Jl. of Technology and Teacher Education (2019) **27**(2), 165-205

Computational Thinking, Mathematics, and Science: Elementary Teachers' Perspectives on Integration

KATHRYN M. RICH, AMAN YADAV, & CHRISTINA V. SCHWARZ
Michigan State University, USA
richkat3@msu.edu
ayadav@msu.edu
cschwarz@msu.edu

In order to create professional development experiences, curriculum materials, and policies that support elementary school teachers to embed computational thinking (CT) in their teaching, researchers and teacher educators must understand ways teachers see CT as connecting to their classroom practices. Taking the viewpoint that teachers' initial ideas about CT can serve as useful resources on which to build educational experiences, we interviewed 12 elementary school teachers to probe their understanding of six components of CT (abstraction, algorithmic thinking, automation, debugging, decomposition, and generalization) and how those components relate to their math and science teaching. Results suggested that teachers saw stronger connections between CT and their mathematics instruction than between CT and their science instruction. We also found that teachers draw upon their existing knowledge of CT-related terminology to make connections to their math and science instruction that could be leveraged in professional development. Teachers were, however, concerned about bringing CT into teaching due to limited class time and the difficulties of addressing high level CT in developmentally appropriate ways. We discuss these results and their implications future research and the design of professional development, sharing examples of how we used teachers' initial ideas as the foundation of a workshop introducing them to computational thinking.

The proliferation and ubiquity of computers and computing in everyday life has led to increasing calls and efforts to bring computer science (CS) education to every K-12 student in the United States (Grover & Pea, 2013; Wing, 2006). Advocates, including computer scientists (Wing, 2006), computer science educators (Barr & Stephenson, 2011; Yadav, Hong, & Stephenson, 2016), and developers of academic standards (College Board, 2017a), agree that computer science education must include more than just learning to program. While the idea of *computational thinking* (CT) has been around since the early 1960s, it has gained popularity as a name for the conceptual and practice-oriented elements of computer science education (Bocconi, Chioccariello, Dettori, Ferrari, & Engelhardt, 2016; Denning, 2017; Wing, 2006). As pointed out the White House (2016) *CS For All* initiative, in order to be active citizens in an increasingly computing-oriented world, students need to understand principles of how a computer works and kinds of problems that could be solved computationally. Just as well-rounded mathematics and science education includes attention to conceptual understanding and reasoning practices (Minner, Levy, & Century, 2010; National Research Council, 2001), well-rounded computer science education should include attention to the practices used in computer science. As such, one of the goals of the *CS For All* movement is to help every student become a computational thinker.

A key question in this movement is how to best prepare and support teachers in embedding computational thinking into their teaching practices (Bocconi et al., 2016; Yadav, Hong, & Stephenson, 2016). Several professional development efforts, often generalized across K-12, have been implemented to help teachers plan and enact effective CT instruction (e.g., Pollock et al., 2017; Reding & Dorn, 2017). This work suggested that teachers feel they need more support in understanding how they might bring CT to students (Pollock et al., 2017). There has been some research on preservice teachers' conceptions of computational thinking and how it could be embedded within K-12 classrooms (Yadav, Mayfield, Zhou, Hambrusch, & Korb, 2014; Yadav, Gretter, Good, & McLean, 2017); however, there is limited knowledge of how inservice teachers conceive of computational thinking ideas, especially in facilitating their curricular needs. Thus, more research is needed to understand the ways teachers think about the prospect of incorporating computational thinking into their teaching.

In this study, we explored the way elementary school teachers think about integrating computational thinking into their mathematics and science instruction. Taking a constructivist perspective, we view teachers' initial ideas about CT and how CT may be already happening in their classrooms as resources to be leveraged during professional development and other educational experiences for teachers. The goal of this study was to better understand elementary teachers' initial ideas about CT and identify potential challenges they anticipate in bringing CT to their classrooms. In this paper, we share our results and discuss their implications for the design of professional development by providing examples of how teachers' initial ideas informed the design of our professional development workshops.

BACKGROUND

In this section we review current discussions of computer science education and the role of teacher thinking, using this literature to justifying our choices to focus on computational thinking, elementary school, and integration with mathematics and science.

Computational Thinking

While specific definitions of computational thinking vary, it is generally agreed that CT involves processes central to computer science, such as creating *abstractions* based on only the most relevant information in a problem, developing *algorithms* to accomplish specific tasks, and *decomposing* problems into smaller and more tractable pieces (Grover & Pea, 2013; Wing, 2006; Yadav, Stephenson, & Hong, 2017). Those advocating for CT in K-12 argue that teaching computer science through CT has the potential to broaden access to computer science in a number of ways. First, a focus on CT can provide a way to illustrate how computing ideas are used in a variety of disciplines (Wing, 2006), allowing for students to see connections between computer science and their areas of disciplinary interest (Margolis, Estrella, Goode, Holme, & Nao, 2008). Further, a focus on CT allows for inquiry-based approaches to teaching computer science, where students identify problems, articulate and test strategies, and develop multiple representations of solutions (Goode, Chapman, & Margolis, 2012). These features make it possible to offer multiple entry points into problem-solving activities, thereby making computer science accessible to more students. Broad accessibility to computer science ideas is critical in today's increasingly computing-oriented world. Even if students choose not to pursue a career directly related to computer science, incorporation of CT into schooling also has the potential to prepare students to be engaged citizens and problem solvers (Gretter & Yadav, 2016).

