap: Children's difficulty conceptualizing spatial intervals as linear measurement units. *Developmental Psychology*, *51*(11), 1564–1573. http://dx.doi.org.dist.lib.usu.edu/10.1037/a0039707

Sorby, S., Veurink, N., & Streiner, S. (2018). Does spatial skills instruction improve STEM outcomes? The answer is 'yes.' *Learning and Individual Differences*, *67*, 209–222. https://doi.org/10.1016/j.lindif.2018.09.001

Sparks, A., Wellberg, S., Geller, J., McMurrer, J., Hatfield, C., & Ketterlin-Geller, L. (2021, June 8–11). *Developing a survey to assess at-home spatial reasoning* [Poster presentation]. National Council on Measurement in Education Conference, Virtual. https://smu.app.box.com/s/rosih63es7gfiv0n8cofpoink52ytk03

Sprenger, P., & Benz, C. (2020). Children's perception of structures when determining cardinality of sets—Results of an eye-tracking study with 5-year-old children. *ZDM*, *52*(4), 753–765. https://doi.org/10.1007/s11858-020-01137-x

Star, J. R. (2005). Reconceptualizing procedural knowledge. *Journal for Research in Mathematics Education*, *36*(5), 404–411. https://doi.org/10.2307/30034943

Stephens, M. (2018). Embedding algorithmic thinking more clearly in the mathematics curriculum. In Y. Shimizu & R. Withal (Eds.), *Proceedings of ICMI study 24 school mathematics curriculum reforms: Challenges, changes and opportunities* (pp. 483–490). University of Tsukuba. https://www.mathunion.org/fileadmin/ICMI/ICMI%20studies/ICMI%20Study%2024/ICMI%20Study%2024%20Proceedings.pdf

Strawhacker, A., & Bers, M. U. (2019). What they learn when they learn coding: Investigating cognitive domains and computer programming knowledge in young children. *Educational Technology Research and Development*, *67*(3), 541–575. https://doi.org/10.1007/s11423-018-9622-x

Strawhacker, A., Sullivan, A., & Bers, M. U. (2013). TUI, GUI, HUI: Is a bimodal interface truly worth the sum of its parts? *Proceedings of the 12th International Conference on Interaction Design and Children* (pp. 309–312). https://doi.org/10/gksm7x

Sullivan, A. A., Bers, M. U., & Mihm, C. (2017). Imagining, playing, and coding with KIBO: Using robotics to foster computational thinking in young children. In S. C. Kong, J. Sheldon, & K. Y. Li (Eds.), *Conference Proceedings of International Conference on Computational Thinking Education 2017* (pp. 110–115). The Education University of Hong Kong.

Sung, W., Ahn, J., & Black, J. B. (2017). Introducing computational thinking to young learners: Practicing computational perspectives through embodiment in mathematics education. *Technology, Knowledge and Learning*, *22*(3), 443–463. https://doi.org/10.1007/s10758-017-9328-x

Szilágyi, J., Clements, D. H., & Sarama, J. (2013). Young children's understandings of length measurement: Evaluating a learning trajectory. *Journal for Research in Mathematics Education*, *44*(3), 581. https://doi.org/10.5951/jresematheduc.44.3.0581

Tang, X., Yin, Y., Lin, Q., Hadad, R., & Zhai, X. (2020). Assessing computational thinking: A systematic review of empirical studies. *Computers & Education*, *148*, Article 103798. https://doi.org/10.1016/j.compedu.2019.103798

Thompson P. W., & Carlson, M. P. (2017). Variation, covariation, and functions: Foundational ways of thinking mathematically. In J. Cai (Ed.), *Compendium for research in mathematics education* (pp. 421–456). National Council of Teachers of Mathematics.

Ulrich, C. (2015). Stages in constructing and coordinating units additively and multiplicatively (part 1). *For the Learning of Mathematics*, *35*(3), 2–7.

Ulrich, C. (2016). Stages in constructing and coordinating units additively and multiplicatively (part 2). *For the Learning of Mathematics*, *36*(1), 34–39.

Uttal, D. H., Meadow, N. G., Tipton, E., Hand, L. L., Alden, A. R., Warren, C., & Newcombe, N. S. (2013). The malleability of spatial skills: A meta-analysis of training studies. *Psychological Bulletin*, *139*(2), 352–402. https://doi.org/10.1037/a0028446

Vanbinst, K., & De Smedt, B. (2016). Individual differences in children's mathematics achievement: The roles of symbolic numerical magnitude processing and domain-general cognitive functions. *Progress in Brain Research*, *227*, 105–130. https://doi.org/10/gjvzc2

Varela, F., Thompson, E. & Rosch, E. (1991). *The embodied mind: Cognitive science and human experience*. Massachusetts Institute of Technology Press.

Venenciano, L., & Dougherty, B. (2014). Addressing priorities for elementary school mathematics. *For the Learning of Mathematics*, *1*, 18–24.

Walkington, C., Chelule, G., Woods, D., & Nathan, M. J. (2019). Collaborative gesture as a case of extended mathematical cognition. *The Journal of Mathematical Behavior*, *55*, Article 100683. https://doi.org/10.1016/j.jmathb.2018.12.002

Wang, X. C., Choi, Y., Benson, K., Eggleston, C., & Weber, D. (2021). Teacher's role in fostering preschoolers' computational thinking: An exploratory case study. *Early Education and Development*, *23*(1), 26–48. https://doi.org/10.1080/10409289.2020.1759012

Weintrop, D., Beheshti, E., Horn, M., Orton, K., Jona, K., Trouille, L., & Wilensky, U. (2016). Defining computational thinking for mathematics and science classrooms. *Journal of Science Education and Technology*, *25*(1), 127–147. https://doi.org/10.1007/s10956-015-9581-5

Welch, L. E., Shumway, J. F., Clarke-Midura, J., & Lee, V. R. (2021). Using coding toys to understand equality. *Australian Primary Mathematics Classroom*, *26*(3), 21-25.

Welch, L. E., Shumway, J. F., Clarke-Midura, J., & Lee, V. R. (2022). Exploring measurement through coding: Children's conceptions of a dynamic linear unit with robot coding toys. *Education Sciences*, *12*(2), 143. https://doi.org/10.3390/educsci12020143

Wilkins, J. L. M., Woodward, D., & Norton, A. (2021). Children's number sequences as predictors of later mathematical development. *Mathematics Education Research Journal 33*, 513–540. https://doi-org.dist.lib.usu.edu/10.1007/s13394-020-00317-y

Wing, J. M. (2006). Computational thinking. *Communications of the ACM, 49*(3), 33–35.

Woolcott, G., Le Tran, T., Mulligan, J., Davis, B., & Mitchelmore, M. (2020). Towards a framework for spatial *reasoning* and primary mathematics learning: An analytical synthesis of intervention studies. *Mathematics Education Research Journal, 34*(3), 37–67. https://doi.org/10.1007/s13394-020-00318-x

Yadav, A., Mayfield, C., Zhou, N., Hambrusch, S., & Korb, J. T. (2014). Computational thinking in elementary and secondary teacher education. *ACM Transactions on Computing Education*, *14*(1), Article 5. https://doi.org/10.1145/2576872

Yang, X., & Pan, Y. (2021). Spatial language of young children during block play in kindergartens in urban China. *Frontiers in Psychology*, *12*, Article 568638. https://doi.org/10.3389/fpsyg.2021.568638

Yin, R. K. (2018). *Case study research and applications: Design and methods* (6th edition). SAGE.

APPENDICES

Appendix A

IRB Approval

IRB Approval

Utah State University Institutional Review Board (IRB) Proposal #9569: Approved

Human Subject Approval or Completion Date: 2018-05-07

Responsible Conduct Approval or Completion Date: 2021-05-10

Amendment #13:

I would like to use this data for my dissertation. Attached is my approved proposal.

Link to uploaded file: LiseWelch_Proposal_21-10-7.docx

Amendment #13 Oct. 7, 2021 *by* Lise Welch

I would like to use this data for my dissertation. Attached is my signed cover sheet.

