

Elementary Computational Thinking Instruction and Assessment: A Learning Trajectory Perspective

FEIYA LUO, University of Alabama

MAYA ISRAEL, University of Florida

BRIAN GANE, University of Illinois at Chicago

There is little empirical research related to how elementary students develop computational thinking (CT) and how they apply CT in problem-solving. To address this gap in knowledge, this study made use of learning trajectories (LTs; hypothesized learning goals, progressions, and activities) in CT concept areas such as sequence, repetition, conditionals, and decomposition to better understand students' CT. This study implemented eight math-CT integrated lessons aligned to U.S. national mathematics education standards and the LTs with third- and fourth-grade students. This basic interpretive qualitative study aimed at gaining a deeper understanding of elementary students' CT by having students express and articulate their CT in cognitive interviews. Participants' ($n = 22$) CT articulation was examined using *a priori* codes translated verbatim from the learning goals in the LTs and was mapped to the learning goals in the LTs. Results revealed a range of students' CT in problem-solving, such as using precise and complete problem-solving instructions, recognizing repeating patterns, and decomposing arithmetic problems. By collecting empirical data on how students expressed and articulated their CT, this study makes theoretical contributions by generating initial empirical evidence to support the hypothesized learning goals and progressions in the LTs. This article also discusses the implications for integrated CT instruction and assessments at the elementary level.

CCS Concepts: • **Social and professional topics** → **Computational thinking**; **K-12 education**;

Additional Key Words and Phrases: Computational thinking instruction, learning trajectory, assessment, elementary education

ACM Reference format:

Feiya Luo, Maya Israel, and Brian Gane. 2022. Elementary Computational Thinking Instruction and Assessment: A Learning Trajectory Perspective. *ACM Trans. Comput. Educ.* 22, 2, Article 19 (February 2022), 26 pages.

<https://doi.org/10.1145/3494579>

1 INTRODUCTION

Computer science (CS) education is increasingly gaining attention from policymakers, educational researchers and practitioners, and the general public, such as parents and computing en-

This study is based upon work supported by the National Science Foundation under award No. 1932920.

Authors' addresses: F. Luo (corresponding author), University of Alabama, 301 Graves Hall, Tuscaloosa, AL, 35401; email: feiya.luo@ua.edu; M. Israel, University of Florida, 2-208 Norman Hall, PO Box 117048, Gainesville, FL 32611; email: misrael@coe.ufl.edu; B. Gane, University of Illinois at Chicago, 1240 W. Harrison St., 1570-L SSB, Chicago, IL 60607; email: bgane@uic.edu.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

© 2022 Association for Computing Machinery.

1946-6226/2022/02-ART19 \$15.00

<https://doi.org/10.1145/3494579>

thusiasts. At the national level, the “Computer Science for All” initiative aims at promoting CS education from kindergarten to high school and equipping students with **computational thinking (CT)** skills to thrive in the digital economy. The Computer Science Teachers Association Computer Science Standards [16] provides detailed expectations for student performance and achievements for CS concepts and practices at the elementary and secondary levels. The standards organize each core concept and practice in the K-12 CS Framework [24] into subconcepts and practices. The detailed descriptive statements explain the suggested benchmarks of students’ CS understanding and skills by each level (grade bands). For example, by the end of grade 2, students can model daily processes, such as teeth-brushing and cooking, by following a step-by-step procedure (i.e., an algorithm); by the end of grade 5, students know how to select from multiple algorithms the most appropriate to finish a task; and so on. Thus, although CS and CT are rather new in the elementary grades, there is growing consensus about which computational concepts should be taught in these grade levels.

In elementary school settings, CS education usually takes the form of promoting students’ CT in preparation for more advanced CS knowledge and skill development [8, 20]. CT is one of the Disciplinary Core Ideas for integrated K-12 science, technology, engineering, and mathematics (STEM) instruction in the Next Generation Science Standards [31]. Therefore, CT is often integrated into existing subject areas, such as mathematics and science [27, 28, 38, 42]. One of the biggest advantages of CT integration is that it ensures equity by providing CS education in subject areas that are already taught to students [49]. Compared to CS education as electives or enrichment activities available to only the few who can afford to participate, such integrated learning opportunities are accessible and available to all students.

Previous research [26] has operationalized the definition of CT to include abstraction (i.e., selecting certain features to model with a computing device), automation (i.e., instructing the computer to efficiently and quickly execute as demanded by human instructions), and analysis (i.e., reflecting and validating actions and decisions). The three dimensions of CT [10] proposed that CT involved computational concepts (basic CS concepts such as sequence, loops, and conditionals), computational practices (iteration, debugging, remixing, and abstraction), and computational perspectives (expressing, connecting, and questioning in programming). In the later research, CT was synthesized by K-12 CS Framework [24] to be “the thought processes involved in expressing solutions as computational steps or algorithms that can be carried out by a computer” (p. 68) and that it “requires understanding the capabilities of computers, formulating problems to be addressed by a computer, and designing algorithms that a computer can execute” (p. 69). This study synthesized the definitions of CT to refer to the thought processes that students have in constructing solutions that can be executed by a computer by applying various computational concepts and practices in problem-solving. As such, this study used cognitive interviews to reveal elementary students’ thought processes while applying CT in problem-solving.

2 PREVIOUS RESEARCH ON ELEMENTARY STUDENTS’ CT

2.1 Elementary CT Instruction

CT instruction in elementary grades often takes form as plugged and unplugged lessons and activities. Plugged activities require the use of a computer or a mobile device, often happening on visual block-based coding platforms such as Scratch and Code.org CS Fundamentals, or with robotics toolkits. CT instruction with plugged lessons engages students in programming or robotics activities while reinforcing knowledge of the target subject [8, 27]. Many studies [6–8] leveraged visual programming tools, such as Alice and programming-infused robotics curricula, and improved students’ programming knowledge. For example, children (between 5 and 6 years old) benefited

from working with the Bee-bot robotic devices, improving their CT by writing and organizing commands to control the tangible robots and applying decomposition (i.e., breaking up a problem into subparts) in problem-solving [1]. Kindergarten children can practice path designing, counting, and measuring skills by playing programming games such as the Ladybug Leaf and the Ladybug Maze. In these games, children could control the movements of the ladybug to reach the leaf or go through the maze on the screen by using directional and turn commands, such as forward, backward, and turn by 90 degrees [20]. A commonly used coding platform for elementary CT instruction, Scratch allows students to learn CT through programming history-related games [48] and to improve understanding of computational concepts and practices including sequence, iteration (looping), conditional statements, threads (parallel execution), event handling, user interface design, and keyboard input [42].

Unplugged activities, however, do not require working on a computer. Rather, they incorporate CT concepts in paper-and-pencil forms or physical acting. An example is Code.org's My Robotic Friend lesson (<https://curriculum.code.org/csf-1718/courseb/6/>) that asks students to design and draw algorithms to stack cups in different patterns. There are a few benefits of using unplugged activities to teach CT. Such activities are "kinesthetic, engaging, and accessible" (p. 501) [41] and provide opportunities for teachers to present, and students to learn, key CT concepts when access to computers or mobile devices is limited or nonexistent [18]. In addition, students can engage with the great ideas in CS without having to develop any programming skills or being distracted by technical issues while working on a computer, such as installing software [5].

Although previous research has explored CT instruction with young students, systematic implementation of CT instruction at the elementary level is a daunting task involving many challenges [28]. First, allocating CT instruction time within the regular school structure is difficult. Most schools in the elementary level already follow a strict curriculum implementation (i.e., content and pacing) that is difficult to fit in a new subject to be taught. Second, most schools do not have the luxury of hiring CS teachers [26]. Therefore, at the elementary level, CT is often situated in and integrated into existing subject areas (science, math, and so on) rather than being taught as an independent subject [40, 42, 48]. For example, an integrated learning activity can take the form of having students program a robot to tell the life cycles of fern plants (where botany is the target subject [27]) or create a Scratch animation where two sprites compare two fraction numbers (where math is the target subject [46]). Although such an integrated approach helps alleviate the challenge of limited instruction time at schools, other challenges and barriers still apply. For one, CS is a subject in which elementary teachers may not already have expertise [23], and thus **professional development (PD)** is often required. However, depending on the location, there may be a lack of PD opportunities, resources, and access to infrastructure (i.e., robotics kits, PCs, or mobile devices with learning software installed and Internet access) to support CT implementation [26]. Elementary teachers who started to integrate CT in their classrooms often lacked the tools and knowledge to make sense of their students' CT learning outcomes to inform their instruction [23]. As such, research is needed to provide usable instructional and evaluative tools for better teaching and student learning outcomes.

2.2 Elementary CT Assessment

Although there is some research in cultivating CT in elementary school students, it has been acknowledged that assessing elementary school students' CT with developmentally appropriate instruments is a challenging task given students' limited reading and understanding skills at a young age [47]. A recent systematic review of CT assessments [47] synthesized four major assessment types commonly used in the literature, including tests, portfolio analysis, interviews, and surveys. Conducting interviews is identified by researchers as a useful approach for revealing students' CT

understanding and how students apply CT [47]. Interviews can happen in a variety of contexts and formats (i.e., soliciting students' reflections of previously made projects or using think-aloud while problem-solving). For example, during artifact-based interviews [10], students reflect on and explain their artifact-making considerations (how they started a project, how the project evolved during development, what were the problems encountered and solutions used, etc.). Such interviews allowed a lens to assess students' CT and identify potential conceptual gaps. For example, students may use certain code blocks in their projects without being able to fully explain how those code blocks work. However, it was acknowledged that if students were interviewed about a project that was previously created, the result would be limited by what the interviewed student was able to remember at the time of the interview [10]. To observe students practicing CT in real time, recent research reported using interviews synchronously with a think-aloud protocol to have students simultaneously articulate their thought processes when solving a computing task [2, 28].

The second type of assessment identified by Tang et al. [47] in their systematic review is the selected- or constructed-response tests. Such tests usually constitute multiple-choice questions, open-ended questions to summatively evaluate students' CT. An example is the study of Chen et al. [11] where the authors developed and implemented a 6-month robotics curriculum with 125 students in fifth grade. Students used a visual programming platform to write and test their programs before taking turns to test programs on a physical robot. The curriculum involved key CS concepts such as algorithms, variables, conditionals, and loops, among others. To assess students' CT, the authors designed and developed an instrument with 23 items (15 multiple-choice questions and eight open-ended questions) grouped into six sets. The items were designed and developed using a five-component framework that aimed to assess students' CT in terms of syntax, data, algorithms, problem representation, and efficiency.