In addition to making CS accessible to more students, a focus on CT also has the potential to diversify the students pursuing CS. From 2007 to 2015, women earned less than 20% of the computer and information sciences bachelor's degrees conferred each year (National Center for Education Statistics [NCES], 2015). A focus on CT has the potential to increase this percentage, as the broader applicability of computer science to other disciplines has been found to be an important factor in women pursuing CS (Google, 2014). Relatedly, in 2017, only 3.8% and 11.9% of the students who took the programming-focused Computer Science A (CSA) Advanced Placement exam were Black/African American or Hispanic/Latino, respectively (College Board, 2017b). By contrast, 7.1% and 19.4% of students who took the Computer Science Principles (CSP) Advanced Placement exam—a course with more focus on CT—were Black/African American or Hispanic/Latino (College Board, 2017b). These proportions are still far from being representative of the general population, but show an improvement from the programming-focused CSA course. Thus, a CT approach to expanding computer science education in K-12 has the potential to increase and diversify students interested in computer science.

The Importance of Teacher Knowledge and Thinking

While the importance of bringing CT to all students has been discussed widely, there are still questions about the kinds of knowledge teachers need to implement CT instruction and how to provide scaffolding to help teachers to infuse CT within K-12 subject areas. In his seminal essay, Shulman (1986) argued for the importance and complexity of the many kinds of knowledge teachers need, including content knowledge, pedagogical content knowledge, and curricular knowledge. Indeed, a wealth of prior research has demonstrated complex interactions between teachers' knowledge and their classroom practices (Toom, 2017). For example, Grossman (1989) found that early career English teachers who completed a teacher education program thought differently, as compared to teachers without formal teacher preparation, about the nature of English as as discipline and how students should approach it. These differences in how teachers thought about their subject area also led to differences in the way they organized their courses and interacted with students. In particular, the teachers with formal preparation prepared lessons that reflected knowledge of common student difficulties, such as lack of interest in literature, whereas the teachers without formal preparation were often surprised by such student difficulties when they came up in the classroom (Grossman, 1989). More recently, Ball, Thames, and Felts (2008) identified several specific subdomains of content knowledge and pedagogical content knowledge needed to manage the "recurrent tasks and problems of teaching mathematics" (p. 395). These subdomains include *knowledge of content and students, knowledge of content and teaching,* and *specialized content knowledge* (contrasted with common content knowledge).

In light of this research, we assert that in order to for teachers to teach CT effectively, they need educational experiences designed to increase their knowledge about CT content as well as how that content relates to their existing practices. Further, we agree with the other scholars who argue that understanding teachers' current thinking is a vital component of designing effective professional development (PD) experiences (Griffith, Ruan, Stepp, & Kimmel, 2014; Loucks-Horsley, Stiles, Mundry, Love, & Hewson, 2010) and educative curriculum materials (Davis & Krajcik, 2005) that will support particular kinds of teaching and learning in classrooms. We take a constructivist perspective on teacher learning and assume that educational experiences are most effective when they build on and leverage what teachers already know (Simon, 1995). Thus, our goal is to understand teachers' thinking about CT as a first step toward designing effective educational experiences for teachers that will support them in bringing computational thinking to their students.

Recently, researchers have begun to explore the role of teacher thinking specifically in relation to CT instruction. Yadav, Krist, Good, and Caeli (2018) found that professional development can help teachers better understand computational thinking and how CT could be helpful in their classroom. However, the authors argued that in order to measure shifts in teacher thinking we need to go beyond black-box assessments, such as closed-ended surveys, and use mechanisms that allow researchers to get more subtle and nuanced views of role of CT in classrooms. In another study, Hesteness, Ketelhut, McGinnis, and Plane (2018) used teacher drawings (about students engaged in science), written reflections, and focus group interviews to examine teacher thinking about introducing CT and related pedagogies within elementary classrooms. The authors found that teachers were able to draw upon their existing knowledge about the curriculum, teaching, and their students to identify how CT might be introduced in elementary classrooms to enhance student learning. We aim to build upon this research on teacher thinking about CT in this study. In the following sections, we discuss the reasons for our choice to focus on elementary school and integration of CT into mathematics and science.

CT in Elementary School

Computer science has typically been taught at the undergraduate level (Nager & Atkinson, 2016) and more recently in high school via the advanced placement Computer Science Principles course (Astrachan, Cuny, Stephenson, & Wilson, 2011). However, we need not wait until high school to introduce students to computer science ideas. Research from as early as the 1980s suggested that CS ideas are accessible to elementary-aged children (Papert, 1980; Battista & Clements, 1986). More recently, work has emerged suggesting that CT concepts are accessible to young children (e.g., Dwyer, Boe, Hill, Franklin, & Harlow, 2013; Fessakis, Gouli, & Mavroudi, 2013; Flannery & Bers, 2013; Seiter, 2015; Seiter & Foreman, 2013). Given this evidence that elementary students can successfully engage with CT ideas, it makes sense to start their work with CT early so students can begin building on their experiences. The complex nature of CS, particularly when its definition includes CT practices, suggests the need for a curriculum that spirals over time (Bocconi et al., 2016).

Teachers of all levels will need educational experiences that prepare them to effectively teach CT concepts. Much of the existing work on teacher thinking and professional development in CT has been generalized across K-12 (e.g., Pollock et al., 2017; Reding & Dorn, 2017), leaving developers little capacity to attend to how the needs and thinking of teachers vary across different grade bands. In this study, we focus specifically on understanding the needs and thinking of elementary school teachers in relation to CT.