Link to uploaded file: Title Page_Welch, Lise_A00584262 - signed.pdf

Amendment #13 Oct. 7, 2021 *by* Lise Welch

Appendix B

Assessment Scoring Sheets

Figure B.1

Assessment Scoring Sheet 1

Assessment Record: Computational Thinking
Algorithmic Thinking, Debugging, Decomposition & Pattern Recognition

Child Name/Study ID____________________________ M / F

School__________________________________ City/Site __________________________________

Grade___________________ Teacher Name/Classroom ________________________________

Date of Assessment: ______/______/_______ Assessor Name ____________________________

Video File Name: ____

If there was any reason to believe this assessment was not a fair assessment of this child, explain why in the space below:

Data Entered: ______/______/_______ by ____

Coded by: ___________________________________ Date of Coding ______/______/_______

To be Completed by Coder:
Stop rule of 3 in a row incorrect met? Y N
All codes completed by assessor? Y N

Figure B.2

Assessment Scoring Sheet 2

Item	Task	Code	Strategy Code	Response/Notes
SR1	Forward	**1A:** 0 1 9		
	Correct: Moves object forward 1 square		Memo:	
SR2	Forward (rotated)	**2A:** 0 1 9		
	Correct: Moves object forward 1 square		Memo:	
SR3	Rotate right	**3A:** 0 1 9		
	Correct: Rotate object to the right		Memo:	
SR4	Rotate right (rotated)	**4A:** 0 1 9		
	Correct: Rotate object to the right		Memo:	
AT1	Sequence unknown path- easy	**1A:** 0 1 9	**B** **C**	
	Correct: 2 or FF		Memo:	
AT2	Move a known path- easy	**2A:** 0 1 9	**B** **C**	
	Correct: Moves object forward 3 squares		Memo:	
AT3	Fill in turn- easy	**3A:** 0 1 9	**B** **C**	
	Correct: L		Memo:	
AT4	Sequence unknown path- easy	**4A:** 0 1 9	**B** **C**	
	Correct: FFF		Memo:	
AT5	Sequence unknown path (rotated)- easy	**5A:** 0 1 9	**B** **C**	
	Correct: FFF		Memo:	
AT6	Execute sequence- medium	**6A:** 0 1 9	**B** **C**	
	Correct: Moves object to diagonally adjacent square		Memo:	

Hillcrest Midpoint 2_20200130

Figure B.3

Assessment Scoring Sheet 3

AT7	Execute sequence (two turns)- hard	**7A:** 0 1 9	**B**	**C**	
	Correct: Move object two squares back from start		Memo:		
AT8	Sequence known path	**8A:** 0 1 9	**B**	**C**	
	Correct: FFLFF		Memo:		
AT9	Sequence unknown path	**9A:** 0 1 9	**B**	**C**	
	Correct: FRF		Memo:		
AT10	Sequence unknown path- hard	**10A:** 0 1 9	**B**	**C**	
	Correct: LFFF or FLFFFLFR		Memo:		
AT11	Use a simple function	**11A:** 0 1 9	**B**	**C**	
	Correct: 3 circles		Memo:		
DB1	Recognize error- easy	**1A:** 0 1 9	**B**	**C**	
	Correct: No		Memo:		
DB2	Recognize error- medium	**2A:** 0 1 9	**B**	**C**	
	Correct: Yes		Memo:		
DB3	Locate error (middle)	**3A:** 0 1 9	**B**	**C**	
	Correct: Point to R		Memo:		
DB4	Locate error (end)	**4A:** 0 1 9	**B**	**C**	
	Correct: Extra F at end		Memo:		

Figure B.4

Assessment Scoring Sheet 4

DB5	Locate error (beginning)	**5A:** 0 1 9	**B**	**C**	
	Correct: Point to L		Memo:		
DB6	Fix error	**6A:** 0 1 9	**B**	**C**	
	Correct: Need F before R		Memo:		
DB7	Show error	**7A:** 0 1 9	**B**	**C**	
	Correct: Needs F at the end		Memo:		
DC 1	Identify part of path	**1A:** 0 1 9			
	Correct: F		Memo:		
DC2	Break down path-first move	**2A:** 0 1 9			
	Correct: first F		Memo:		
DC3	Break down path-last move	**3A:** 0 1 9			
	Correct: last F		Memo:		
DC4	One designation, multiple paths	**4A:** 0 1 9			
	Correct: any path with no diagonals		Memo:		
DC5	Break down path	**5A:** 0 1 9			
	Correct: any path with no diagonals that transects apple, ends in watermelon		Memo:		

Hillcrest Midpoint 2_20200130

Figure B.5

Assessment Scoring Sheet 5

A Codes
0 Incorrect
1 Correct
9 No response

B Codes
1 Moves object one square at a time, matching arrows move-by-move
2 Turn omits forward arrow
3 Does not go forward far enough to reach destination
4 Mixes up left/right arrows
5 Mixes up forward/back arrows
6 Places arrows on grid to match movements
7 Places arrows on grid, turning them in the direction of movement
8 Strategy not observed

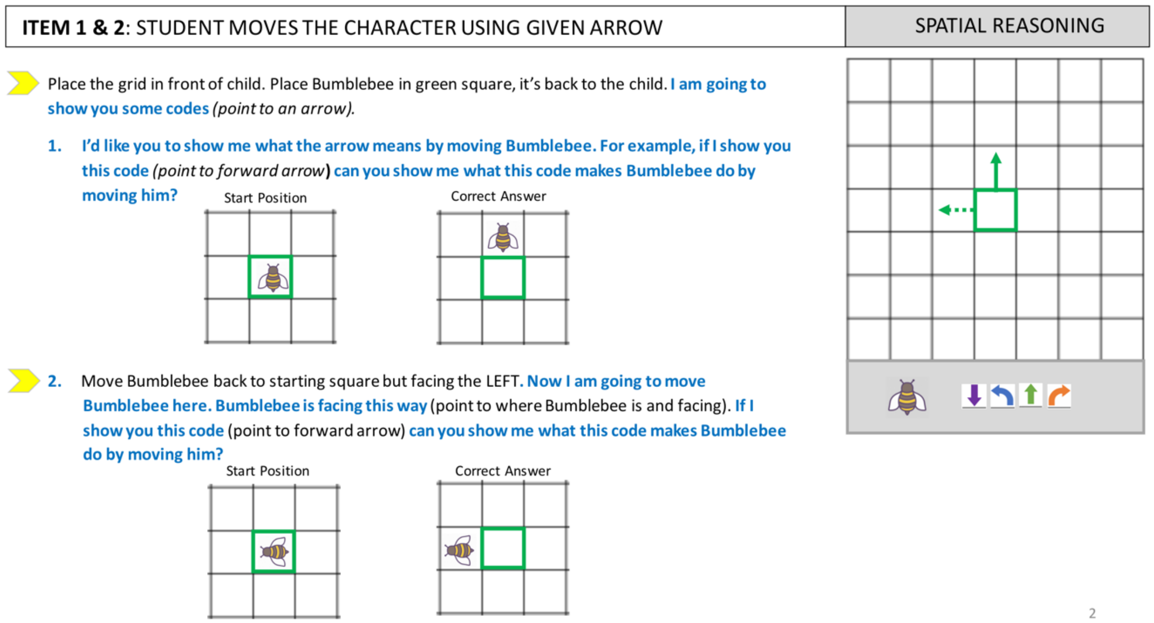
C Codes (for DB/AT items)
1 Add on
2 Wipe and start over
3 Pinpoint bug and replace
4 Replace end piece
5 Remove end piece

Appendix C

Assessment Items

Figure C.1

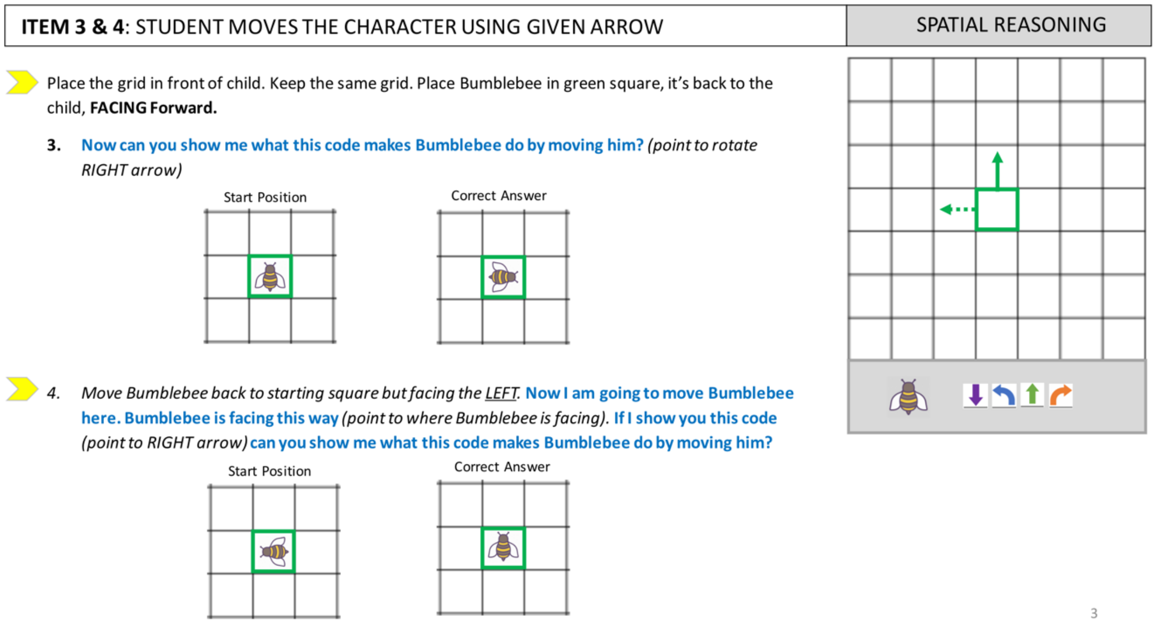
Items CP1 and CP2



Note. Item *1* in this image depicts *CP1*. Item *2* depicts *CP2*.