The different assessment types provide insights into what instruments or tools may be more appropriate given the different purposes of CT assessments. For example, interviews can be used to reveal students' thinking processes in problem-solving and to identify conceptual gaps but are usually time-consuming [10, 47]. In addition, tests can be used to summatively evaluate the students' CT as a learning product but miss the CT in action. As such, combining the two assessment types may potentially address the gap in the lack of substantial research in understanding the process of how students apply CT [47].

2.3 Learning Trajectories as the Theoretical Foundation

Recent research has explored the use of **learning trajectories (LTs)** as the theoretical foundation for systematically implementing and assessing CT at the K-9 level [36, 39]. LT is originally an established construct in mathematics research and practice, and it is commonly acknowledged to contain three components: learning goals, learning progressions, and the instructional activities that support the learning progressions [12, 45]. Learning goals are a collection of landmarks that children are expected to meet as they engage in instruction. These goals are usually defined with a broad agreement, as reflected in previous literature and/or acknowledged standards [12]. Learning progressions refer to hypothetical developmental paths that students take as they progress toward increasingly sophisticated understanding during learning [12, 17]. Learning progressions are hypothetical in nature because such progressions are often conceptualized and created by either synthesizing the literature or working with a particular, usually small, group of students and then applied to a different group [17]. In addition to the learning goals and the learning progressions, a third key component of a hypothetical LT is the instructional activities that help children develop higher levels of thinking while moving along the progression [13, 45]. Barrett and Battista [3] emphasized that such a progression "is tied to, and must interact with, instruction" (p. 102).

Without instruction, stages of development of student understanding and competencies may be obviously different.

The construct of LTs differs from the conventional “scope and sequence” for learning [19]. A scope and sequence is a plan for what learning goals to meet and what instructional activities to be carried out; however, LTs afford empirical evidence that students’ understanding and skills actually develop as learning progressions have hypothesized [14, 15, 43]. Research in mathematics [14, 15] has accrued evidence that highlights the efficacy of the LT approach. In a study that involved teaching shape composition to preschool children, Clements et al. [14] compared an instructional approach that was based on an empirically validated LT with one that focused on one target goal (skipping the intermediate consecutive target levels). The study concluded that after engaging the children in more than eight brief sessions (9 minutes each) over 5 weeks, the children in the LT-based instruction did better in terms of strategy-using and answering completely correctly the assessment items than children taught to one target goal. In addition, children in the LT group expressed less counterproductive frustration than the other group [14].

The research in mathematics LT provides a promising outlook for CS education, especially in exploring how children progress in CT learning and how instruction can be designed to support learning progressions. Rich et al. [36–39] created LTs with hypothesized K-8 students’ learning goals and progressions in the common CT concepts, such as sequence, repetition, conditionals, decomposition, debugging, and variables. The learning goals and progressions in these LTs are not tied to grade levels in that they do not specify which learning goals should be taught at which grade; rather, they provide a general road map of CT learning in the K-8 levels. Each LT details the different levels of expectations for student learning and the relationships among the learning goals for a specific CT concept. For example, the Sequence LT emphasizes students’ understanding and using precise and complete problem-solving instructions in the beginning level, the order of instructions in the intermediate level, and the manipulation of the order in the advanced level. The Repetition LT stresses pattern recognition, constructing instructions using repetition, and controlling the repetition (i.e., how and when to stop a repetition) using different commands in the three levels. The Conditional LT starts with the binary status (true or false) of conditions and evaluation of conditional statements for an intended outcome in the beginning level; in the intermediate level, more complexity is added with multiple conditions using different controls; and finally Boolean variables are included in the advanced level [39]. The Decomposition LT starts with competencies to break down a system and a complex problem into smaller parts and extends into writing, reusing, and repurposing code and procedures [36]. This article does not discuss the debugging and variables LTs, as they are outside the scope of this study.

2.4 Gaps in Current Elementary CT Literature

Research and practice in CT integration at the elementary level are only in the infant stage. Researchers proposed hypothesized LTs with learning goals and progressions to guide K-8 CT instruction [36–39]. However, these LTs have not yet been tested empirically. To know that the LTs can be broadly adopted to guide CT instruction, empirical evidence is needed to reveal what CT competencies students possess and whether students progress toward more advanced learning as hypothesized by the LTs. In addition, existing research on CT has primarily focused on examining the “products” of students’ learning, such as computational artifacts that students built and programs that students wrote [8, 50]. Given that many have assessed CT as a learning product, research that focuses on collecting in-depth, qualitative evidence to understand how students develop and apply CT is still lacking [47]. Therefore, this research study aspires to un-

derstand how elementary students apply CT in problem-solving by uncovering their thought processes.

3 RESEARCH AIM

The purpose of this study was to gain a better understanding of how elementary students apply CT in problem-solving from an LT perspective. This study used a series of integrated math-CT lessons [46] that are tied to the fractions content in the Everyday Mathematics (EM, 4th edition), which is aligned to the Common Core State Standards for Math (CCSS-M). The integrated lessons, called *Action Fractions*, were designed following the learning goals and progressions hypothesized in the CT LTs developed by Rich et al. [36, 39]. The integrated lessons were part of a year-long implementation at an elementary school in the midwestern United States, which was supported by a National Science Foundation-funded grant project, Learning Trajectories for Everyday Computing (award no. 1932920). CT concepts and practices such as sequence, repetition, decomposition, and conditionals are interwoven into the fraction lessons for third- and fourth-grade students. For the purpose of this study, G3 and G4 were used to refer to third and fourth grade, respectively.

Following the implementation pace of the integrated lessons, G3 students participated in eight lessons that focused on sequence, repetition, and decomposition, and G4 students participated in eight lessons that focused on repetition, conditionals, and decomposition. As such, the **research questions (RQs)** that guided this study were as follows:

1. How do G3 and G4 students express and articulate CT in math-CT problem-solving scenarios? Specifically,
 - (1) How do G3 students express and articulate CT in the area of sequence?
 - (2) How do G4 students express and articulate CT in the area of conditionals?
 - (3) How do G3 and G4 students express and articulate CT in the area of repetition?
 - (4) How do G3 and G4 students express and articulate CT in the area of decomposition?
2. How does students' articulated CT correspond to the learning goals and the learning progressions of the CT LTs?

4 RESEARCH METHOD

4.1 Participants

The participants of this study were sampled from four classes (two in G3 and two in G4) at the aforementioned elementary school in the midwestern United States that implemented the integrated lessons. Twelve G3 and 10 G4 students provided parental consent and assent to take part in the study and thus were identified as participants. During the 2019–2020 school year, the four teachers (each leading one of the four classes) implemented the integrated math-CT instruction and assessments during math classes with their students. The selection of student participants had the following criteria: (1) participants needed to be present in most of the integrated lessons implemented; (2) given the nature of cognitive interviews, participants needed to have the ability to verbalize their thinking (the teachers would inform the research team if any of the participants struggled with verbal communication in regular classrooms); and (3) participants needed to provide written consent and assent under the Institutional Review Board-approved protocols. For this study, the detailed demographics of the 10 G4 participants were available (five males, five females; eight White and two Black; three enrolled in special education; and five applied for free lunch at school). The 12 G3 students included both male and female students, representing White, Black, and Asian ethnicities. However, the specific breakdown of demographics was not available due to missing data. According to an official state database, during the 2019–2020 school year, the school had a student population that was 42.3% White, 30.3% Black, 9% Asian, 6.1% Hispanic, and

12.2% of two or more races. The school had a diverse representation of students (i.e., 48.1% from low-income families, 17% enrolled in special education with Individualized Education Programs, and 7.3% English learners). Since 2013, the school has adopted a school-wide CT and CS initiative where the school faculty and community are committed to fostering a CT mindset in students in the K-5 levels. All students learn coding or programming and have access to computers. Given the school's history in CS education participation, in general, students from these classes have had prior CT experience and exposure in all grade levels.

4.2 Instructional Activities: Action Fractions

The Action Fractions lessons implemented with the participants included a combination of plugged lessons (i.e., requiring the use of a computer; highlighting hands-on coding exercises where students use the Scratch programming platform to build projects while learning math and building CT skills) and unplugged lessons (i.e., does not require the use of a computer with discussion prompts and reflections). Prior to teaching the integrated lessons, the teachers participated in PD related to both using the Scratch programming environment and the specifics of the lesson components. Teachers were provided with lesson plans, PowerPoint presentation slides, and instruction-supporting videos developed by the research team. Two researchers from the research team were present during implementation and facilitated the lesson implementation. The researchers were available to answer questions before and during classroom implementation.

In general, a lesson begins with a cover page that starts with the "Math Connections" and the "CT Connections" that introduce the content and activities relevant to math and CT. Then, relevant computing vocabulary is listed. The rest of the cover page includes an at-a-glance plan of the lesson with explicit learning goals (as stated in the relevant LTs and as "I can" statements for students), anticipated barriers for student learning, and options for adapting the lesson to students' preferences. For example, the third-grade lesson, "Fraction Comic Animation," integrated two learning goals pertaining to the Sequence LT and one learning goal relevant to the Decomposition LT. This lesson presents activities for students to compare equivalent fractions by completing an animation in Scratch in steps. Specifically, students are first asked to complete a storyboard organizer to sequence the different scenes involved to compare two fractions. Then, students assemble the instructions to synchronize the actions and scenes in the animation by adding and adjusting the appropriate code blocks. As another example, the fourth-grade lesson, "Comparing Fractions: Slicing Sandwiches," maps to one learning goal from each of the Repetition, Conditionals, and Variables LT. In this lesson, students complete a Scratch project to compare fractions with the same denominator while considering what actions need to be repeated (using the "Repeat" block), how to make the program determine the result of the fractions comparison (using the "If...then" conditional block), and how to represent numerators with variables. The complete set of the Action Fractions lessons and materials, including teacher and student guides, is available on the grant project website (<http://everydaycomputing.org/lessons/action-fractions>). The relevant learning goals are specified in each lesson in the teacher lesson guide. A more detailed description of the lesson sequence and content coverage is published by Strickland et al. [46].