While research on CT in elementary schools is sparse, there is some work emerging on elementary teacher professional development and how elementary teachers think about computational thinking. Yadav, Krist, Good, and Caeli (2018) used classroom teaching vignettes to examine how elementary teachers' understanding of computational thinking emerged over the course of a year. The results suggested that at the beginning of the professional development, teachers were more familiar with computational thinking ideas than expected; however, over the course of the year teachers' conceptions of CT evolved from "generalized, coarse-grained ideas (e.g., broadly defining CT as problem-solving) to more elaborated versions of these ideas" (p. 392). For example, before the PD, teachers said CT involved logical thinking, whereas after the PD they made more specific references to how CT involves conditional logic—a particular kind of logical thinking. In another study, Krist and colleagues (2017) presented a case study of how teachers recognize or take up CT as a part of their own science inquiry. The teachers in this project explicitly identified particular thinking strategies, such as using a flowchart to manage possible explanations, as "computational thinking" and took ownership of the strategies as they engaged in science inquiry.

Existing research on the concerns of elementary school teachers has also indicated that adding computer science to the elementary curriculum raises a practical problem of fitting it into the already full school day. In a recent study, elementary school teachers said that the demands of their current curriculum make it infeasible to devote several hours of instruction per week to an entirely new subject; however, teachers were willing and eager to implement CT to support another subject area (Israel, Pearson, Tapia, Wherfel, & Reese, 2015). Given this challenge for elementary teachers, we aim to develop a professional development program that supports elementary teachers in integrating CT into other subject areas. Thus, as a first step in the development of such a program, we focused this study specifically on understanding how elementary teachers relate CT ideas to their existing instructional practices in other subjects. In addition to addressing a practical need of elementary teachers, integration of CT into core subjects has the advantage of providing more equitable access to computing ideas for all students. If CT is taught as a separate subject, it runs the risk of being treated as an enrichment activity provided only to students who show high achievement in other subjects (Weintrop et al., 2016).

Connections between CT and Mathematics and Science

Much of the existing work on integrating CT into other subject has focused on mathematics and science, as the thinking practices of these disciplines bear many similarities to CT (Common Core State Standards Initiative [CCSSI], 2010; Next Generation Science Standards [NGSS], 2013). Table 1 shows an example of how two Computational Thinking Practices from the Advanced Placement Computer Science Principles curriculum framework (College Board, 2017a) map onto the first Common Core State Standard for Mathematical Practice, "Make sense of problems and persevere in solving them," (CCSSI, 2010) and one of the Next Generation Science Standards' Science and Engineering practices, "Developing and using models" (NGSS, 2013). Examination of these practices reveals some close similarities. For example, in CT, students evaluate and analyze their own and others' computational artifacts, and in mathematics, students analyze their own and others' approaches to solving problems (first row of Table 1). Relatedly, in CT, students develop computational models and use them to make predictions about the world, which can be seen as parallel to the science and engineering practice of developing and using models (second row of Table 1). The NGSS also include a practice called "Mathematical and computational thinking." We provide these examples only as illustrations of the most readily apparent connections. Certainly, more connections between computational thinking and the practices in mathematics and science can and should be made when they provide useful leverage points for integrating instruction.

Table 1

Computational thinking practices aligned with mathematics and science practices

CT Practices (College Board, 2017a)	Mathematics or Science Practice (CCSSI, 2010; NGSS, 2013)
Analyzing Problems and Artifacts. Students in this course ... evaluate and analyze their own computational work as well as the computational work others have produced. Students are expected to evaluate a proposed solution to a problem, locate and correct errors, explain how an artifact functions, and justify the appropriateness and correctness of a solution, model, or artifact.	**Make sense of problems and persevere in solving them.** Mathematically proficient students check their answers to problems using a different method, and they continually ask themselves, "Does this make sense?" They can understand the approaches of others to solving complex problems and identify correspondences between different approaches.
Abstracting. Students in this course use abstraction to develop models and simulations of natural and artificial phenomena, use them to make predictions about the world, and analyze their efficacy and validity.	**Developing and Using Models.** Models include diagrams, physical replicas, mathematical representations, analogies, and computer simulations. Models bring certain features into focus while obscuring others.

Research Questions

The goal of this exploratory study was not to examine teachers' conceptions or misconceptions about CT, but rather to dig deeper into their thinking on how CT could be integrated within their existing elementary school curricula. Given that some have argued for computational thinking as a precursor to computer science (Yadav, Krist, Good, & Caeli, 2018), it is important to better understand how teachers might see its role in elementary classrooms. We conceptualize our work not as identifying the ways in which teacher thinking differs from commonly accepted conceptualizations of CT, but rather as identifying similarities in their thinking and leveraging them as starting points for professional and curriculum development. We draw on a general resources perspective of teacher learning (Harlow, Bianchini, Swanson, & Dwyer, 2013) based on theories such as diSessa's (1993) Knowledge in Pieces (KiP) and assume that teachers possess resources, such as pieces of knowledge, that relate to CT, the ways they might incorporate it in their practice, and the potential barriers for doing so. We see the goal of this study as the identification of relevant knowledge resources teachers possess for the purpose of eventually leveraging them in the design of educational experiences that facilitate teachers' elaboration and connection of the resources into coherent knowledge for teaching CT.

This study aimed to answer the following research questions:

- How do elementary school teachers view computational thinking to be related to their mathematics and science teaching practices?
- What challenges do elementary school teachers anticipate in bringing CT into their math and science teaching?

We begin by explaining our methods for answering the above research questions. We then present the results of our analysis and describe the knowledge teachers already possess related to CT and its potential integration into elementary mathematics and science. We end the paper with a discussion of how these findings can inform the development of educational experience for teachers, providing examples to ways in which we used the results to inform the design of professional development workshops for in-service teachers.

METHODS

We used qualitative methods gain insights into how elementary school teachers think about integrating CT into their mathematics and science instruction (Bogdan & Biklen, 2007; Weiss, 1994). In this section, we provide information about the study participants and our data collection and analysis procedures.