Figure C.2

Items CP3 and CP4



ITEM 3 & 4: STUDENT MOVES THE CHARACTER USING GIVEN ARROW — SPATIAL REASONING

Place the grid in front of child. Keep the same grid. Place Bumblebee in green square, it's back to the child, **FACING Forward.**

3. **Now can you show me what this code makes Bumblebee do by moving him?** *(point to rotate RIGHT arrow)*

Start Position — Correct Answer

4. *Move Bumblebee back to starting square but facing the LEFT.* **Now I am going to move Bumblebee here. Bumblebee is facing this way** *(point to where Bumblebee is facing).* **If I show you this code** *(point to RIGHT arrow)* **can you show me what this code makes Bumblebee do by moving him?**

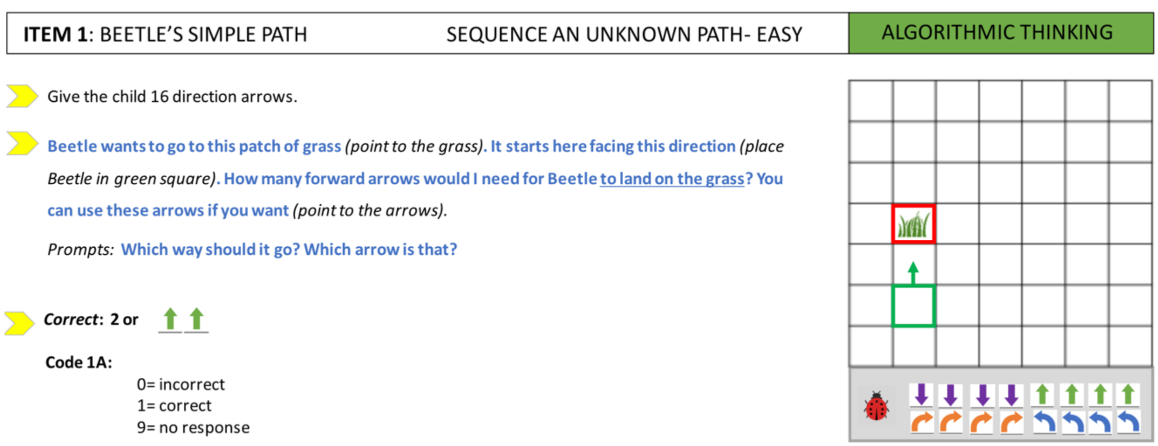
Start Position — Correct Answer

3

Note. Item *3* in this image depicts *CP3.* Item *3* depicts *CP4*.

Figure C.3

Item CT1



ITEM 1: BEETLE'S SIMPLE PATH — SEQUENCE AN UNKNOWN PATH- EASY — ALGORITHMIC THINKING

Give the child 16 direction arrows.

Beetle wants to go to this patch of grass *(point to the grass).* **It starts here facing this direction** *(place Beetle in green square).* **How many forward arrows would I need for Beetle to land on the grass? You can use these arrows if you want** *(point to the arrows).*

Prompts: **Which way should it go? Which arrow is that?**

Correct: 2 or

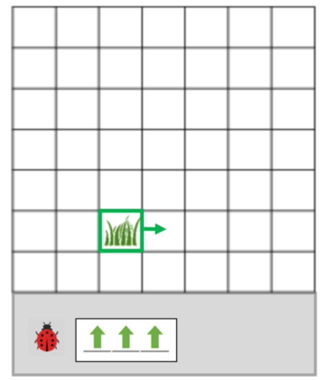
Code 1A:

0= incorrect
1= correct
9= no response

Figure C.4

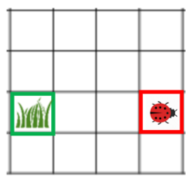
Item CT2

ITEM 2: BEETLES'S SIMPLE PATH	MOVE A KNOWN PATH- EASY	ALGORITHMIC THINKING



- Place the grid in front of child.
- Beetle is sitting here on this grass *(point to the grass)*. Facing this direction *(put beetle down)*. Beetle used these instructions ⬆ ⬆ ⬆ *(show the code strip that has 3 forward arrows)*. Please move Beetle using these instructions *(point to the code strip)*.

  Prompts: Which way should it go?
- ***Correct***:



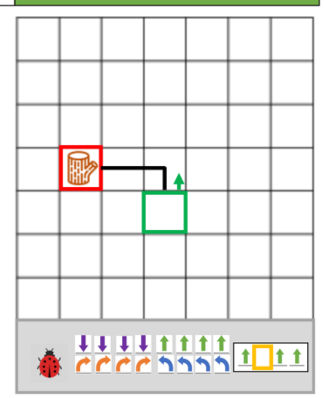
Code 2A:

0= incorrect
1= correct
9= no response

Figure C.5

Item CT3

ITEM 3: BEETLE'S PATH	FILL IN THE TURN- EASY	ALGORITHMIC THINKING



- Place the grid and program organizer in front of child.
- Beetle wants to take this path to the tree stump. *(trace path, starting with Beetle and ending with tree stump)*. It starts here *(place Beetle in the green square)* using these instructions *(place code strip in front of child)*, ⬆ ☐ ⬆ ⬆ but an arrow is missing. Which arrow belongs in this blank square?

  Prompts: Which way does it rotate? Which arrow is that?
- ***Correct***: ↰

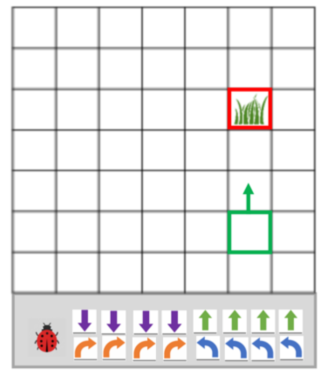
Code 3A:

0= incorrect
1= correct
9= no response

Figure C.6

Item CT4

ITEM 4: BEETLE'S SIMPLE PATH	SEQUENCE AN UNKNOWN PATH- EASY	ALGORITHMIC THINKING



- Place the grid and program organizer in front of child.
- Beetle wants to land on this patch of grass *(point to the grass)*. It starts here facing this direction *(place Beetle in green square)*. Use these arrows to tell Beetle how to land on the grass.

  Prompts: Which way should it go? Which arrow is that?

- ***Correct*:** 

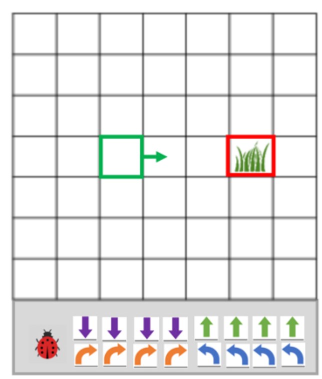
Code 4A:

0= incorrect
1= correct
9= no response

Figure C.7

Item CT5

ITEM 5: BEETLE'S SIMPLE PATH	SEQUENCE AN UNKNOWN PATH- EASY	ALGORITHMIC THINKING



- Place the grid and program organizer in front of child.
- Beetle wants to land on this patch of grass *(point to the grass)*. It starts here facing this direction *(place Beetle in green square)*. Use these arrows to tell Beetle how to land on the grass.

  Prompts: Which way should it go? Which arrow is that?

- ***Correct*:** 

Code 5A:

0= incorrect
1= correct
9= no response

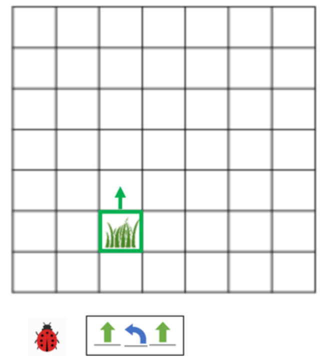
Figure C.8

Item CT6

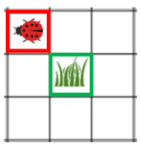
ITEM 6: BEETLES'S PATH EXECUTE A SEQUENCE- MEDIUM	ALGORITHMIC THINKING

Place the grid in front of child.