4.3 Assessments

The integrated assessments used in this study were conceptualized and designed from the four CT LTs to assess students' understanding of each of the four CT concepts: sequence, repetition, conditionals, and decomposition as students progressed through the integrated lessons. Given the broad context of the grant project, which integrated CT in elementary math, the paper-based assessment items involved coding scenarios in the Scratch interface (e.g., with code blocks), word problems embedded in the math context, or number problems [21]. The design of these integrated

Table 1. Distribution of Items by CT Concept

	Sequence (G3)	Repetition (G3 & G4)	Conditional (G4)	Decomposition (G3 & G4)	Total
Item ID	S.01.a	R.01.a	C.02.e	DC.02.a	
	S.04.b	R.01.b	C.03.b	DC.02.b	
	S.02.a	R.03.c		DC.06.c	
		R.05.a		DC.08.a	
	3	4	2	4	13

assessments was grounded in the LTs and guided by the evidence-centered design framework. This framework guided the assessment designers to take an argument-based approach to validate assessment items by following established design patterns and collecting empirical evidence of the competencies students articulate or demonstrate [21]. The assessment items were designed to reflect a range of knowledge, skills, and abilities (KSAs) indicated by the learning goals in an LT (i.e., a CT concept). For each grade (G3 and G4), an early, a mid, and a late assessment were designed. During the 2019–2020 implementation year, the assessments were administered at three time points for each grade: late October 2019, February 2020, and May 2020.

This study used items purposefully selected from the assessments. The inclusion criteria included the following: (a) selected items needed to assess CT in the three LTs for each grade as reflected in the RQs (sequence, repetition, and decomposition for G3 and repetition, decomposition, and conditionals for G4); (b) selected items should reflect a wide range of CT and therefore have the capacity to elicit different levels of CT understanding, if applicable; (c) selected items should include ones that used both illustrations of Scratch code and word problems not necessarily associated with code blocks; and (d) true-or-false items were excluded given the binary nature of possible answers. Following the item selection criteria, 13 items mapped to a range of learning goals in the LTs were selected for the cognitive interview, including three assessing sequence, four assessing repetition, two assessing conditionals, and four assessing decomposition (Table 1). Among the 13 items, four were embedded in the Scratch context, six were presented as word problems, and three were as number problems. The item ID starts with an alphabet letter that indicates from which LT (concept) an item assesses knowledge and skills. For example, item S.01.a refers to item 1a that assesses knowledge and skills in the Sequence LT, and item DC.02.a refers to item 2a that assesses knowledge and skills in the Decomposition LT.

4.4 Data Collection

This qualitative study using the cognitive interview technique [33, 34] was categorized as a “basic interpretative study” [30] with a general purpose to understand how students demonstrate and articulate their CT competencies in problem-solving. In this study, cognitive interviews were conducted in two different phases (in the fall and spring semesters, respectively) during the 2019–2020 school year. In each phase, each of the 12 G3 participants was presented with four assessment items assessing sequence, repetition, and decomposition, and each of the 10 G4 participants was presented with three items assessing repetition, conditionals, and decomposition. All interviews were audiotaped and then transcribed. Four researchers including the first and the second authors participated in a training session before conducting the cognitive interviews using the think-aloud protocol (Appendix A) at the research site. The training session included an introduction to the protocol, a think-aloud interview demonstration, and an in-depth discussion on how to adhere to protocol prompts and avoid deviations from the protocol. All four researchers had prior ex-

perience in teaching and interacting with elementary students. Conducting interviews was identified as an appropriate yet under-used approach for revealing students' CT understanding and how students apply CT [47]. An example where such a method was used was in the study of Atmatzidou and Demetriadis [2] where interviews were used synchronously with a think-aloud protocol to have students simultaneously articulate their problem-solving processes while working on a programming task.

4.4.1 Phase 1. In mid-November 2019, phase 1 cognitive interviews took place after students finished four lessons in the integrated math-CT curriculum. Each student participated individually in the cognitive interview wherein they were thinking aloud while solving each of the given assessment items. Each cognitive interview followed the procedures below. First, the researcher/interviewer briefly introduced themselves to the participant. Then, the researcher modeled "thinking aloud" with a sample assessment item by verbalizing the thought processes and rationale for solving the item. Then, the participant was given the paper-based items. For each item, the interviewer reminded the participant to think aloud if there was a period (more than 5 seconds) of silence. After a participant finished an item, the interviewer asked retrospective questions to get a more complete understanding of the student's thinking. The cognitive interview ended after a participant finished all four items and answered the retrospective questions.

4.4.2 Phase 2. Phase 2 interviews took place in late February 2020, after another four lessons were taught for each grade. Similar to the phase 1 data collection, each student participated in a cognitive interview again. Three researchers conducted the interviews, and the interview process followed the same process as in phase 1. Each interview started with a brief introduction of the researcher, followed by a demonstration of the think-aloud activity. Participants then worked on the given assessment items with the researcher making prompts for think-aloud whenever necessary. Retrospective questions were asked at the end of each interview for a more holistic understanding of student thinking. The only change made to the think-aloud protocol was that only one retrospective question (i.e., "What do you think this question is asking you to do?") was asked, as the researchers concurred that the other two retrospective questions did not elicit meaningful data in phase 1.

4.5 Data Analysis

4.5.1 Analytical Processes for RQ1. The data analysis for the first RQ (understanding students' articulated CT in both phase 1 and phase 2 cognitive interviews) followed a two-step process: *a priori* coding and **constant comparison analysis (CCA)** [9]. First, the learning goals in each of the LTs were used verbatim as *a priori* codes for a specific CT concept to categorize students' think-aloud verbalization during problem-solving (e.g., Table 2 lists all learning goals in the Sequence LT as *a priori* codes). In this step, students' CT articulation in each interview was examined against the *a priori* codes. The important questions in this step were as follows: What understanding/proficiencies of this specific CT concept did the participant articulate and demonstrate? What *a priori* codes best describe the characteristics of the participant's articulation in this particular interview? What characteristics did excerpts in this interview have in common? For example, if a participant articulated an ordered, precise, and complete program using the code blocks given in the problem instructions, then *a priori* codes pertaining to "precision and completeness in computer programs" and "order of execution" were assigned to those excerpts.

The second step was the construction of themes using CCA after all interviews were coded using *a priori* codes. Previous research has operationalized CCA to constantly compare data segments within a single unit (e.g., an interview), then across units (e.g., multiple interviews) to put data revealing similar meanings in the same category [9]. In this study, the single unit comparison re-

Table 2. *A Priori* Codes for Sequence

	Learning Goals in the Sequence LT			
Beginning	1. Precise instructions are more likely to produce the intended outcome than general ones.	3.1 Precision and completeness are important when writing instructions in advance.	3.2 Programs are made by assembling instructions from a limited set.	4.1 Computers require precise instructions using limited commands.
Intermediate	2. Different sets of instructions can produce the same outcome.	3. The order in which instructions are carried out can affect the outcome.	4. Computers have a default order of execution, so order matters in programming.	5. Creating working programs requires considering both appropriate commands and their order.
Advanced	6. Some commands modify the default order of execution, altering when and which instructions are executed.	7. The position of a new command can affect outcomes.		

ferred to comparing students' responses to one assessment item within a CT concept. In this step, different participants' articulations that were assigned the same *a priori* codes were grouped in the same category, whereas those with different assigned codes were grouped in another category. For example, item S.04.b asks participants to provide two sets of instructions for Aisha to carry eight toys from the kitchen to her room while carrying at most three toys each trip. Participant responses that specify the number of toys in each of the multiple trips would be in one category, "using complete, precise, and ordered instructions." However, responses without a specified number of toys for each of the multiple trips would be in a different category, "no articulation of complete, precise, or ordered instruction." The across-unit analysis referred to constantly comparing response categories across different items within the same CT concept. For example, for the sequence items, student responses that fall in the "using complete, precise, and ordered instructions" category for item S.01.a would be compared with responses that fall in the same category for S.04.a. When comparing responses in similar categories of articulated CT, there were a few important questions considered: Did one category of participants' responses to one sequence item share similar characteristics with the same category of responses to another sequence item (e.g., did both categories involve complete, precise, and ordered instructions)? Are there distinct differences in participants' articulation of a CT concept between different categories (e.g., the "no articulation" category did not have characteristics of complete, precise, and ordered instructions)?

Next, themes were constructed for each LT (i.e., CT concept) by making reflections and interpretations [30] of the different categories of responses. When drafting themes to describe the general patterns of students articulated CT, questions involved the following: Is the theme constructed exhaustive (i.e., speaks for all characteristics in one category)? Are different themes mutually ex-

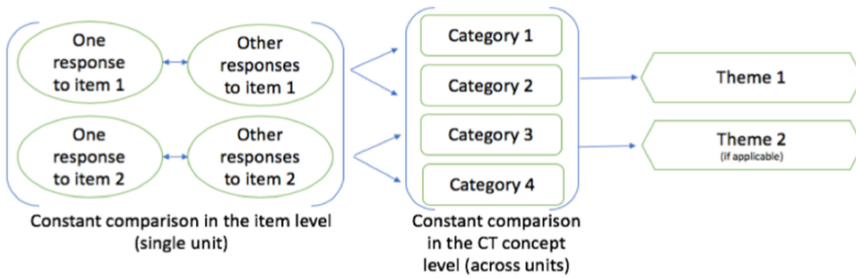


Fig. 1. Analysis using constant comparison for each CT concept.