Participants

The participants in this study were elementary school teachers sampled from a large urban intermediate school district (ISD) in the Midwestern United States. The ISD's leadership has recently forged a partnership with our research team. The partnership centers on a commitment to co-develop instructional activities by providing professional development for elementary school teachers focused on computational thinking. This study is, therefore, one aspect of a larger program of work, designed to advance the overall goal of bringing CT instruction to elementary school students in the ISD. The twelve teachers who participated in this study formed the first cohort of the project and were recruited through the ISD. The grade levels they were teaching in the 2017-2018 academic year, as well as some demographic information about their schools, are shown in Table 2. Their years of experience ranged from three to more than 20 years as classroom teachers. Most teachers taught all subjects to a single class, although a few taught math and science to multiple classes as a part of team teaching in their school. None of the teachers had any prior experience with CT. Each teacher received a $50 gift card for their participation in the interview.

Table 2
Study Participants and School Demographics

<table>
<tr><th>Teacher Pseudonym</th><th>Grade</th><th>School Pseudonym</th><th>%Non-white</th><th>% Free and Reduced Lunch</th><th>% English Language Learners</th></tr>
<tr><td>Abbott</td><td>4</td><td rowspan="2">Newberry</td><td rowspan="2">52.7</td><td rowspan="2">76.0</td><td rowspan="2">3.9</td></tr>
<tr><td>Bennett</td><td>5</td></tr>
<tr><td>Carson</td><td>4</td><td rowspan="2">Oakwood</td><td rowspan="2">26.3</td><td rowspan="2">84.8</td><td rowspan="2">75.9</td></tr>
<tr><td>Duncan</td><td>Media specialist</td></tr>
<tr><td>Edwards</td><td>5</td><td rowspan="4">Potter</td><td rowspan="4">66.3</td><td rowspan="4">58.6</td><td rowspan="4">**</td></tr>
<tr><td>Franks</td><td>3</td></tr>
<tr><td>Goodson</td><td>5</td></tr>
<tr><td>Hastings</td><td>4</td></tr>
<tr><td>Jackson</td><td>4</td><td rowspan="2">Richland</td><td rowspan="2">41.8</td><td rowspan="2">62.5</td><td rowspan="2">**</td></tr>
<tr><td>Keller</td><td>5</td></tr>
<tr><td>Lowry</td><td>5</td><td>Smith</td><td>87.3</td><td>65.4</td><td>31.5</td></tr>
<tr><td>Miller</td><td>1</td><td>Tribune</td><td>42.7</td><td>69.0</td><td>3.1</td></tr>
</table>

**Percent ELL for these schools was not available.

Procedures

Interview format and content. The first author, who had no previous relationship with the teachers, conducted a semi-structured interview (Lichtman, 2006) with each teacher. The interview protocol is included in Appendix A. The interviews focused on the meaning of computational thinking, its use in elementary school classrooms, and the participant's ideas about strategies for integrating CT into mathematics and science instruction and the potential benefits and challenges of doing so. The interviewer began each interview by asking about the participant's current teaching situation, then asking for the participants' first thoughts about the meaning of the term *computational thinking*. The remainder of the interview was structured around a handout, provided in Appendix B, that gave a brief description of key components of CT described in a recent report on computational thinking in education (Bocconi et al., 2016): abstraction, algorithmic thinking, automation, debugging, decomposition, and generalization. Teachers were asked to review the handout, talk about any examples of CT they thought were already happening in their classrooms, and share teaching strategies they use to stimulate similar thinking in their students. They were also asked to compare the CT practices to mathematics and science practices identified in widely-used standards documents, also provided on handouts (provided in Appendices C and D; CCSSI, 2010; NGSS, 2013). The conversation also covered teachers' thinking about the short- and long-term benefits of teaching mathematics and science through CT and any challenges they foresaw in doing so.

With participants' consent, the interviewer audio recorded the interviews. As soon as possible after the conclusion of each interview, the interviewer wrote a brief set of notes (as advocated by Patton, 1987) about the themes that came up in the interview. Each interview was later transcribed for further analysis.

Analysis. Two researchers (the first author and a research assistant) collaboratively used a process of open coding to capture initial themes in two of the interviews. Using the general categories of codes developed through this initial coding, the first author independently coded the remaining transcripts. After examining all instances of each code, the same researcher expanded and collapsed codes, created a code-book, and recoded interviews as necessary. The second author then coded relevant sections of the transcripts using the code-book. This second coding resulted in 81% inter-rater agreement, which we deemed sufficient to proceed with the original coding. The first author iteratively reviewed the codes to examine teachers' ideas about the overall relationships among CT, mathematics, and science; their thinking about how the six CT components relate to their mathematics and science teaching; and their thoughts about the challenges of incorporating CT into their teaching. This process led to the four findings outlined in the next section. The sections of the code-book that apply to each finding are included in tables in each finding section.

RESULTS

We organize our results into four sections, each detailing a particular finding. The first finding relates to teachers' overall conceptions of CT and how it relates specifically to mathematics and science, with particular attention to the ways that mathematics and science instruction play out in their classrooms. The second and third findings related to the connections teachers made between their existing practices and each of the six components of CT presented on the interview handout. The last finding lays out two major challenges discussed by teachers in bringing CT into their instruction. We interpret each of these findings as knowledge resources that could be leveraged in the design of educational experiences for teachers (Harlow et al., 2013). We return to this idea in the Discussion.

Finding #1: Teachers think of CT as problem solving with stronger connections to their mathematics instruction than to their science instruction.

Teachers' responses to the question, "If a colleague asked you to explain what computational thinking is, what would you say?" fell into two categories: CT as problem solving and CT as mathematics. Examples of excerpts coded with each code are shown in Table 3.