Beetle starts here on this patch of grass *(point to the grass)*. **It starts here facing this direction** *(place Beetle in green square)*. **Please move Beetle using these instructions**  *(show them the code strip)*. *Prompts:* **Please move Beetle using the instructions.**



***Correct*:**



Code 6A:

0= incorrect
1= correct
9= no response

Figure C.9

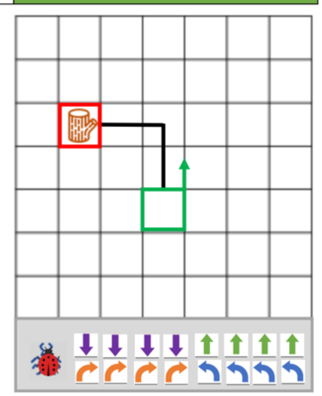
Item CT7

ITEM 8: BEETLE'S PATH SEQUENCE A KNOWN PATH	ALGORITHMIC THINKING

Place the grid and program organizer in front of child. *Simulate movement along path.*

Beetle wants to get to the tree stump, and needs to take this path. *(trace path, starting with beetle and ending with tree stump)*. **It starts here** *(place beetle in the green square)* **and moves like this** *(move along path one square at a time, and end in the red square)*. **Use the arrows to tell Beetle how to *land on the tree stump*. Use this organizer to build your program.**

Prompts: **Which way does it rotate? Which arrow is that?**



***Correct*:** 

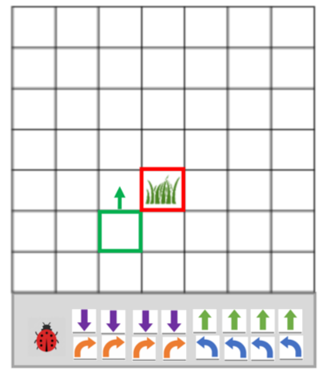
Code 8A:

0= incorrect
1= correct
9= no response

Figure C.10

Item CT8

ITEM 9: BEETLE'S SIMPLE PATH	WHAT PATH?	ALGORITHMIC THINKING



- Place the grid and program organizer in front of child. Give the child 16 directional arrows.
- Beetle wants to land on this patch of grass *(point to the grass)*. It starts here facing this direction *(place Beetle in green square)*. Use these arrows to tell Beetle how to ***end on the grass***.
  Prompts: You can move the Beetle if you like.
- ***Correct***:

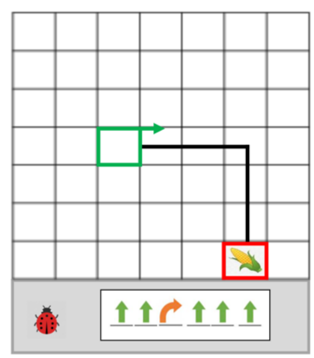
Code 9A:

0= incorrect
1= correct
9= no response

Figure C.11

Item CT9

ITEM 6: BEETLE'S PATH TO CORN	FIX CODING ERROR	DEBUGGING



- Place the grid in front of child. Place beetle in green square, facing direction of arrow.
- Beetle wants to take this path to land on the corn *(trace path from start to end)*. He has these instructions  but there is something wrong with them *(place code strip under grid below grid)*. Use these arrows *(hand them the arrows)* to make instructions that work.
  Prompts: Can you fix the program? Rewrite it to get Beetle there.
- ***Correct***: Needs a  before the 

Code 6:

0= incorrect
1= correct
9= no response

Figure C.12

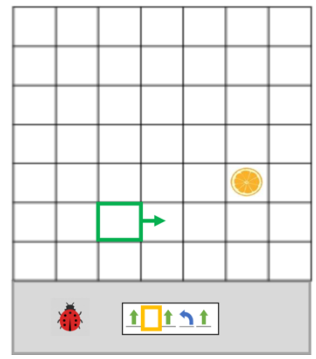
Item CT10

ITEM 1: BEETLE'S MISSING MOVE	IDENTIFYING PART OF PATH	DECOMPOSITION

Place the grid in front of child. Place Caterpillar in green square, facing direction of arrow.

Beetle followed these instructions ↑ □ ↑ ↰ ↑ *(place code strip under grid)* **to get to the orange** *(point to orange).* **But, oops, he missed one move. Which arrow does he need to put in the yellow box to complete his instructions?**

Correct**:** ↑



Code 1:

0= incorrect
1= correct
9= no response

Appendix D

Co-Occurrence Models

Figure D.1

CT1 Co-Occurrence Model

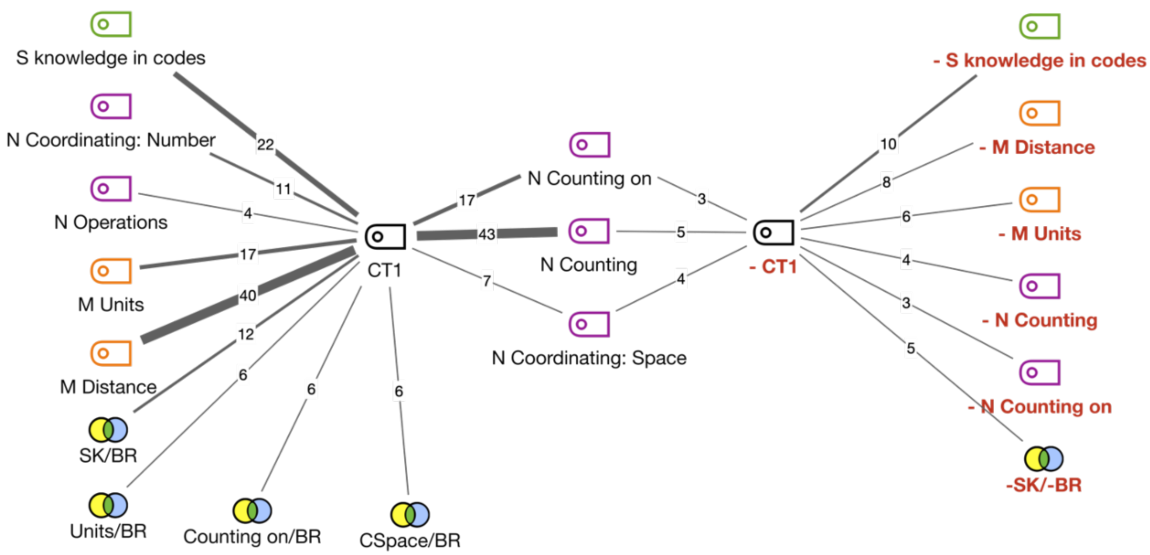


Figure D.2

CT2 Co-Occurrence Model

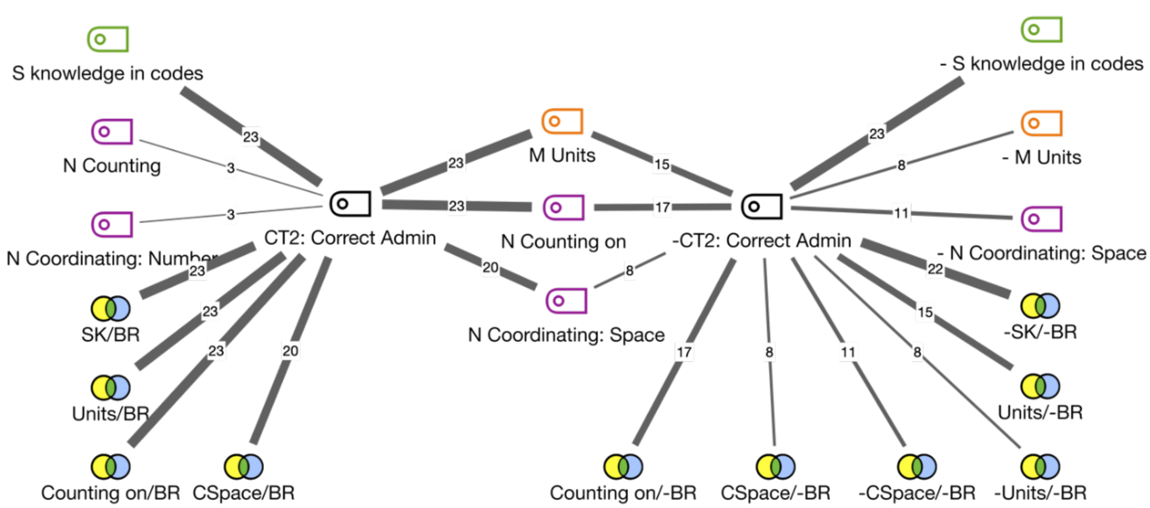


Figure D.3

CT3 Co-Occurrence Model

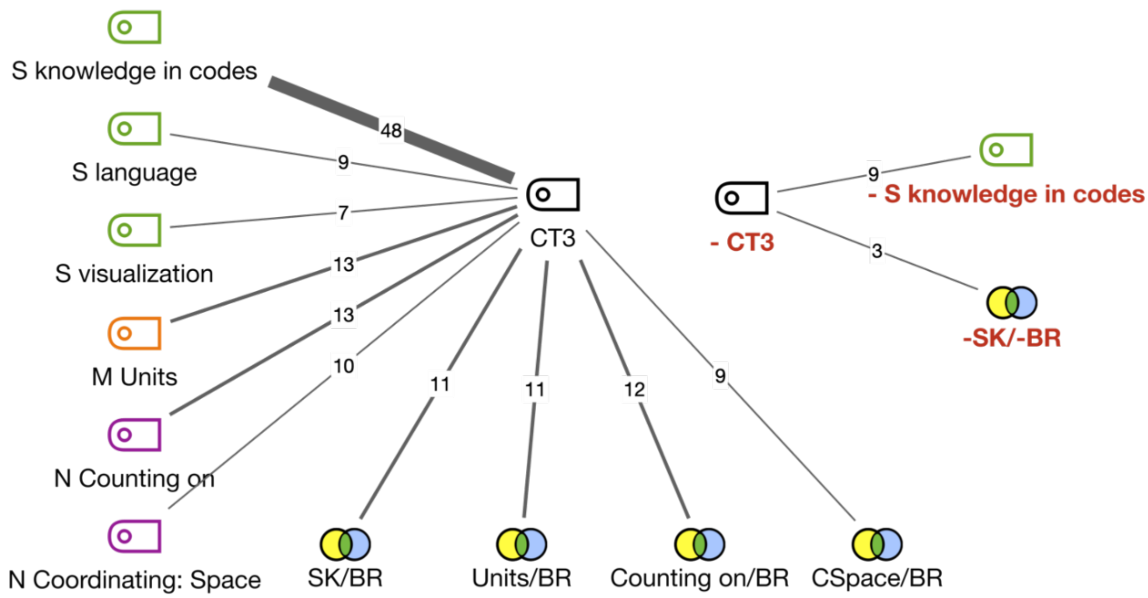


Figure D.4

CT4 Co-Occurrence Model

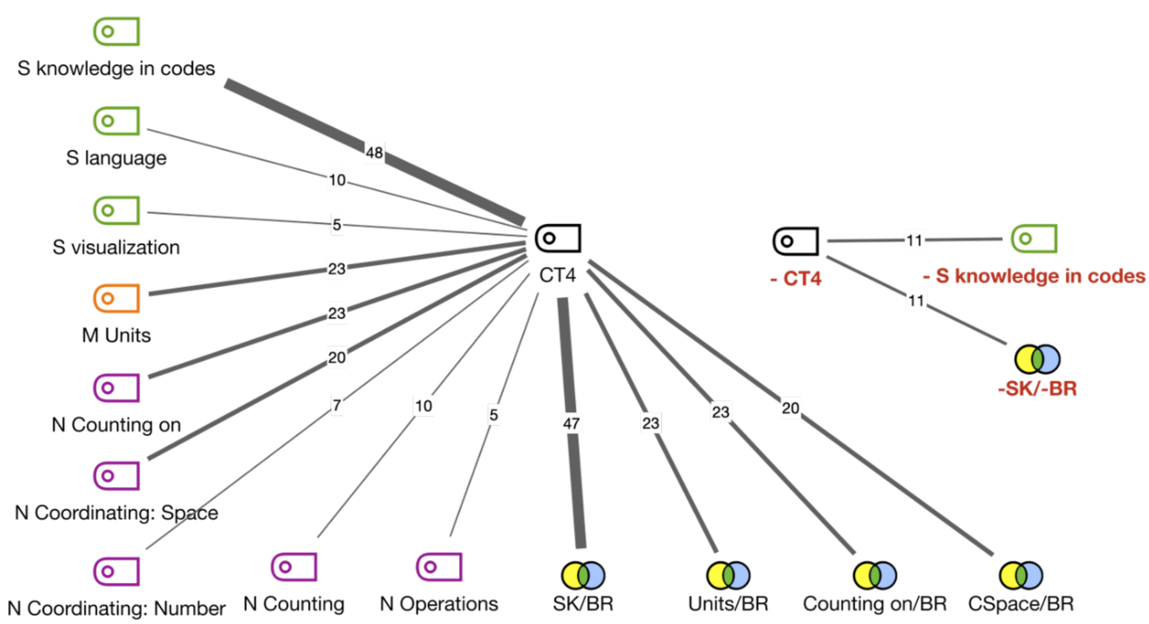


Figure D.5

CT5 Co-Occurrence Model

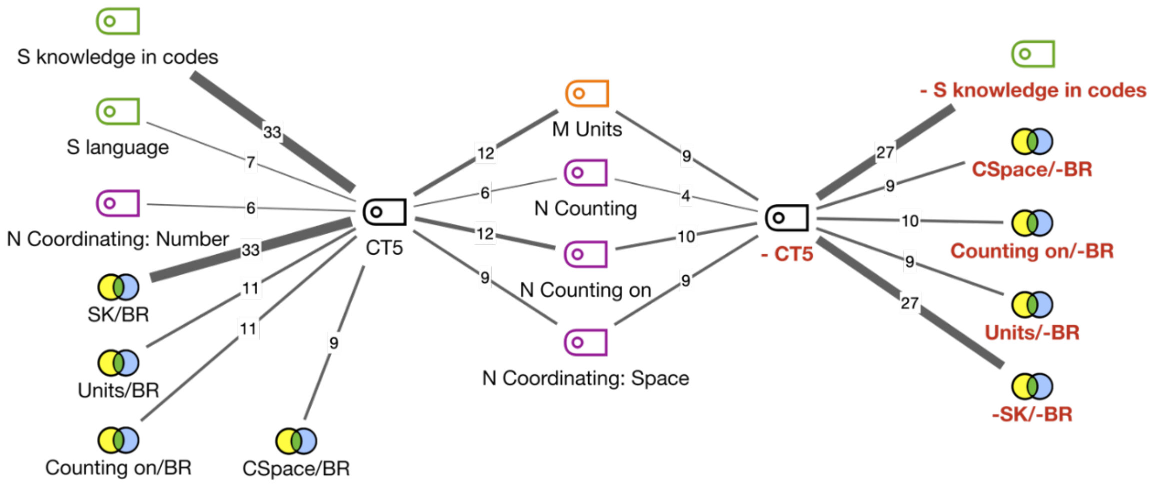


Figure D.6

CT6 Co-Occurrence Model

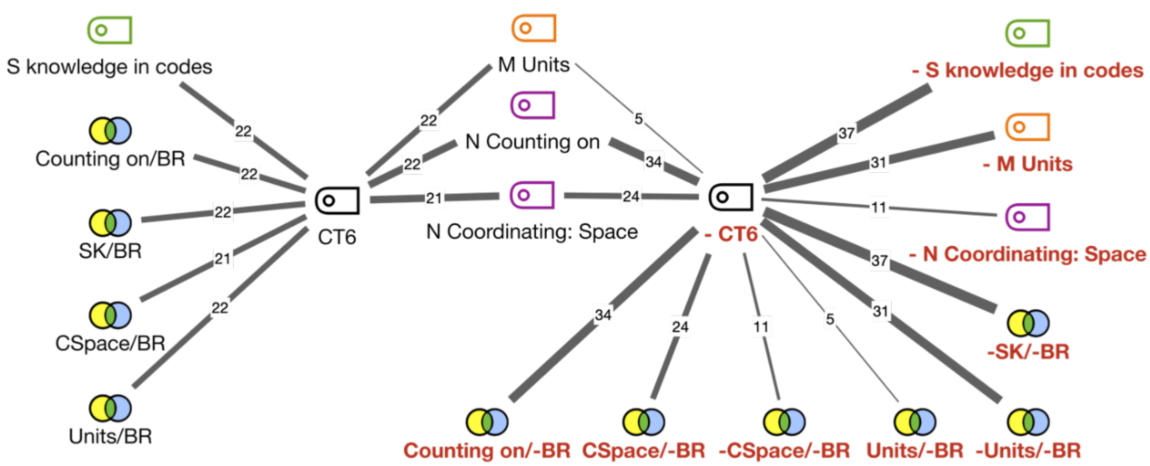


Figure D.7

CT7 Co-Occurrence Model

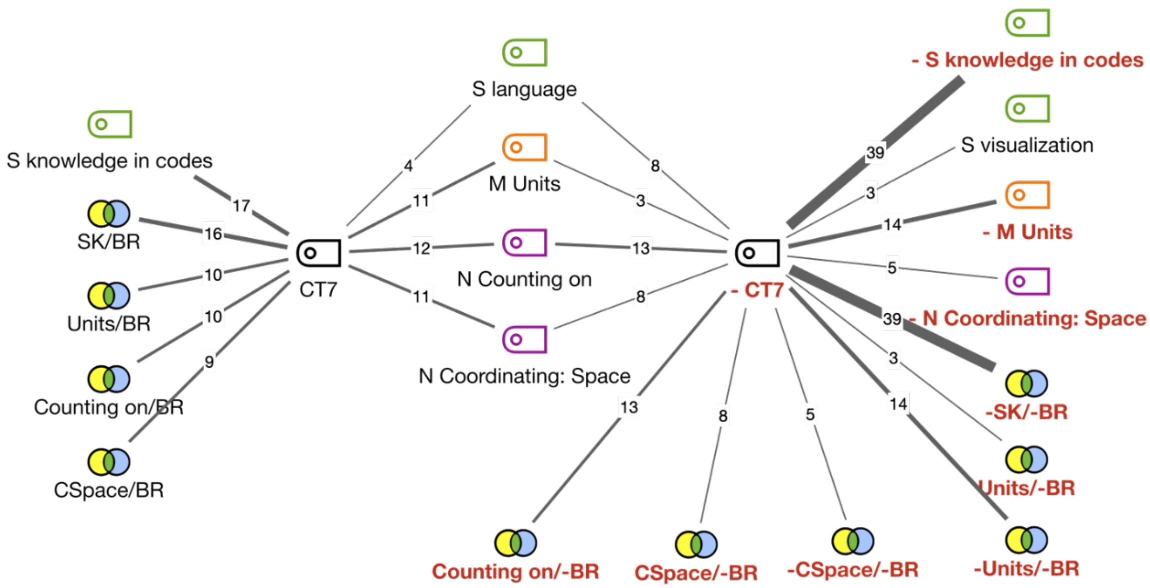


Figure D.8

CT8 Co-Occurrence Model

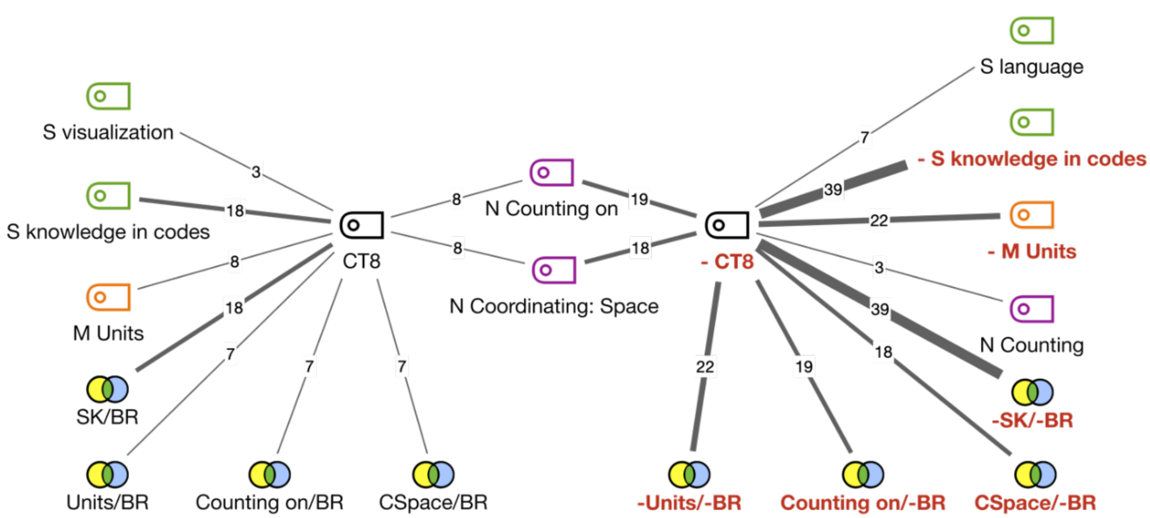


Figure D.9

CT9 Co-Occurrence Model

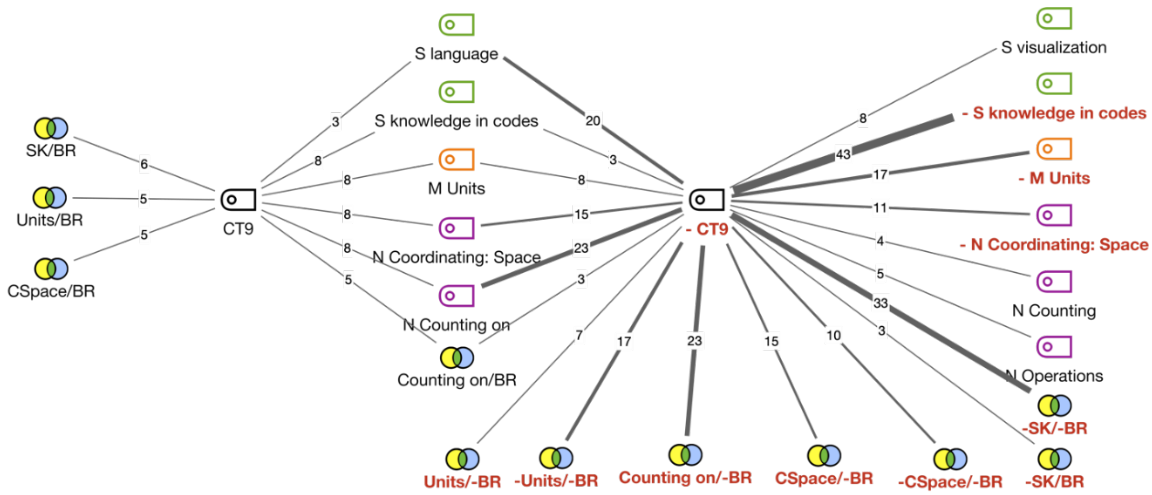
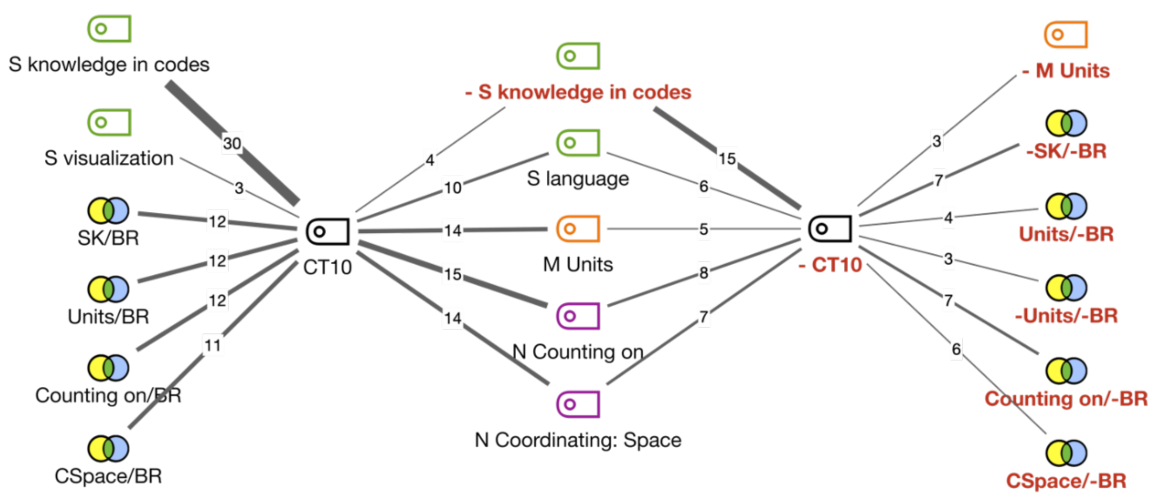


Figure D.10