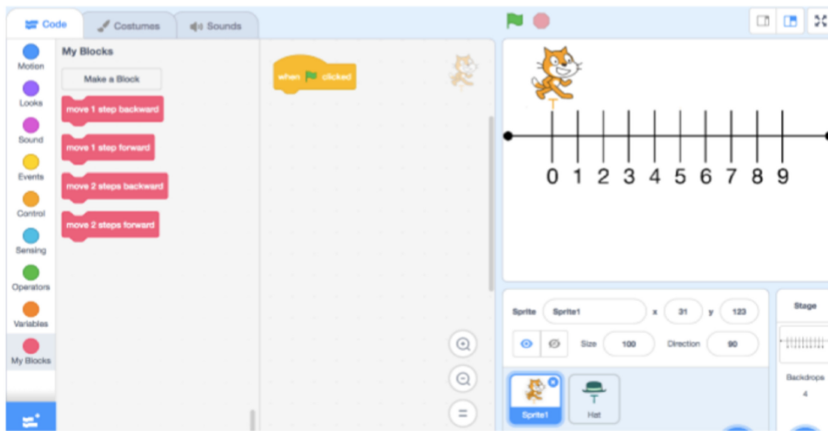
Fig. 2. Direct correspondence between articulated CT and CT LTs.

clusive (i.e., characteristics in one category should not appear in another)? As reflections and interpretations were made, the themes were adjusted to be as comprehensive as possible. For example, the categories from the aforementioned items S.04.b and S.01.a would be holistically examined to devise the potential theme: G3 students generally demonstrated understanding and ability in using complete, precise, and ordered instructions. Figure 1 visualizes the CCA analytical processes for each CT concept.

4.5.2 Analytical Processes for RQ2. To map the correspondence between students' articulated CT and the LTs (the second RQ), recall that the *a priori* codes were verbatim translations of the learning goals in the LTs. Therefore, the *a priori* codes assigned to students' articulation provided a direct correspondence between students' articulated CT and an LT. Figure 2 explains the relationship between the learning goals as *a priori* codes and students' articulated CT.

4.6 Rigor and Trustworthiness

Multiple measures were employed in this study to ensure the credibility, dependability, transferability, and confirmability [22] of the results. For example, to ensure credibility, multiple researchers and investigators with experiences in elementary CT instruction were trained to independently perform data collection (investigator triangulation) [30]. To ensure consistencies of findings, two researchers, who have had experience working on scoring CT assessments associated with this project, coded 40% of participant interviews together to operationalize and check the consistency of the coding of data using *a priori* codes (analyst triangulation) [32]. Given that the two researchers coded the data collaboratively, no percent agreement was calculated. Transferability [22] was ensured by providing a rich, thick description [30] of the research design, method, and findings. To ensure confirmability, a codebook together with the Dedoose software were used to keep track of the assigning of codes with examples from the data, and the codebook was constantly refined during the collaborative coding process. Last but not the least, the first author cross-checked the interpretations made with a CS content expert (an associate professor of CS) who has expertise in K-12 CS education.



Create 2 different scripts (sets of instructions) to move the cat so that he stops at 5 on the number line. Use only the blocks shown above. Write or draw your scripts in the boxes.

Fig. 3. Item S.01.a. (Adapted from materials in the Action Fractions lessons.)

5 FINDINGS

5.1 Participants' Articulated CT

Participants' articulated CT in phase 1 and phase 2 interviews were analyzed using *a priori* codes before constructing themes using constant comparison. The following sections reported the themes that emerged for RQ1: How do G3 and G4 students express and articulate CT in math-CT problem-solving scenarios? Justifications and examples to support the themes were also included. Specifically, the themes and examples were explained for each of the four concept areas: sequence, conditionals, repetition, and decomposition.

5.1.1 Sequence. Three of the 13 items used for the cognitive interviews assessed a range of competencies in the Sequence LT. One theme emerged from students' articulation in the cognitive interviews: Participants (all in G3) generally demonstrated understanding and ability in using complete, precise, and ordered instructions.

The majority of the G3 participants demonstrated successful understanding and use of complete, precise, and ordered instructions in solving the sequence items. For example, the sequence item S.01.a (Figure 3), mapped to all the beginning and the first intermediate learning goals in the Sequence LT, asked participants to write two different programs using the given Scratch code blocks to move the cat sprite from 0 to 5 on the number line. This item was designed to seek evidence of students' abilities and knowledge in constructing two different sets of instructions with the appropriate direction (move back and forward) and the number of steps (one or two steps). All five G3 participants who answered this question provided solutions showing evidence of using and creating complete, precise, and ordered instructions by selecting from the given code blocks to produce the intended outcome, such as the following: "Move one step forward five times" (Participant 3–4); "Move forward 2 and then move forward another 2. And then move forward 1. And then now I'm at 5" (P3–3). In addition, all five participants provided a second set of complete, precise, and ordered instructions that would move the cat sprite from 0 to 5 on the number line, such as "Move two steps forward three times and then one step backward" (P3–19).

Another sequence item (S.02.a), mapped to almost all beginning and intermediate goals in the Sequence LT, asked the students to write or draw the Scratch code to make the cat go back from 5 to 1 on the number line, pick up the hat at 1, and move to 6. Here, students needed to take into

What sound (or sounds) will play if you run this code? _____

If $5 < 8$, then play a “pop” sound.
 If $5 > 7$, then play a “bing” sound.

Fig. 4. Item C.02.e. (Adapted from materials in the Action Fractions lessons.)

consideration the order of actions, the direction (forward or backward), and specify the number of steps to take each time. All 12 G3 participants answered this item, and all but one provided a solution showing evidence of using and creating complete, precise, and ordered instructions by selecting from the given code blocks to produce the intended outcome. For example:

He’s going to move two steps backward, that’s three. So he’s going to move two steps backward again, which is one, pick up the hat, move two steps forward, he’s going to be at 3, move two steps forward again, he’s going to be at five, and one step forward, he’s going to be at 6. (P3–14).

5.1.2 Conditionals. Two of the 13 items used in the cognitive interviews assessed the beginning level of the Conditionals LT, which involves knowing that a condition is something that can be true or false and that a conditional statement connects a condition to an outcome, and having the ability to use conditional statements (e.g., if-then, if-then-else, and event handlers) to evaluate a condition to determine an outcome. Two themes emerged from students’ (all in G4) articulation.

Theme 1. Participants showed an inconsistent understanding of evaluating conditions involving number comparisons.

The G4 participants generally did not articulate how to evaluate the true or false status of conditions while solving the conditional assessment items. Specifically, item C.02.e (Figure 4) provides two conditional statements (If $5 < 8$, then play a “pop” sound; If $5 > 7$, then play a “bing” sound) and asks students what sound(s) will play if the code is run. Mapped to the beginning level learning goals in the Conditionals LT (i.e., A condition is something that can be true or false; A conditional connects a condition to an outcome), this item was designed to seek evidence of students’ understanding that they needed to first evaluate whether the number sentence is true or false in each conditional statement, then decide whether the sound would play. This item was used in both phase 1 and phase 2 interviews. Many of the participants showed evidence of evaluating whether the number comparison expressions were correct or not (e.g., “The first one [pop sound will play]... Because 8 is greater than 5. And 5 is actually less than 7” (P4–1) in the phase 1 cognitive interview. However, most participants (three of four) did not articulate the result of the conditional statements (i.e., whether “pop” and “bing” would play) in the phase 2 interview. Interestingly, two of the three participants who responded to this item in both interviews articulated an understanding of evaluating conditions in the phase 1 interview but failed to do so in the phase 2 interview.

Theme 2. Participants generally showed no understanding of evaluating conditions using user input.

The other conditional item, C.03.b (Figure 5), is also mapped to the beginning-level learning goals in the Conditionals LT. The item asks students to take the user input of the number 2, evaluate the condition ($2 < 5$), and decide what the final outcome is. Only one of nine responses demonstrated evidence of evaluating the “true or false” status of conditions in this item, “Yes... Because 2 is under 5 so [the pop sound will play]” (P4–5). Eight of nine participants did not understand how to take the user input, place the input value in the conditional statement, and evaluate whether the condition is true or false. Some examples of participants’ answers were the following: “I think yes, because if you input 2, the start sound will pop twice” (P4–12); “It’s not gonna make the sound. Because

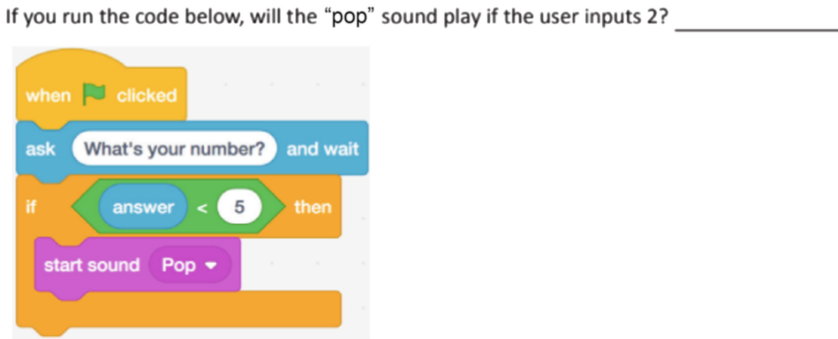


Fig. 5. Item C.03.b. (Adapted from materials in the Action Fractions lessons.)

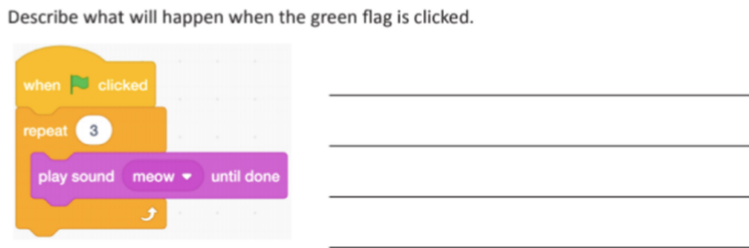


Fig. 6. Item R.03.c. (Adapted from materials in the Action Fractions lessons.)

it says only for 5. It's only gonna work for 5. And that's 2, so that's not gonna work" (P4–14); "I don't know. This is tricky. Because I have no way to explain this, but I know it'll be 1 because 1 is less than 2 (P4–6)."

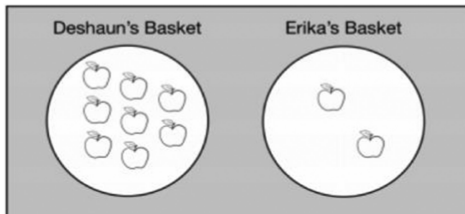
5.1.3 Repetition. Four of the 13 items used in the cognitive interviews assessed students' understanding of cumulative effects (i.e., when repeat commands are used, specific actions will be repeated for a predetermined number of times) and the ability to use repeat commands to create cumulative effects. Two themes emerged from students' (in G3 and G4) articulation.