Table 3
Codes for Teachers' Descriptions of CT

Code	Example
CT as problem solving	"Yeah, yeah, I think it's a lot of problem solving skills."
CT as mathematics	"I automatically think math and how we're going to solve things and work things out."

Eight of the 12 teachers described computational thinking as a kind of problem solving. This is highlighted by Hastings, who described CT as "the ability to understand and break down concepts to solve problems." Duncan talked about CT as a problem solving process across subjects, saying, "And I think it is like, what I think about it, it doesn't just pertain to math and science. I think about it how it pertains to lots of different things, like thinking about problems around the world."

Nine teachers described computational thinking as being related to mathematics. For example, Goodson said computational thinking was "math that's done with values in an operation, you know, clearly indicated." Another teacher, Lowry, described CT as "thinking in mathematical terms." Teachers also responded in ways that combined both of these categories, making references to problem solving in the context of mathematics. For example, Keller stated:

> I mean in the most basic level it would be just the ability to problem-solve through multiplication, division, knowing—choosing operations, being able to use them correctly, come to the answers—even the multi-step problems, just understanding what they need to do from point A to point B to point C to getting that final answer.

In response to this initial question, none of the teachers spoke about how computational thinking connected to science, which suggests that, at least initially, elementary school teachers were able to draw a stronger connection between CT and mathematics than between CT and science.

Aware that this pattern of findings had been demonstrated in other studies, we took advantage of our interview setting to further probe teachers' thinking about CT specifically as connected to their instructional practices in mathematics and science. First, we examined how their responses changed after reading a more thorough description of CT on the handout. After teachers had reviewed the CT handout, almost half of them (N = 5) saw computational thinking as a general process that could be applied across subjects. For example, when considering what the benefits of teaching through CT might be for students, Edwards said, "So just computational thinking I think would be excellent for math and science or for any subject. Especially with the computer science involved, because there's so much technology."

Despite this greater tendency to generalize across subjects after reading the handout, however, when pressed for specifics they still noted fewer connections between CT and their science instruction than between CT and their math instruction. For example, when specifically asked to comment on how the ideas on the CT handout related to her science teaching, Hastings said, "I thought this was a math deal. I've never thought about it [how the CT ideas would connect to science] ... I could see generalization there. The other stuff not so much, though." Another teacher, Jackson, felt like the big idea of asking questions, which she saw as key in science, was missing from the CT handout.

As a final way to probe teachers' thinking about connections between CT and their math and science instruction, we examined their responses when asked to note similarities and differences between the CT handout and the lists of mathematics and science practices in the other handouts. In the set of 12 interviews, teachers made 17 specific connections between the CT ideas and the Standards for Mathematical Practice (CCSSI, 2010). For example, Franks said of the CCSS practices, "I even feel like that model—like 'Model [with] mathematics' and 'Use appropriate tools strategically' probably goes with that [CT component of] algorithmic thinking." By contrast, teachers made only three specific connections between the CT ideas and the Science and Engineering Practices (NGSS, 2013). For example, Abbott said of the NGSS practices, "'Constructing explanations and designing solutions' seems like, oh, like kind of, [the CT component of] abstraction." The large difference between the number of math connections and the number of science connections seems to further support the idea that teachers, overall, more easily connect CT to their mathematics instruction than to their science instruction. One teacher, Franks, noted this directly, saying the following about the CT handout: "I feel like these words are more meaningful to me in a mathematics setting, and I never really have thought about them in a scientific setting."

Finding #2: Teachers identified elements of their practice that can serve as productive starting points for learning and instruction of decomposition, algorithmic thinking, and automation.

We also focused our analysis on teachers' characterizations of how they saw the CT components outlined on the handout—abstraction, algorithmic thinking, automation, debugging, decomposition, and generalization—happening in their classrooms. We categorized teachers' statements about each of the CT components into two to five groups. For three of the six CT components (algorithmic thinking, automation, and decomposition), teachers' responses reflected the influence of shared vocabulary between mathematics content and pedagogy and computational thinking. The categories into which we coded teachers' descriptions of these three components are shown in Table 4. We discuss teachers' descriptions of these three components in this section.

Algorithmic thinking. In the case of algorithmic thinking, two-thirds of the teachers (N = 8) drew a connection to their students' use of standard algorithms for the four basic operations in mathematics. It was common for teachers to specifically contrast the use of standard algorithms from other methods of doing arithmetic, relating the former to the algorithmic thinking component of CT. For example, Abbott said her students were "certainly doing algorithmic thinking" because:

> we have done the, the breakdown, we've done it some other ways but we do the traditional algorithm both for multiplication and division. So it's just clear, you know, the steps that I have anchor charts all over and they're able to see the steps that we're, that we're using.

This connection to algorithmic thinking expressed by more than half of the teachers seems to be a reflection of the primary use of the word *algorithm* in mathematics education to refer to standardized, traditional methods for performing arithmetic computations. In research on reform-oriented mathematics pedagogy, student-invented procedures are often contrasted with these standard algorithms and referred to using the more generic term *strategies* (e.g., Carpenter, Franke, Jacobs, Fennema, & Empson, 1998; Forman & Ansell, 2001).

Table 4

Teachers' Connections to CT Components Reflecting Shared Mathematics Vocabulary

Overarching CT Component	**Codes for Teacher Descriptions**	**Examples**
Algorithmic thinking	Following steps	"Algorithmic thinking- I mean we are like knee deep- I mean no no no- they're neck deep in the traditional division algorithm."
	Discovering and explaining strategies	"We do algorithmic thinking, I mean the [curriculum] is set up where, you know, it does give the students a chance to discover and then explain."
Automation	Automatic recall	"As far as automation, the first thing that's coming to me ... everyday they have to do a fact fluency."
	Offloading to technology	"We're going to learn how to make technology do stuff for us… you know, like the automation part of it."
	Coding	"Automation- I don't think I've done very much, but I know our STEAM teacher has when she's taught coding."