CT10 Co-Occurrence Model



Appendix E

Mathematical Knowledge Computational Thinking Co-Occurrence Table

Table E.1

Mathematical Knowledge Computational Thinking Co-Occurrence

	CT codes											
MK codes	Plans program	- PP	Builds an intended algorithm	- BIA	Read/enact program	- REP	Recognize bug	- RB	Fix bug	- FB	Decomposition	- DC
Spatial visualization	1	0	10	12	9	6	7	2	6	1	6	2
- SV	0	0	0	1	0	0	0	0	0	0	0	1
Spatial language	2	0	24	37	31	30	15	6	13	4	21	4
- SL	0	0	0	3	0	3	0	0	0	0	0	0
Spatial knowledge in codes	4	0	129	5	133	4	21	1	20	1	34	2
- SK	0	0	0	154	3	135	9	13	5	8	16	7
Units of measure	3	0	60	25	133	43	21	4	19	3	36	3
- UM	0	0	0	58	1	87	5	4	3	3	12	5
Distance	0	0	5	1	2	1	2	1	2	0	3	0
- D	0	0	0	2	0	2	0	0	0	0	0	2
Counting	1	0	23	15	21	13	11	2	10	2	16	0
- C	0	0	0	1	0	2	0	0	0	0	0	2
Counting on	3	0	60	71	135	115	25	7	21	6	47	7
- CO	0	0	0	2	0	3	0	1	0	0	0	0