Theme 1. Participants generally showed evidence of understanding and recognizing repetition.

Participants showed evidence of clearly articulating their understanding of the cumulative effect of the repeat block in a coding context. For example, item R.03.c asked students to describe what will happen when the following Scratch code is run (Figure 6). The item was designed to seek evidence of the beginning- and intermediate-level learning goals in the Repetition LT (i.e., Computers use repeat commands; Repeating things can have a cumulative effect): when repeat commands are used in a computer program, specific actions will be repeated for a predetermined number of times. Ten of the 11 participants who answered this item provided a solution showing evidence of understanding of the cumulative effect of the “repeat” block. For example, students responded: “When you click the green flag, it will repeat meow three times” (P3–3); “It will play meow three times until done” (P4–15).

Theme 2. Participants (in G3 and G4) had difficulty recognizing and constructing repeat instructions in word problems.

Item R.05.a (Figure 7) was a word problem that asked participants to decide how many apples Erika will have after following the repeated instruction: Repeat “taking one apple from DeShaun's



Deshaun's Basket Erika's Basket

Deshaun's basket has 8 apples and Erika's basket has 2 apples.

Instructions:
Repeat 4 times:

- Take 1 apple from Deshaun's basket and put it on the table.
- Take 1 apple from the table and put it in Erika's basket.

If you follow these instructions, how many apples will be in Erika's basket? _____

Fig. 7. Item R.05.a. (Adapted from materials in the Action Fractions lessons.)

Andre has 9 cookies that he wants to give away to 3 friends.
He wants to give each friend an equal number of whole cookies.
Write instructions for giving the cookies to the 3 friends.
Use the instruction "repeat 3 times" at least once.

Fig. 8. Item R.01.a. (Adapted from materials in the Action Fractions lessons.)

basket" four times. Similar to item R.03.c, this item was designed to seek evidence of students' recognizing and understanding what actions were to be repeated and the result of the repeated actions (the beginning and intermediate level learning goals in the Repetition LT). Four of the 10 participants (seven in G3 and three in G4) provided an answer showing evidence of understanding of the cumulative effect of the repeating action, such as "I counted with my fingers, by doing 3, cuz there are already 2 [in Erika's basket], then plus 3, I mean, plus 1, plus 1 and that's 4, plus another 1 and that's 5, plus another 1 and that's 6" (P3-1); "Cuz he [DeShaun originally] has 8 apples and she [Erika originally] has 2, then you repeat it [the instruction] 4 times, she gets 6 apples" (P4-15). However, the majority (6) of the 10 participants who answered this item did not show evidence of understanding the cumulative effect of the repeated action. Participants either did not understand what was to be repeated or only partially repeated the action, such as "It says, take 1 apple away and put it on the table and then put it on Erika's basket. It tells me now Erika has 3 and he has 7" (P3-16); "I took four apples out of Deshawn's basket and I have to put it in her basket and now she has 3 apples" (P3-19).

Items R.01.a (Figure 8; in the phase 1 cognitive interview) and R.01.b (Figure 9; in the phase 2 interview) are two versions of the same word problem, which asked students to write instructions for Andre to give away nine cookies to his three friends using the "repeat 3 times" instruction at least once. The two items were designed to seek evidence of students' ability to identify the pattern of actions (i.e., what could be repeated) and create the cumulative effect by using "repeat" commands instead of using the same commands multiple times (the beginning- and intermediate-level learning goals in the Repetition LT). The only difference between the two items was that R.01.a did not present a sample instruction, whereas R.01.b did. For R.01.a, 11 of the 12 students (5 in G3 and 7 in G4) who answered this item did not show any evidence of using repetition in their instruction but rather simply explained the mathematical results: "So I drew three circles and put three cookies in each so then it goes 3, 6, and 9" (P3-5); "Three times 3 is 9 and 9 divided by 3 is 3" (P4-6). Similarly, for R.01.b, 7 of 11 (4 in G3 and 7 in G4) did not show any evidence of using repetition in their instruction: "... he gave away 3 to Sally, and then 3 to Val and then 3 to Lee"

Andre has 9 cookies to give away to his friends Sally, Val, and Lee.
 He wants to give each friend an equal number of cookies.
 Andre wrote instructions for how to give away the cookies.

Rewrite his instructions. Use the instruction “repeat 3 times” at least once.

Andre’s instructions: • Give Sally 1 cookie • Give Val 1 cookie • Give Lee 1 cookie • Give Sally 1 cookie • Give Val 1 cookie • Give Lee 1 cookie • Give Sally 1 cookie • Give Val 1 cookie • Give Lee 1 cookie	Your instructions:
---	----------------------------------

Fig. 9. Item R.01.b. (Adapted from materials in the Action Fractions lessons.)

(P3–2); “... you add 3 times 3 is 9. And 9 divided by 3 is 3” (P4–6). The other four participants (all in G4) incorporated the “repeat” command in their solution. For example:

I just put a [repeat] block there to represent it’ll be repeating something, cuz you don’t want to put something in the code that says “give Sally one, give Val one, give Lee one” just over and over again. This is just a quicker way. So, Val gets one cookie, and then Lee gets one, and then Sally gets one cookie, and then they are gonna repeat that three times. (P4–14).

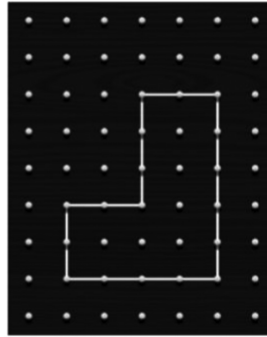
First, I wrote of a repeat [three times] block. Then, then I wrote, give Sally one cookie, then give Val one cookie, and then give Lee one cookie. (P4–16).

5.1.4 Decomposition. Four of the 13 items in the cognitive interviews assessed students’ knowledge that systems (e.g., a number or a problem) are made up of smaller and distinct parts and their ability to break a complex problem into a set of simpler problems. One theme emerged from students’ (in G3 and G4) articulation: Participants used decomposition in number problems but had difficulty decomposing the math-integrated word problem.

Data analysis showed abundant evidence of participants’ identifying the components of a system in the context of mathematics (a number, a number problem, etc.). For example, item DC.02.a asked students to “write one or more addition number sentences that mean the same as the multiplication number sentence: $5 \times 4 = 20$.” Mapped to the initial learning goal in the Decomposition LT (i.e., Systems are made up of smaller parts), the item was designed to seek evidence of students’ understanding and ability to identify what smaller numbers constitute the number 20. The majority of participants (four of six) who answered this item provided a solution showing evidence of identifying the subcomponents of the number, 20. For example, “10 plus 10 equals 20, 19 plus 1 equals 20, 9 plus 11 equals 20” (P3–8); “18 plus 2 equals 20” (P3–5).

Item DC.02.b asked students to decompose the problem: $(5 \times 2) + (3 \times 2)$. In addition to students’ identifying the subcomponents of a number, this item also involved students’ breaking the problem into a set of simpler problems (i.e., multiplication first, then addition). This item is mapped to three learning goals in the Decomposition LT (i.e., Systems are made up of smaller parts; Complex problems can be broken into smaller parts; Problem decomposition is a useful early step in problem-solving). Almost all participants (15 of 17) who answered this item showed evidence of breaking down the number problem into a step-by-step process. For example, “5 times 2 is 10, 3 times 2 is 6. So 6 plus 10 equals 16” (P4–6); “Basically 5 times 2 equals 10, and 3 times 2 makes 6, and if you join them and it has a plus, if you join them, it’ll be 16” (P3–5).

Pretend you want to find the area of the shape outlined in white.



This problem requires multiple steps. Break down the problem into steps.
Describe your steps.

Fig. 10. Item DC.06.c. (Adapted from materials in the Action Fractions lessons.)

Item DC.06.c (Figure 10) is a word problem that asks students to list the multiple steps involved in finding the area of a polygon. This item is mapped to the same learning goals as item DC.02.b is. The item seeks evidence of students' ability to break up the shape into two rectangles or multiple small squares, find the area for those smaller shapes, before adding them up for the final area. The majority of the responses (30 of 38) did not show evidence of using decomposition to break down the problem. Many provided a solution that was not logical or irrelevant to finding the area of the polygon. For example:

[First I] count the dots... then I would like, add all the sides up. (P4–16).

So I think the whole in the perimeter equals, so I think the perimeter is 18 for the whole thing and then the multiplication sentence is 6×3 . (P3–12).

Eight of the 38 responses to this item mentioned breaking up the polygon into smaller squares or rectangles, then finding the area of the smaller squares before adding them up for the total area. For example, one participant (P3–19) said:

So I draw the little lines here to, like make little boxes, to see. So step 1, I would make little squares. [writing] And after I make the squares, I would count them. [writing] but I can also break it in half... So if I break it [the polygon] into 2 rectangles, I could count 1,2,3,4,5,6 [squares in the top rectangle]. Then another one, 12,3,4,5,6,7,8 [squares in the bottom rectangle]. 6 plus 8 equals what? Oh, 14. So I can make 2 rectangles. And then I could count it. After I count it [them] both, I can add [them]. Like I said, 6 on the top [writing] 8 on the bottom.