Overarching CT Component	**Codes for Teacher Descriptions**	**Examples**
Decomposition	Breaking down numbers to compute	"So we do matrix multiplication, box multiplication. So they decompose into place-value pieces and then distribute."
	Breaking down a problem to plan	"[O]ur kids live in the seven habits. So like Habit 2 is 'begin with the end in mind' and thinking about like the big problem to solve. Habit 3 is 'put first things first,' so it's like putting a plan together."
	Breaking down systems	"[D]ecomposition, I guess I'm thinking the life cycle and the- I mean breaking it all down- and breaking the food chains down."

A quarter of the teachers (N = 3) went beyond the idea that algorithmic thinking could be described as students following prescribed steps and included the idea that algorithmic thinking involved *developing* step-by-step approaches. Interestingly, rather than connecting to something that already happens in her classroom, one teacher (Miller) noted she wished more development of algorithms would happen: "Let me see- for the algorithmic thinking, honestly I wish that I would have my students do more of that, where like they give me the steps to do it or like plan out the steps themselves of how they want to solve it."

Automation. Teachers' characterizations of automation similarly reflected a particular use of the term *automaticity* frequently used within mathematics education. Specifically, half of the teachers referenced fast or automatic recall of basic facts—commonly discussed in mathematics education research as fact fluency (Baroody, Eiland, Purpura, & Reid, 2013)—when connecting automation to their practice. For example, Franks said,

> [W]e spend a lot of time with automation. Just trying to get the kids to understand something quickly so that we can move on to something more complex. So memorize those multiplication facts so that we can do a two digit by two digit multiplication problem. Instead of, you know, instead of skip counting, instead of adding over and over and again, instead of getting counters out at some point we want something to be automatic.

Teachers' focus on connecting this particular idea related to automation to their instruction is not surprising given the heavy focus on automatic fact recall in elementary school mathematics standards and curricula (CCSSI, 2010).

Three teachers expressed different understandings of automation. Two discussed the idea of offloading certain tasks to technology when thinking about automation. For example, Jackson said,

> Automation. I'm seeing that as basically your ability to use like a technology- computer technology. Or well, we use the computer. We don't use really calculators anymore because that's on the computers - well we use our computers like all the time. We use them to- the kids will use them to help when they're- you know if they're trying to solve like a more challenging problem but like they're still struggling with math facts.

A third teacher, Bennett, acknowledged a connection between automation and coding, but noted it was the STEAM teacher who addressed this with students: "Automation- I don't think I've done very much, but I know our STEAM teacher has when she's taught coding."

Decomposition. Five teachers related the CT component of decomposition to multiplication and division strategies that rely on decomposition of numbers. For example, Abbott said, "Decomposition, again I'm thinking of math and decomposing numbers so that they- especially, we really go over place value when we're multiplying and dividing so they see what- how they really are fitting into the problems." It is not surprising that this was the most common connection teachers made between decomposition and their teaching practices, as the Common Core State Standards (2010) contain multiple references to decomposing numbers and geometric figures.

In addition, four other teachers also discussed decomposition as breaking down problems or systems. Two teachers discussed decomposition as breaking down problems as a part of developing a plan for solving them. This was highlighted by Jackson, who stated, "Decomposition. I mean exactly what it is. Decomposing the problems, breaking them down, identifying the steps that you need." Two other teachers mentioned decomposition as breaking down systems within science to understand their parts. One referenced breaking down a food chain and the other referenced breaking down different parts of a solution in chemistry.

Finding #3: Teachers made the fewest connections to debugging and abstraction, and the most connections to generalization.

For the remaining three components of CT (debugging, abstraction, and generalization), teachers' responses did not reflect prior knowledge of use of similar terms in mathematics or science. In the cases of debugging and abstraction, fewer teachers spoke about connections to their math and science teaching than spoke about connections to automation, algorithmic thinking, or decomposition. Generalization, on the other hand, was the most commonly discussed CT component. The codes we used to categorize teachers' descriptions of these CT components are shown in Table 5. We discuss teachers' characterizations of these three CT components in this section.

Debugging. Only five of the twelve teachers connected debugging to their math and science teaching when reading the CT handout. Two of these teachers related debugging to checking whether the answer to a mathematics problem makes sense. For example, Hastings said, "So debugging for me would be estimating before you do the problem to make sure it makes sense when you get your response." Three other teachers related debugging to more general processes of sensemaking, such as analyzing a problem or working through a complex task. For example, Edwards related debugging to performance tasks on standardized assessments:

> [W]ell performance tasks is like, you know, they give you a situation and you have all these different things about it that you have to answer. There's writing involved, there's … So I'm thinking that is a- that as debugging. I guess, I don't know. That would be the tracing and logical thinking.

Table 5

Teachers' Connections to CT Component Not Reflecting Shared Mathematics Vocabulary

Overarching CT Component	Codes for Teacher Descriptions	Examples
Debugging	Checking an answer	"To have like a remainder that is larger than your divisor so they might say well that doesn't work out so then they might go back and then work through and test it."
	Problem analysis	"A little bit with the debugging, I guess more talking about like different problems in math. It's like, what are they asking us- is it the quotient that's the answer? Is the remainder the answer? It's like what are they asking us?"