(*table continues*)

MK codes	CT codes											
	Plans program	- PP	Builds an intended algorithm	- BIA	Read/enact program	- REP	Recognize bug	- RB	Fix bug	- FB	Decomposition	- DC
Coordinating counts: Space	3	0	54	55	119	76	18	4	15	5	41	4
- CC:S	0	0	1	19	2	39	4	4	3	1	2	4
Coordinating counts: Number	1	0	14	4	16	67	5	1	4	1	8	0
- CC:N	0	0	0	1	0	2	0	0	1	0	1	0
Operations	1	0	9	9	7	3	10	0	9	1	6	2
- O	0	0	0	0	0	0	0	0	0	0	0	0

Note. N = 822 item enactments. The table's left column lists the MK codes, and the top row lists the CT codes. As described in Chapter III, the MK and CT codes include codes to identify specific knowledge as indicated or developing. Indicated (or existing) knowledge codes are spelled out (i.e., *Spatial knowledge in codes, Read/enact program*). Codes to indicate knowledge in development are abbreviated with a dash in front (i.e., - *SK, - REP*). For example, in an instance where a student correctly counted on but did not correctly build an intended algorithm, the student's enactment would be coded as *Counting on*, and - *BIP*.

CURRICULUM VITAE

LISE E. WELCH BOND

1556 N 225 W Harrisville, UT 84404
(801) 815-0829 | lwelchbond@hotmail.com

EDUCATION

Ph.D. **Education**
Specialization in Curriculum and Instruction
Concentration in Mathematics Education and Leadership
Utah State University, Logan, UT
Dissertation: *Connections Between Mathematics and Computational Thinking: Kindergarten Students' Demonstration of Mathematics Knowledge in a Computational Thinking Assessment*
(Chair: Jessica F. Shumway)
June 2023, Utah State University, Logan, UT

M.S. **Instructional Technology and Learning Sciences**
December 2014, Utah State University, Logan, UT

B.S. **Elementary Education**
December 2007, Utah State University, Logan, UT

CERTIFICATIONS

Research-Based Early Maths Assessment (REMA) administration (2019–2020)
Elementary Mathematics Endorsement, (2013)
Utah Elementary Education Teaching License (1–8), Level 2 (2008–Present)

PROFESSIONAL EXPERIENCE

Ogden School District
August 2022–Present

LEA Elementary Mathematics Specialist
Guide and support Ogden School District administration, instructional coaches, and teachers in mathematics instruction policy and practice for grades K–5 at 12 elementary schools, 11 of which are Title I. Promote equitable teaching practices to support all learners within a highly diverse school district. Build leadership capacity in teachers and teacher leaders.

Utah State University
Spring 2023

University Instructor of Record – ELED 4062/4056
Taught a math methods course to build prospective teachers' pedagogical content knowledge of rational numbers. Provide regular and meaningful feedback to individual students to support their learning. Support students during a 3-week practicum. Utilize Canvas and other technological tools to successfully facilitate blended course.

Utah State University
Spring 2023

University Instructor of Record – ELED 4061
Taught a math methods course to build prospective teachers' pedagogical content knowledge of rational numbers. Provide regular and meaningful feedback to individual students to support their learning. Utilize Canvas and other technological tools to successfully facilitate an asynchronous course.

Utah State University
January 2019–May 2022

Graduate Research Assistant, NSF STEM+C Funded Project
Primary Investigators: Drs. Jody Clarke-Midura, Jessica Shumway, Victor Lee ($1,120,807). Coding in Kindergarten (Research on the Development of An Assessment to Measure Kindergarten Children's Abilities to Reason Computationally with Mathematical Problem-Solving Skills)
Project Purpose: Create and test mathematics and computational thinking (CT) curriculum and assessment for Kindergarten classrooms.
My Role: Co-develop curriculum, co-develop a CT assessment, data collection, code video data, quantitatively analyze data, read and write about current literature, mentor undergraduate research assistants, participate in team papers and presentations, and participate in research meetings with an interdisciplinary team.

Utah State University
Fall 2021

University Instructor of Record – ELED 1010
Taught a foundations course for the elementary education program via web broadcast and utilizing Canvas, a university LMS system. Utilize interactive technologies to engage a class of 35 prospective teachers and provide regular, individual feedback for each student.

Utah State Board of Education (USBE)
2012–2021

Contracted Course Designer and Facilitator
Developed and taught online USBE mathematics education courses for elementary mathematics teachers (various semesters 2015–2021). Co-developed and taught in-person professional development for elementary mathematics teachers (summers, 2012–2014).

Davis School District August 2018–July 2019	**Sixth Grade Teacher** Taught sixth-grade math and science at Odyssey Elementary, Kaysville, UT. Incorporated technology to enhance science and mathematics instruction using a learning management system (Canvas) and online resources.
Davis School District January 2014–August 2018	**Curriculum Development Specialist** Co-authored a comprehensive research-based, digital mathematics curriculum for Davis School District, Farmington, UT, in grades 2–6. Developed and facilitated 32 professional development sessions for Davis School District teachers in grades 2–6.
Davis School District July 2013–June 2014	**Instructional Math Coach** Used a coaching model with teachers to reflect upon and improve their mathematics instruction. Developed and facilitated professional learning opportunities for teachers. Supported teachers in becoming mathematics leaders within their schools and communities. Helped facilitate a yearly, district-wide Math-Science Olympiad.
Davis School District January 2012–June 2013	**Second Grade District Mentor** Mentored 16 new second grade teachers. Held regular meetings with the teachers to support effective pedagogical practices and provide additional support.
Davis School District January 2010–June 2011	**Second Grade Curriculum Writer** Wrote second grade math lessons to align to new district standards. Worked with a team to design and organize second grade lessons into a complete district curriculum.
Davis School District July 2008–June 2013	**Second Grade Teacher** Taught second grade at Eagle Bay Elementary, Farmington, UT (2008–2010) and Endeavour Elementary, Kaysville, UT (2010–2013). Organized school math nights.
Granite School District January 2008–June 2008	**Kindergarten Teacher** Taught kindergarten at Twin Peaks Elementary, Murray, UT.

RESEARCH INTERESTS

- Children's multi-modal demonstrations of mathematics knowledge
- Connections between mathematics knowledge and computational thinking
- Using coding tools to integrate mathematics and computational thinking
- Early childhood spatial reasoning

SCHOLARLY PUBLICATIONS

PEER-REVIEWED JOURNAL ARTICLES

Welch, L. E., Shumway, J. F., Clarke-Midura, J., & Lee, V. R. (2022). Exploring measurement through coding: Children's conceptions of a dynamic linear unit with robot coding toys. *Education Sciences*, *12*(2), 143. https://doi.org/10.3390/educsci12020143

Shumway, J. F., **Welch, L. E**., Kozlowski, J., Clarke-Midura, J., & Lee, V. R. (2021). Kindergarten students' mathematics knowledge at work: The mathematics for programming robot toys. *Mathematical Thinking and Learning*. https://doi-org.dist.lib.usu.edu/10.1080/10986065.2021.1982666

PRACTIONER JOURNALS

Welch, L. E., Shumway, J. F., Clarke-Midura, J., & Lee, V. R. (2021). Using coding toys to understand equality. *Australian Primary Mathematics Classroom*, *26*(3), 21-25.

PEER-REVIEWED CONFERENCE PROCEEDINGS

Welch, L. E., Silvis, D., Clarke-Midura, J., Shumway, J. F., & Lee, V. R. (2022, June 6–10). *Assessment Designs that Elicit Multimodal Strategies: What We Can Learn about Early Childhood CT by Design*. [Paper Session]. International Conference of Learning Sciences (ICLS), Hiroshima, Japan and Virtual.

BOOK CHAPTERS

Shumway, J. F., Clarke-Midura, J., Lee, V. R., Silvis, D., **Welch, L. E.**, & Kozlowski, J. S. (accepted). Teaching coding in kindergarten: Supporting students' activity with robot coding toys. In Eds. A. Fluck & T. Keane, *Teaching coding K-12*. Springer.

TEXTBOOKS

McInelly, A., **Welch, L. E.,** Stevenson, C., Erickson, M., & Brown, C. (2017). *Advantage Math, Grades 2–6*. Mathematics digital textbook. Davis School District.