5.2 Correspondence between Students' Articulated CT and the CT LTs

This section reported the correspondence between students' articulated CT and the CT LTs (RQ2), as evidenced by the *a priori* codes (verbatim translation of learning goals in the LTs) assigned to students' CT articulation. Tables 3–6 list examples of students' CT articulation to an item and the *a priori* codes assigned during data analysis. As illustrated by Figure 2 earlier in this article, since the *a priori* codes directly correspond to the learning goals in each of the LTs, the mapping between students' articulated responses and the LTs was possible. *Sequence*: The G3 participants' articulation of complete, precise, and ordered instructions was mapped onto most of the beginning-

Table 3. Participants' Articulation in Item S.02.a (Left Column) and the Assigned *a Priori* Codes (i.e., the Corresponding Learning Goals in the Sequence LT) (Right Column)

Examples of Participants' Response to S.02.a	Corresponding Learning Goals in the Sequence LT (i.e., <i>a priori</i> codes assigned)
"So I see he's [the cat is] at 5, so he has to move back 1,2,3,4. So there's a 'move 2 back [block],' so I put 2 of those [blocks]. [writing] Move 2 steps backward, another one, move 2 steps backward, so 1, 2, 1, 2. Now move forward, wait I gotta pick up [the] hat. Then when I'm at the hat, I can move forward 1, 2, 3, 4, 5. So I can put two of these [move 2 forward blocks] and then move 1 forward. [writing 'move 2 steps forward, move 2 steps forward'] Then 1 step forward" (P3-19).	1. Precise instructions are more likely to produce the intended outcome than general ones. (B) 3.1 Precision and completeness are important when writing instructions in advance. (B) 3.2 Programs are made by assembling instructions from a limited set. (B) 4.1 Computers require precise instructions using limited commands. (B) 3. The order in which instructions are carried out can affect the outcome. (I) 4. Computers have a default order of execution, so order matters in programming. (I) 5. Creating working programs requires considering both appropriate commands and their order. (I)
"I'll just put [turn] right. Turn right and then forward, 5, and then, [he] grabs the hat, and then he turns over left, and then he goes to 9, um 4 steps" (P3-12).	No articulation of sequence.

Table 4. Participants' Articulation in item C.02.e (Left Column) and the Corresponding Learning Goals in the Conditional LT (Right Column)

Examples of Participants' Response to C.02.e	Corresponding Learning Goals in the Conditionals LT (i.e., <i>a priori</i> codes assigned)
"5 is less than 8 so [it will play pop]" (P4-16).	0. Actions often result from specific causes. (B) 1. A condition is something that can be true or false. (B) 2. A conditional connects a condition to an outcome. (B)
"Pop and bing [will play]" (P4-1, P4-7).	No articulation of conditionals.

and intermediate-level learning goals in the Sequence LT, suggesting that the participants generally moved along the LT towards the advanced level. *Conditionals*: The G4 participants' limited understanding of evaluating the true-or-false state of a condition involving number comparisons and user input suggested that they were yet to meet the beginning-level learning goals in the Conditionals LT. *Repetition*: Participants' (both G3 and G4) articulated understanding of cumulative effects was mapped to most of the learning goals in the beginning level and one in the intermediate level, suggesting that participants were progressing toward the intermediate level of the Repetition LT. *Decomposition*: Participants' (both G3 and G4) articulation of decomposing numbers and problems suggested that participants met the initial learning goal in the Decomposition LT but all were not progressing toward the next learning goals.

Table 5. Participants' Articulation in Item R.01.a (Left Column) and the Corresponding Learning Goals in the Repetition LT (Right Column)

Examples of Participants' Responses to R.01.a	Corresponding Learning Goals in the Repetition LT (i.e., <i>a priori</i> codes assigned)
"One friend has three. Another one another friend has three and the last one has three. [Writing 'one friend has 3 cookies'] Maybe repeats three times" (P3–14). "First I wrote of a repeat block. Then. Then I wrote, give Sally one cookie then give Val one cookie and then give Lee one cookie" (P4–16).	2. Some tasks involve repeating actions. (B) 3. Instructions like "Step 3 times" do the same thing as "Step, step, step." (B) 1. Repeating things can have a cumulative effect. (I)
"I'm drawing 3 friends and I have a circle by each friend and then I'm gonna give cookies to each friend. And then gonna see if they have an equal amount of cookies" (P3–15). " So I drew three circles and put three cookies in each so then it goes 3 6 and 9" (P3–5). "You add 3 times three is 9. And 9 divided by 3 is 3" (P4–6).	No articulation of repetition.

Table 6. Participants' Articulation in Item DC.06.c (Left Column) and the Corresponding Learning Goals in the Decomposition LT (Right Column)

Examples of Participants' Response to DC.06.c	Corresponding Learning Goals in the Decomposition LT (i.e., <i>a priori</i> codes assigned)
"So you could split this [polygon]. So there's 1,2,3,4,6,7 right here [in the top shape]. And then 1,2,3,4,5,6,7,8,9,10,11, 11 here [in the bottom shape]. So add those 2 together, 11 plus 7 equals 18" (P4–1).	1. Systems are made up of smaller parts. 2. Complex problems can be broken into smaller parts. 3. Problem decomposition is a useful early step in problem-solving.
"So I think the whole in the perimeter equals, so I think the perimeter is 18 for the whole thing and then the multiplication sentence is 6×3" (P3–12).	No articulation of decomposition.

6 DISCUSSION

Emerging research, updated standards, and LTs have suggested "what to teach" and "what should be learned" in K-8 CS education. The Computer Science Teachers Association standards include grade bands for concepts and practices that should be taught, whereas the LTs [36, 39] provide

hypotheses of students' learning of CT concepts that are not grade-specific. Although there are similarities between these, a systematic match between what CT instruction is offered in existing curricula and what CT competencies elementary students are demonstrating is yet to be established [51], especially in the context of integrated instruction.

This study provided an example for integrating CT in mathematics and generated initial empirical evidence for this sample of G3 and G4 students' current CT competencies after engaging in the integrated instruction. Specifically, G3 students progressed far along the Sequence LT and demonstrated the competencies of using complete, precise, and ordered instructions in coding and daily life problem-solving. However, G4 students' understanding of conditionals was limited, demonstrating no consistent understanding of evaluating the true-or-false status of conditions and had difficulty articulating results depending on the true-or-false state of conditions. In terms of repetition and decomposition, G3 and G4 students generally understood and recognized the cumulative effect of using repeat blocks in the context of coding and could decompose numbers into smaller parts. But when problems involved more complexities, such as requiring the understanding of an additional CT concept or math content knowledge, students were less likely to articulate clear problem-solving reasoning. Such empirical evidence of elementary students' CT has important implications for elementary CT instruction and assessment in supporting students' progression toward increasingly advanced CT learning.

6.1 Implications for Elementary CT Instruction

It is important to emphasize the need to closely examine instruction in how elementary students are exposed to the various CT concepts. The lessons implemented in this study integrated CT concepts into third- and fourth-grade fractions content in the Everyday Mathematics curriculum (i.e., EM, 4th edition) based on the hypothesized LTs. As the LTs are not grade-specific by design, these integrated lessons were an initial, exploratory attempt to attach learning goals in the LTs to third- and fourth-grade EM lessons. The purpose of such an LT-based approach was to offer instruction based on students' current levels of knowledge development so students can confidently develop increasingly sophisticated knowledge while meeting the hypothesized learning goals [12, 17].

On one hand, the findings were promising in that the G3 students learned sequence as they should (i.e., by meeting almost all beginning- and intermediate-level learning goals in the Sequence LT). Such findings of students' learning corroborated previous studies in that young children were capable of selecting and ordering instructions in simple programming tasks that did not involve control flows [8, 25, 27]. With such increasing evidence, research can better inform when CT instruction of sequence can move to the next level, such as altering the order of execution by using the various commands (e.g., repeat blocks). As such, instruction provides an onramp for the students to learn other CT concepts by using the sequence concept as a "springboard" [39]. That said, however, further research is warranted to expand the empirical evidence base for a better understanding of when to introduce the various CT concepts to elementary students in a specific grade level.

On the other hand, the findings also revealed potential challenges when elementary students are beginning to learn the CT concepts. For example, students had difficulty constructing repeat instructions in word problems and evaluating conditional statements. Given the limited prior research in elementary students' understanding of repetition and conditionals, such empirical evidence provided important insight into elementary students' CT in integrated learning and highlighted the need for more instructional exposure on these specific CT concepts. With more instructional exposure, students are allowed sufficient time to connect new ideas and knowledge with their pre-existing intuitions [35], thus avoiding potential enlarged learning gaps and skipping levels. In addition, students may have difficulty solving a problem that involves combined

CT concepts (e.g., evaluating a conditional statement using variables to store user input) when students' knowledge of individual concepts is not yet established. In such a case, future research should explore whether increasing instructional exposure alone can address the challenges or the introduction of the various CT concepts should be more carefully sequenced. A recent systematic review concluded that previous research generally introduced standalone CT concepts in fourth grade before moving to combined CT concepts (e.g., conditionals before conditional loops) toward the end of sixth grade [51]. Future research may explore whether such an instructional sequence is consistent with elementary students' cognitive development.

6.2 Implications for CT Assessments

Generally speaking, the assessments complement the LT-based instruction by providing evidence of students' understanding of the four CT LTs. This study sets an example of using assessment results to inform what CT competencies elementary students can demonstrate after engaging in integrated CT instruction. Although research in CT expanded during the past decade, teachers do not yet have access to general diagnostic assessments of CT that provide the necessary information about knowledge gaps that prohibit learning progressions. The assessment results provided teachers and researchers with an initial understanding of where students were likely to have knowledge gaps after engaging in a series of integrated lessons. However, this study also acknowledges that more research is needed to establish the general diagnostic power of such CT assessments, especially in the context of integrated instruction, so results can be interpreted to inform instruction.

In addition, findings suggested that the difficulty level of assessment items in an integrated context is affected by the embedded content knowledge of the subject area that CT is integrated into (e.g., math), the CT competencies an item elicits (i.e., various learning goals), and the context in which the item is embedded (i.e., word problems, number problems, and coding-based problems). Findings revealed that the math knowledge required in answering some of the assessment items can interfere with students' ability to solve an item. For example, when students are not familiar with finding the area, they may not be able to articulate how to decompose the problem to find the area of a polygon, even when no mathematical calculation with numbers is needed. In such cases, content knowledge of the subject is likely to become a confounding factor. Therefore, future research should take into consideration the difficulty level of the content knowledge when designing CT assessments for integrated contexts.

As well, although items may be mapped to the same learning goals in an LT, students' ability to solve a number problem did not necessarily mean that they could solve an applied word problem mapped to the same learning goals as effectively. Previous research suggested that word problems may have introduced confounding complexities, such as reading fluency and comprehension (i.e., in-depth analysis of concepts and relationships in the problem text) [44] and content area (i.e., math) competency, thus adding variability in students' articulated CT. Therefore, word problems can be more challenging to students than number problems because of the semantic and mathematical structures involved [44]. Word problems require students to engage in a flexible and holistic analysis of the problem. Such analysis includes translating the problem into mathematical expressions while taking into consideration the underlying concepts and relationships between quantities. Although identifying the numerical units poses no obvious challenge, students are often result-driven and are inclined to go straight into calculations without first evaluating the reasons and strategies for doing them [44]. Therefore, CT assessments should have the flexibility to assess students' CT competencies by providing different problem-solving contexts to precisely capture students' CT.