Overarching CT Component	**Codes for Teacher Descriptions**	**Examples**
Abstraction	Analyzing	"[T]hey have to talk through it, and like one of the options is eliminated, how is that eliminated, how do you know."
	Early step in problem solving	"I think that- at least as it's phrased here, the abstraction step is written here they're take- often is the critical thinking before you write anything down, maybe before you even do any math."
	Eliminating repeated steps	"[A] couple of the group said, well can't we just keep adding the next object in and just continue to measure the difference of the water?" (See further detail in text below.)
	Expressing ideas	"So like where they kind of move from seeing something in their brain to being able to express it."
	Removing irrelevant information	"And more problems- taking things out that don't mean anything to the problem, you know?"
Generalization	Connecting to prior knowledge	"Generalization, when I think of this, based on the first couple of words, I kind of, kind of, like the schema that comes to mind. Like what is their schema. What knowledge do they already have?"
	Noticing and using patterns	"And generalization, I would think, so for sure, you know we try to identify like patterns and how things are similar and how things connect."
	Drawing conclusions in science	"I think that could be the same in science as well as math but what we're asking them to come up with is this bigger understanding of whatever we're teaching so I could see generalization there."

Abstraction. Seven of the teachers made a connection between abstraction and some part of their math and science teaching, but these connections were varied with no particular trends in teacher thinking. Two teachers connected abstraction to general processes of analysis. For example, Edwards said, "Well for abstraction, I'm thinking of science and we- we did a unit on rocks and fossils, so the kids, you know, got a chance to look at those and talk about those and analyze them." Two teachers described abstraction as an early part of problem solving. Goodson, for example, said, "abstraction, to me would come under the read and understand [the problem]." Another teacher, Keller, mentioned a time when her students realized they could eliminate repeated steps when measuring volume by displacement:

> [M]aybe the abstraction, like I know we were doing measuring volume of objects last week. And we were putting things in the graduated cylinder, and they were putting one thing in and then pouring everything back out. And a couple of the group said, well can't we just keep adding the next object in and just continue to measure the difference of the water? And I thought well that was really great, because I said, well, yeah think about how much time that's going to save you, to do this simple- just add add add, instead of undoing and redoing it every single time.

Franks described abstraction as the act of expressing ideas: "And then I like that abstraction. So like where they kind of move from seeing something in their brain to being able to express it down in there thinking." Finally, Abbott described abstraction as a process of eliminating irrelevant information from a story problem: "[W]hen I'm reading this- choosing the right details to hide- my first thought is, I'm focusing on math … taking things out that don't mean anything to the problem, you know?"

Generalization. All but two of the teachers connected generalization to some aspect of their math and science teaching, and their connections were less varied than for debugging and abstraction. Six teachers related generalization to moments when their students related new information to their prior knowledge. For example, Bennett said,

> Generalization- how is it different? Oh we talked about like how is this similar to a problem I've already solved, how is it different? We've talked about that when we've been talking about division. They'll- they'll say oh I want to- you know I have a connection to one of the ones, you know, a problem that we've done before.

While some teachers talked about connections to things students learned within a discipline, others discussed generalization as making connections across disciplines. This was highlighted by Miller:

> [W]e were talking about - what was it - artifacts in social studies, and we were learning about like past and present and how we learn about the past and stuff like that. And they tied that into like stories we were reading in English and so they could generalize, like oh this is, that is and this is what's going on.

Half of the teachers (N = 6) also spoke about generalization as a process of noticing and applying patterns. For example, Miller described a moment when her students noticed a pattern relating addition and subtraction facts:

> In math, they like made the connection and saw the pattern that we started learning about fact families. We weren't quite there yet, but they put it together that like, ten minus five they knew was five because I know five plus five is ten. And like they made those generalizations.

Five out of six of these descriptions of using patterns related to mathematics, and one related to social studies.

Although science did not come up in teachers' discussions of patterns, three teachers did relate generalization to science in a different way. Specifically, they characterized generalization as drawing conclusions when doing science. For example, Goodson said,

> [Y]ou could've told them at the beginning what happens when you mix an acid with an acid. But for them to derive their own generalization. The whole idea- that's why they did the experiment. So you've got to make sure it... at the end they come up with a generalization.

Finding #4: The challenges of teaching math and science through CT most anticipated by teachers related to limited class time and developmental appropriateness.

When discussing challenges of bringing CT into elementary math and science teaching, teachers brought up several concerns that are common to most classroom initiatives, such as lack of parental support, potential language barriers, and student motivation. There were two categories of common concerns that related more specifically to CT, summarized in Table 6.

First, seven of the teachers expressed concern about the limited class time available. For example, Carson said her main challenge in teaching through CT would be:

> the balance of everything. The time allowed for it. With our demographic here we also have an hour a day for reading intervention, and so that just, you know, it's an hour you're taking from classroom instruction. It's helpful of course but I just- I already have trouble with time, and so to be able to really integrate it well with what I'm already doing, because that would be I think the only way to kind of be able to do everything.

Table 6

Codes for Teachers' Concerns about Bringing CT to Their Math and Science Instruction

Code	Example
Time	"I just don't think about these things as I'm teaching you know this- there's just so much to do and so little time."
Developmental appropriateness	"The most challenging thing about these ideas is that it is a very mature way of looking at learning."

Despite acknowledging that the increased amount of time needed to teach through CT might be valuable for students, teachers expressed a need to balance that with the need to prepare them for standardized tests. For example, Edwards noted that a challenge would be, "the planning part and figuring out what is the best for the kids, because there's so much ... we look at the test, which teachers- I typically don't like doing that. But I know they have to be prepared."