NATIONAL CONFERENCE PRESENTATIONS

Welch, L. E., Kozlowski, J., Silvis, D. A., Clarke-Midura, J. E., Shumway, J. F., & Lee, V. R. (2022, April 21–26). *Identifying Kindergarten Students' Strategies as they Solve Computational Thinking Performance Assessment Tasks.* [Paper Session]. American Educational Research Association Annual Meeting, San Diego, CA.

Welch, L. E., Shumway, J. F., Clarke-Midura, J., & Lee, V. R. (2021, April). *Kindergarteners' Conceptions of a Dynamic Linear Unit with Robot Toys.* [Paper Roundtable Session]. Annual Meeting of the American Educational Research Association (AERA), Virtual Platforms and Online.

Shumway, J. F., Clarke-Midura, J. E., Lee, V. R., **Welch, L. E.**, Kozlowski, J. & Evans, H. (2020, April 17–21) *Identifying the Mathematics in Kindergarteners' Play with Coding Toys.* [Paper Roundtable Session]. Annual Meeting of the American Educational Research Association (AERA), San Francisco, CA http://tinyurl.com/shh4hle (Conference Canceled).

Clarke-Midura, J. E., Kozlowski, J., Shumway, J. F., Evans, H., Lee, V. R. & **Welch, L. E.** (2020, April 17–21) *Perspectives and Shifts of Young Children Playing with Coding Toys.* [Paper Session]. Annual Meeting of the American Educational Research Association (AERA), San Francisco, CA http://tinyurl.com/rhkhvka (Conference Canceled).

Lee, V. R., Clarke-Midura, J. E., Shumway, J. F., Kozlowski, J., **Welch, L. E.** & Evans, H. (2020, April 17–21) *Capturing Kindergarteners' Computational Thinking Through Commercial Toy-Centered Task and Assessment Development.* [Symposium]. Annual Meeting of the American Educational Research Association (AERA), San Francisco, CA http://tinyurl.com/yx2wzh53 (Conference Canceled).

Clarke-Midura, J., Shumway, J. F., **Welch, L. E.,** Kozlowski, J. S., & Evans, H. (2019, November). *Integrated STEM: Coding Toys in Kindergarten Math Class.* Presentation for researchers, School Science and Mathematics Association (SSMA) National Convention, Salt Lake City, Utah.

Shumway, J. F., Clarke-Midura, J., Kozlowski, J. S., **Welch, L. E.**, & Evans, H. (2019, November). *Coding and Math: Playing with Screen-Free Robots to Develop Spatial and Measurement Reasoning.* Presentation for educators and researchers, School Science and Mathematics Association (SSMA) National Convention, Salt Lake City, Utah.

REGIONAL PRESENTATIONS

Welch Bond, L. E., Beck, K., Basham, M., Kozlowski, J., & Shumway, J. F., (2023, January). *Math and Coding Connections in Elementary*. A presentation for K–5 teachers at the Utah Council of Teachers of Mathematics (UCTM) conference, Provo, UT.

Welch, L. E., Childers, K., Shumway, J., Clarke-Midura, J. (2022, February) *Merging Math and Programing with Robots*. A presentation for K–6 teachers at the Utah Council of Teachers of Mathematics conference, Layton, UT.

Welch, L. E. (2020, February). *Finding Common Ground: Integrating Computational Thinking with Mathematics Instruction*. A webinar presentation for K–12 Utah mathematics educators. Utah Council of Teachers of Mathematics.

Kozlowski, J. S., **Welch, L. E.,** & Evans, H. Mentors: Shumway, J. F., Clarke-Midura, J., & Lee, V. R. (2019, April). *An Exploration of Kindergarten Students' Use of Perspective and Computational Thinking.* Presentation, USU Student Research Symposium, Logan, Utah.

Welch, L. E., Kozlowski, J. S., & Evans, H. Mentors: Shumway, J. F., Clarke-Midura, J., & Lee, V. R. (2019, April). *Coding to Develop Early Mathematical and Computational Thinking in Kindergarten: A Case Study*. Presentation, USU Student Research Symposium, Logan, UT.

Welch, L. E., Erickson, M. (2017, August) *Utilizing Technology to Support Effective Mathematics Teaching Practices*. A presentation for K–6 teachers at the Utah Council of Teachers of Mathematics conference, Ogden, UT.

Welch, L. E. (2013, August). *Using Formative Assessment in Mathematics*. A presentation for K–6 teachers at the Northern Utah Curriculum Council Conference, Layton, UT.

SCHOOL DISTRICT PRESENTATIONS

Welch, L. E., & Erickson, M. (2017, July). *Utilizing Technology to Support Effective Mathematics Teaching Practices*. A presentation for K–6 teachers at the Davis School District Technology Conference, Farmington, UT.

Welch, L. E. (2014, July). *Using Technology to Math Journal*. A presentation for K–6 teachers at the Davis School District Technology Conference, Farmington, UT.

Welch, L. E. (2010, February). *Using Math Centers for Differentiation*. A workshop for the Davis School District Professional Day, Farmington, UT.

UNIVERSITY TEACHING EXPERIENCE

University Instructor of Record, ELED 4062/4056 Teaching Elementary School Mathematics II: Number, Operations, & Algebraic Reasoning (2023, Spring). Utah State University, Broadcast.
Description: Students in this course develop pedagogical content knowledge in number, operations, and algebraic reasoning for teaching grades Preschool to Grade 6. Students also learn about methods for designing and implementing mathematics instruction, assessment, remediation, and intervention.

University Instructor of Record, ELED 4061 Teaching Elementary Mathematics I: Rational Numbers, Operations, and Proportional Reasoning (2022, Fall). Utah State University, Asynchronous.
Description: Students in this course develop pedagogical content knowledge of rational numbers, operations, and proportional reasoning in preparation to teach grades Preschool–6. Students also build an understanding of the characteristics of effective instruction, assessment, remediation, and intervention.

University Instructor of Record, ELED 1010 Orientation to Elementary Education (2021, Fall). Utah State University, Web Broadcast.
Description: Students assess themselves as prospective teachers by examining the attributes of effective teachers, practicing classroom relationship skills, and learning about the conditions and stresses of teaching while becoming familiar with Utah State's education program and the application process.

Guest Lecture, TEAL 7551 Mathematics Education Research Foundations (for Jessica Shumway) (2021, April). Utah State University, Web broadcast.

Guest Lecture, ELED 4062 Teaching Elementary School Mathematics II: Number, Operations, & Algebraic Reasoning (for Jessica Shumway) (2021, April). Utah State University, Web broadcast.

Guest Lecture, ELED 4062 Teaching Elementary School Mathematics II: Number, Operations, & Algebraic Reasoning (for Beth MacDonald (2021, April). Utah State University, Web broadcast.

Guest Lecture, ELED 4062 Teaching Elementary School Mathematics II: Number, Operations, & Algebraic Reasoning (for Jessica Shumway) (2020, December). Utah State University, Web broadcast.

Guest Lecture, ELED 4062 Teaching Elementary School Mathematics II: Number, Operations, & Algebraic Reasoning (for Katherine Vela) (2020, December). Utah State University, Web broadcast.

Guest Lecture, ELED 4062 Teaching Elementary School Mathematics II: Number, Operations, & Algebraic Reasoning (for Jessica Shumway) (2020, February). Utah State University. Face-to-face format for two different class sections.

SERVICE

Reviewer (2021). Research Conference Proposals, American Education Research Association.

Reviewer (2021). Research Conference Proposals, ACM Interaction Design and Children.

Paper Session Chair (2021). American Education Research Association Conference, Virtual.

Judge (2020), Student Research Symposium. Utah State University

Reviewer (2020). Article Submissions, *Mathematics Teacher: Learning and Teaching PreK–12*

Board Member (2017–2021). Utah Council of Teachers of Mathematics.

PROFESSIONAL COMMITTEES

Utah Quest Assessment Item Writing Committee (2019). Sixth Grade, Utah State Board of Education.

Elementary Language Arts Textbook Review Committee (2018). Utah State Board of Education.

Utah SAGE Rubric Verification Committee (2017). Fourth Grade, Utah State Board of Education.

Utah SAGE Assessment Item Writing Committee (2017). Fourth Grade, Utah State Board of Education.

Elementary Mathematics Curriculum Guide Writing and Revision (2016–2017). Utah State Board of Education.

K–12 OER Collaborative Rapid Prototype Evaluation (2015). Achieve.org.

Utah Mathematics Core Standards Revision Committee (2015). 2nd Grade, Utah State Board of Education.

Utah Textbook Review Committee (2013, 2017). Utah State Board of Education.

PROFESSIONAL AFFILIATIONS

American Education Research Association (2019–Present)
National Council of Teachers of Mathematics (2013–Present)
School Science and Mathematics Association (2019–2021)
National Council of Supervisors of Mathematics (2017–2019)