Finally, this study demonstrated the benefits of using cognitive interviews as a diagnostic approach to assess elementary students' CT and investigate where students were having conceptual

gaps based on the LTs. Cognitive interviews reveal the “ideas, concepts, and skills that children can express, articulate, and develop when they are given the opportunity to engage with rich and interesting activities” and allow students “to express and reconsider their experience and understanding” (p. 245) [17]. The cognitive interviews offer affordances that the other assessment types (i.e., tests or project portfolios) may miss. For example, in a learning product-focused assessment (i.e., a test or a programming project), students can guess an answer to a multiple-choice question without having to provide the rationale that supports their answers, or they can use certain blocks without knowing how they work. However, in a cognitive interview, students need to justify their answers by verbalizing their thought processes. As such, students’ misconceptions and/or learning gaps, if any, can likely be revealed. Such verbal data of students’ problem-solving rationale afford a deeper understanding of how students are applying CT, which is often missed by using traditional tests that assess CT as a learning product.

6.3 Implications for CT Integration in Subject Areas Besides Math

It was the intention of the authors that the findings of this study, such as students’ demonstration of CT understanding and the recommendations for integrated CT instruction and assessment, will better inform the design decisions for elementary CT integration in areas besides math. The empirical evidence generated by this study provided us with an initial understanding of how far upper elementary grade students can progress along the LTs after engaging in integrated instruction mapped to the learning goals in the four CT LTs. Researchers who aspire to integrate CT into various subjects may design instruction and assessments by following the learning goals in the theorized LTs and considering the empirical findings, challenges, and the potential learning gaps presented. As an illustration of CT integration in language arts, a fourth-grade lesson can be designed to introduce and reinforce students’ understanding of conditional statements. In this integrated lesson, short statements that describe the procedures or concepts in a historical or a scientific event may be presented, and students will evaluate whether the statements are true or false (i.e., whether they are accurate descriptions or not). Then, a follow-up activity can have students design and execute (e.g., acting out) different outcomes or actions associated with the true and false conditions, respectively. For example, if a statement is true, then the class will cheer and move on to evaluating the next statement. If false, the class will revise the statement. Such integrated instruction can be repeated or enriched by additional hands-on coding activities. It is important to note that the complexities of the LTs may place a burden on educators and teachers who wish to implement LT-based instruction. To reduce the burden, PD opportunities may be designed to help educators and teachers gain a better understanding of the learning goals in the LTs and what student learning outcomes may be expected based on empirical evidence such as that presented in this study. Researchers and educators are also encouraged to use cognitive interviews to understand students’ problem-solving in different subjects where CT is integrated. Such data is expected to inform the field of how students apply CT in different contexts and how the focal subject content may interfere or facilitate students’ applying CT in problem-solving.

6.4 Limitations

Findings from this study should be taken with the following limitations in mind. First, as the purpose of this study was to examine how elementary students apply CT in problem-solving, only a small number of students from the G3 and G4 classes in the same elementary school participated in the cognitive interviews. In addition, due to missing data in the demographic information of the G3 participants, the findings should be referenced acknowledging the specific sample of students that participated in the math-CT integrated lessons. The interpretations of the findings were limited by the number of assessment items sampled for each LT, and the assessment items sampled

for this study cover only specific portions of the LTs. Although the repetition, conditionals, and decomposition items received responses that demonstrated a range of competencies, the sequence items received more consistent responses. Therefore, it is possible that students have a more advanced understanding that is not captured by the sequence items used. Finally, the findings and the suggestions made in this study should be interpreted acknowledging the interrelated design and the complementary relationships among the LTs, the lessons, and the assessments. Interpretations may not be transferable to cases where the curriculum and assessments were not aligned to the LTs.

7 CONCLUSION

This study provided a detailed qualitative account of third- and fourth-grade students' CT articulations as they progressed in integrated math-CT instruction. Using the cognitive interviews, the study collected evidence of students' articulated CT proficiencies and examined students' CT with respect to the learning goals and progression hypothesized in the CT LTs. The findings provided initial empirical evidence to validate the LTs and highlight the benefits of LT-based instruction in supporting elementary students' CT learning. The study also contributed to the body of literature in elementary CT assessments and served as an example for integrating CT in elementary subject areas besides math.

APPENDIX

A THINK-ALOUD PROTOCOL

Prepping the participant

[Intro] Hello. I am *researcher name*, a student at * school name*. I am interested in learning about how you solve the following problems. This activity will take about 15 to 20 minutes. Your parents said it's Okay for you to do it. And I just need you to sign this (assent) before we start. I am recording our conversation so I don't have to take notes. The set ID is # - # (e.g., 3-1).

We are going to do a "think-aloud" activity. Basically, I'm asking that you tell me what you are thinking as you work through the problems. I didn't write these questions, you can say whatever you are thinking while working on them. You will not be graded on this. Whatever you say won't hurt my feelings.

Now I will show you an example of "thinking aloud" when solving a problem [show participant the item].

Kristen uses her fraction strips to compare $\frac{1}{3}$ and $\frac{1}{4}$.

$\frac{1}{3}$	$\frac{1}{3}$	$\frac{1}{3}$	
$\frac{1}{4}$	$\frac{1}{4}$	$\frac{1}{4}$	$\frac{1}{4}$

Kristen writes this number sentence: $\frac{1}{3} < \frac{1}{4}$

Do you agree with Kristen? _____

Use Kristen's fraction strips to help explain your answer.

[anno: So the question reads, Kristen uses her fraction strips to compare $\frac{1}{3}$ and $\frac{1}{4}$. And she writes $\frac{1}{3}$ is smaller than $\frac{1}{4}$. Do you agree with Kristen? Use fraction strips to help explain your answer. So I don't agree with Kristen because $\frac{1}{3}$ should be bigger than $\frac{1}{4}$. So if you divide the same area into three equal parts, each is $\frac{1}{3}$, and that's how big each part is (referring to the top strip and marking the first

$\frac{1}{3}$). If you divide the same area into four equal parts, each is $\frac{1}{4}$ and this big (referring to the bottom strip and marking the first $\frac{1}{4}$). So $\frac{1}{3}$ is bigger than $\frac{1}{4}$.]

Now I will give you a couple of problems. Please remember to talk out loud just like I did when working on a problem. If you go silent for more than 5 seconds, I might ask you to describe what you are thinking. I will also ask you to tell me what you think of the item. Is it okay?

Prompting questions—DURING think-aloud

- (At the very beginning) Now go ahead and start by reading aloud the question.
- Can you tell me what you are thinking right now?
 - (if the student goes silent for more than 5 seconds; if you know the student is thinking but not talking).

Questions should be asked with the goal of probing for the following if the student did not provide enough “thinking”:

- For multiple-choice: why an answer is or is not selected?
- For open-ended tasks: probe for the rationale of the given solution.

Wrap-up reflection question—AFTER think-aloud

- What do you think this question is asking you to do?
- What subject or subjects you learn at school do you think this is?
- What do you think that your classmates will need to know in order to answer this question correctly (probe for examples/detail)?

Remember: we are NOT trying to help participants answer the questions correctly, rather we are uncovering whatever the participant is thinking when problem-solving.

REFERENCES

- [1] Charoula Angeli and Nicos Valanides. 2020. Developing young children’s computational thinking with educational robotics: An interaction effect between gender and scaffolding strategy. *Computers in Human Behavior* 105 (2020), 105954.
- [2] Soumela Atmatzidou and Stavros Demetriadis. 2016. Advancing students’ computational thinking skills through educational robotics: A study on age and gender relevant differences. *Robotics and Autonomous Systems* 75 (2016), 661–670.
- [3] Jeffrey E. Barrett and Michael T. Battista. 2014. Two approaches to describing the development of students’ reasoning about length. In *Learning over Time: Learning Trajectories in Mathematics Education*, Alan P. Maloney, Jere Confrey, and Kenny H. Nguyen (eds.). Information Age Publishing, Charlotte, NC, 97.
- [4] Valerie Barr and Chris Stephenson. 2011. Bringing computational thinking to K-12: What is involved and what is the role of the computer science education community? *ACM Inroads* 2, 1 (2011), 48–54.
- [5] Tim Bell and Jan Vahrenhold. 2018. CS unplugged—How is it used, and does it work? In *Adventures between Lower Bounds and Higher Altitudes*. Springer, 497–521.
- [6] Matthew Berland and Uri Wilensky. 2015. Comparing virtual and physical robotics environments for supporting complex systems and computational thinking. *Journal of Science Education and Technology* 24, 5 (2015), 628–647.
- [7] Marina U. Bers. 2010. The TangleK robotics program: Applied computational thinking for young children. *Early Childhood Research & Practice* 12, 2 (2010), 1–20.
- [8] Marina U. Bers, Louise Flannery, Elizabeth R. Kazakoff, and Amanda Sullivan. 2014. Computational thinking and tinkering: Exploration of an early childhood robotics curriculum. *Computers & Education* 72 (2014), 145–157.
- [9] Hennie Boeijs. 2002. A purposeful approach to the constant comparative method in the analysis of qualitative interviews. *Quality and Quantity* 36, 4 (2002), 391–409.
- [10] Karen Brennan and Mitchel Resnick. 2012. New frameworks for studying and assessing the development of computational thinking. In *Proceedings of the 2012 Annual Meeting of the American Educational Research Association*. 1–25.
- [11] Guanhua Chen, Ji Shen, Lauren Barth-Cohen, Shiyang Jiang, Xiaoting Huang, and Moataz Eltoukhy. 2017. Assessing elementary students’ computational thinking in everyday reasoning and robotics programming. *Computers & Education* 109 (2017), 162–175.

- [12] Douglas H. Clements and Julie Sarama. 2004. Learning trajectories in mathematics education. *Mathematical Thinking and Learning* 6, 2 (2004), 81–89.
- [13] Douglas H. Clements and Julie Sarama. 2009. Learning trajectories in early mathematics—Sequences of acquisition and teaching. In *Encyclopedia of Language and Literacy Development*, R. E. Tremblay, M. Bolvin, and R. Peters (eds.). Online, 1–7.
- [14] Douglas H. Clements, Julie Sarama, Arthur J. Baroody, Candace Joswick, and Christopher B. Wolfe. 2019. Evaluating the efficacy of a learning trajectory for early shape composition. *American Educational Research Journal* 56, 6 (2019), 2509–2530.
- [15] Douglas H. Clements, Julie Sarama, Arthur J. Baroody, and Candace Joswick. 2020. Efficacy of a learning trajectory approach compared to a teach-to-target approach for addition and subtraction. *ZDM Mathematics Education* 52 (2020), 1–12.
- [16] Computer Science Teachers Association. (2017). CSTA K-12 Computer Science Standards. Retrieved January 14, 2022 from <https://www.doe.k12.de.us/cms/lib/DE01922744/Centricity/Domain/176/CSTA%20Computer%20Science%20Standards%20Revised%202017.pdf>.
- [17] Jere Confrey, Alan P. Maloney, and Kenny H. Nguyen. 2014. Learning trajectories going forward. In *Learning over Time: Learning Trajectories in Mathematics Education*, Alan P. Maloney, Jere Confrey, and Kenny H. Nguyen (eds.). Information Age Publishing, Charlotte, NC, 243.
- [18] Thomas J. Cortina. 2015. Reaching a broader population of students through “unplugged” activities. *Communications of the ACM* 58, 3 (2015), 25–27.
- [19] Phil Daro, Frederic A. Mosher, and Thomas B. Corcoran. 2011. Learning trajectories in mathematics: A foundation for standards, curriculum, assessment, and instruction. *Research Report. Consortium for Policy Research in Education (CPRE)*. http://repository.upenn.edu/cpre_researchreports/60.
- [20] Georgios Fessakis, Evangelia Gouli, and E. Mavroudi. 2013. Problem solving by 5–6 years old kindergarten children in a computer programming environment: A case study. *Computers & Education* 63 (2013), 87–97.
- [21] Brian Gane, Maya Israel, Noor Elagha, Wei Yan, Feiya Luo, and Jim Pellegrino. 2021. Design and validation of learning trajectory-based assessments for computational thinking in upper elementary grades. *Computer Science Education* 31, 2 (2021), 141–168.
- [22] Egon G. Guba and Yvonna S. Lincoln. 1981. Effective evaluation: Improving the usefulness of evaluation results through responsive and naturalistic approaches. Jossey-Bass.
- [23] Maya Israel, Ruohan Liu, Wei Yan, Heather Sherwood, Wendy Martin, Cheri Fancseli, Edgar Rivera-Cash, and Alexandra Adair. Understanding barriers to school-wide computational thinking integration at the elementary grades: Lessons from three schools. *ACM Special Issue on Computational Thinking*. In press.
- [24] K–12 Computer Science Steering Committee. 2016. K12 Computer Science Framework. Retrieved January 14, 2022 from <https://k12cs.org/wp-content/uploads/2016/09/K%E2%80%9312-Computer-Science-Framework.pdf>.
- [25] Elizabeth R. Kazakoff, Amanda Sullivan, and Marina U. Bers. 2013. The effect of a classroom-based intensive robotics and programming workshop on sequencing ability in early childhood. *Early Childhood Education Journal* 41, 4 (2013), 245–255.
- [26] Irene Lee, Fred Martin, Jill Denner, Bob Coulter, Walter Allan, Jeri Erickson, Joyce Malyn-Smith, and Linda Werner. 2011. Computational thinking for youth in practice. *ACM Inroads* 2, 1 (2011), 32–37.
- [27] Feiya Luo, Pavlo “Pasha” Antonenko, and E. Christine Davis. 2020. Exploring the evolution of two girls’ conceptions and practices in computational thinking in science. *Computers & Education* 146 (2020), 103759.
- [28] Feiya Luo, Maya Israel, Ruohan Liu, Wei Yan, Brian Gane, and John Hampton. 2020. Understanding students’ computational thinking through cognitive interviews: A learning trajectory-based analysis. In *Proceedings of the 51st ACM Technical Symposium on Computer Science Education* ACM, New York, NY, 919–925.
- [29] Alan P. Maloney, Jere Confrey, and Kenny H. Nguyen. 2014. *Learning Over Time: Learning Trajectories in Mathematics Education*. Information Age Publishing, Charlotte, NC.
- [30] Sharan B. Merriam and Elizabeth J. Tisdell. 2015. *Qualitative Research: A Guide to Design and Implementation*. John Wiley & Sons.
- [31] NGSS Lead States. 2013. *Next Generation Science Standards: For States, by States*. The National Academies Press, Washington, DC. <https://www.nextgenscience.org/>.
- [32] Michael Quinn Patton. 1999. Enhancing the quality and credibility of qualitative analysis. *Health Services Research* 34, 5 Pt 2 (1999), 1189–1208.
- [33] David Pepper, Jeremy Hodgen, Katri Lamesoo, Pille Kõiv, and Jos Tolboom. 2018. Think aloud: Using cognitive interviewing to validate the PISA assessment of student self-efficacy in mathematics. *International Journal of Research & Method in Education* 41, 1 (2018), 3–16.
- [34] Christina Hamme Peterson, N. Andrew Peterson, and Kristen Gilmore Powell. 2017. Cognitive interviewing for item

- development: Validity evidence based on content and response processes. *Measurement and Evaluation in Counseling and Development* 50, 4 (2017), 217–223.
- [35] Mitchel Resnick, Amy Bruckman, and Fred Martin. 1996. Pianos not stereos: Creating computational construction kits. *Interactions* 3, 5 (1996), 40–50.
 - [36] Kathryn M. Rich, T. Andrew Binkowski, Carla Strickland, and Diana Franklin. 2018. Decomposition: A K-8 computational thinking learning trajectory. In *Proceedings of the 2018 ACM Conference on International Computing Education Research*. ACM, New York, NY, 124–132.
 - [37] Kathryn M. Rich, Diana Franklin, Carla Strickland, Andy Isaacs, and Donna Eatinger. 2020. A learning trajectory for variables based in computational thinking literature: Using levels of thinking to develop instruction. *Computer Science Education* 2020 (2020), 1–22.
 - [38] Kathryn M. Rich, Carla Strickland, T. Andrew Binkowski, and Diana Franklin. 2019. A K-8 debugging learning trajectory derived from research literature. In *Proceedings of the 50th ACM Technical Symposium on Computer Science Education*. ACM, New York, NY, 745–751.
 - [39] Kathryn M. Rich, Carla Strickland, T. Andrew Binkowski, Cheryl Moran, and Diana Franklin. 2017. K-8 learning trajectories derived from research literature: Sequence, repetition, conditionals. In *Proceedings of the 2017 ACM Conference on International Computing Education Research*. ACM, New York, NY, 182–190.
 - [40] Kathryn M. Rich, Aman Yadav, and Christina V. Schwarz. 2019. Computational thinking, mathematics, and science: Elementary teachers' perspectives on integration. *Journal of Technology and Teacher Education* 27, 2 (2019), 165–205.
 - [41] Brandon Rodriguez, Stephen Kennicutt, Cyndi Rader, and Tracy Camp. 2017. Assessing computational thinking in CS unplugged activities. In *Proceedings of the 2017 ACM SIGCSE Technical Symposium on Computer Science Education*. ACM, New York, NY, 501–506.
 - [42] José-Manuel Sáez-López, Marcos Román-González, and Esteban Vázquez-Cano. 2016. Visual programming languages integrated across the curriculum in elementary school: A two year case study using “Scratch” in five schools. *Computers & Education* 97 (2016), 129–141.
 - [43] Julie Sarama, Douglas H. Clements, Jeffrey Barrett, Douglas W. Van Dine, and Jennifer S. McDonel. 2011. Evaluation of a learning trajectory for length in the early years. *ZDM Mathematics Education* 43, 5 (2011), 667.
 - [44] Annie Savard and Elena Polotskaia. 2017. Who's wrong? Tasks fostering understanding of mathematical relationships in word problems in elementary students. *ZDM Mathematics Education* 49, 6 (2017), 823–833.
 - [45] Martin A. Simon. 1995. Reconstructing mathematics pedagogy from a constructivist perspective. *Journal for Research in Mathematics Education* 26, 2 (1995), 114–145.
 - [46] Carla Strickland, Kathryn M. Rich, Donna Eatinger, Todd Lash, Andy Isaacs, Maya Israel, and Diana Franklin. 2021. Action Fractions: The design and pilot of an integrated math+CS elementary curriculum based on learning trajectories. In *Proceedings of the 52nd ACM Technical Symposium on Computer Science Education*. ACM, New York, NY, 1149–1155.
 - [47] Xiaodan Tang, Yue Yin, Qiao Lin, Roxana Hadad, and Xiaoming Zhai. 2020. Assessing computational thinking: A systematic review of empirical studies. *Computers & Education* 148 (2020), 103798.
 - [48] Christiane Gresse Von Wangenheim, Nathalia Cruz Alves, Pedro Eurico Rodrigues, and Jean Carlo Hauck. 2017. Teaching computing in a multidisciplinary way in social studies classes in school—A case study. *International Journal of Computer Science Education in Schools* 1, 2 (2017), 1–14.
 - [49] David Weintrop, Elham Beheshti, Michael Horn, Kai Orton, Kemi Jona, Laura Trouille, and Uri Wilensky. 2016. Defining computational thinking for mathematics and science classrooms. *Journal of Science Education and Technology* 25, 1 (2016), 127–147.
 - [50] Linda Werner, Jill Denner, Shannon Campe, and Damon Chizuru Kawamoto. 2012. The fairy performance assessment: Measuring computational thinking in middle school. In *Proceedings of the 43rd ACM Technical Symposium on Computer Science Education*. ACM, New York, NY, 215–220.
 - [51] LeChen Zhang and Jalal Nouri. 2019. A systematic review of learning computational thinking through Scratch in K-9. *Computers & Education* 141 (2019), 103607.

Received February 2021; revised September 2021; accepted October 2021