Second, nine of the teachers expressed concerns about the abstract nature or high levels of thinking espoused in CT, and how those ideas could be made developmentally appropriate for students in elementary school. Abbott said she was "kind of anxious to see how this all fits into the elementary. … it all just seems so high." Several teachers worried that students' lack of basic skills would hinder their ability to think computationally. Some teachers pointed out that the transfer of CT skills to various contexts was likely to be difficult, as they have found it difficult to help students transfer other skills. As Hastings said, "it's hard to put it all together."

DISCUSSION

Our study revealed elementary teachers' knowledge and resources that relate to the prospect of integrating CT into their mathematics and science instruction. Teachers in our study understood computational thinking as a kind of problem solving, making more connections to their mathematics teaching than to their science teaching. For three components of CT—decomposition, automation, and algorithmic thinking—teachers' connections to their mathematics instruction reflected their prior knowledge about how similar terminology is used in common mathematics pedagogical and instructional practices. For the other three components of CT—abstraction, debugging, and generalization—teachers did not seem to leverage particular knowledge of similar terms related to mathematics instruction. When discussing anticipated challenges about integrating CT into their teaching, teachers' most often discussed limited classroom time and concerns about making the high-level thinking involved in CT accessible to their young students.

Teachers' conceptions of CT as problem-solving and as connected to mathematics echo results of previous work with preservice (Yadav, Zhou, Mayfield, Hambrusch, & Korb, 2011) and inservice teachers (Sands, Yadav, & Good, 2018). The current study extends that work by going beyond surveys to use in-depth interviews to probe teacher thinking about specific components of CT and how those components relate to their existing practices. It is important to understand how elementary school teachers describe the ways in which their students might engage in CT to provide effective support for professional development (Guskey, 2002) and educative curriculum materials (Davis & Krajcik, 2005). The results of this study highlight several knowledge resources possessed by elementary teachers that could be leveraged in educational experiences (Hestness, Ketelhut, McGinnis, & Plane, 2018) to help teachers build and refine their existing conceptions of CT (Simon, 1995; Yadav, Krist, Good, & Caeli, 2018). We discuss specific ways teacher educators might utilize our results in the sections that follow. In each section, we describe a preliminary step we have taken to use the information derived from this study to inform our work with teachers.

Supporting Connections between CT and Existing Mathematics and Science Instruction

Teachers in this study made many connections between CT components and aspects of their mathematics teaching, whereas they made far fewer connections to their science teaching. There are several possible explanations for this. Given the heavy focus on mathematics and literacy in most elementary school curricula, it may be that the lower number of connections teachers made to science reflect teachers' lower level of content and pedagogical knowledge for science than for mathematics. Alternatively, common discussions of science instruction and pedagogy may simply share fewer common terms with CT than does mathematics, leaving teachers less able to readily see connections. For example, teachers did not readily see connections between scientific modeling and abstraction. By contrast, teachers were readily able to connect problem decomposition to breaking down problems in mathematics.

From a mathematics perspective, this finding suggests that integration of CT into elementary mathematics teaching is a tractable approach for exposing students to CT early in their school experiences, although more research is still needed to effective ways to leverage teachers' mathematics and CT connections in the classroom. From a science perspective, the finding suggests that curriculum materials and professional development aimed at helping elementary teachers infuse CT into their science teaching will likely need to include activities supporting both teachers' CT knowledge and strategies for how to think about and incorporate CT into science. There is great opportunity to do so given the emphasis on scientific modeling and computational thinking in the Next Generation Science Standards.

Preliminary steps for professional development. In an early professional development session, our partner teachers engaged in a science activity involving representations and modeling to help understand the role of abstraction and patterns in science. In particular, we asked teachers to imagine the phenomenon of someone sitting next to the exit of a water slide and to figure out "Why didn't [the person] get splashed every time someone came down the water slide?" This phenomena addresses NGSS standard 4-PS3-3 "Ask questions and predict outcomes about the changes in energy that occur when objects collide."

In the professional development workshop, teachers were asked to design and conduct an experiment, collect data, and present the data in some way to answer the question. They had access to marbles with different sizes and weights, ramps, cups, water, and additional equipment so that they could design environments with some similarities and differences to the water slide context (*abstracting* relevant components).Then they ran experiments to collect and represent data about objects (marbles) going down a slide and how they might impact an object (e.g., a cup, water) at the end of the slide. The data collection and representation involved *patterning*. Further, the problem had to be *decomposed* to determine what aspects (e.g., size of marble, height is slide) were critical to be tested and groups had to *debug* their testable environment to collect usable and meaningful data. We view these experiences embedding CT ideas as the first step for teachers to see how they could bring CT ideas into their science lessons.

Considering the Compatibility between Shared Vocabulary

Many of the teachers readily made connections between familiar mathematical vocabulary and the CT components of algorithmic thinking, automation, and decomposition. This connects to the work of other scholars who have attempted to identify both points of compatibility and points of departure between mathematical and computational thinking (Rich, Spaepen, Strickland, & Moran, 2019). Professional development and curriculum materials aimed at helping teachers connect CT to their existing teaching practices should, therefore, take into account the meanings of that terminology used in mathematics standards and pedagogical approaches and consider how best to build on teachers' existing knowledge and instructional practices.

Decomposition. In the case of decomposition, teachers commonly referenced the idea of breaking down numbers into parts in order to aid computation. Students might multiply 23 by 5, for example, by thinking of 23 as 20 + 3, and multiplying 20 * 5 and 3 * 5 separately. This mathematical idea of breaking a complex problem like 23 * 5 into simpler problems, 20 * 5 and 3 * 5, maps well onto the CT idea of problem decomposition as described in the K-12 CS Framework (2016): "Decompose complex real-world problems into manageable subproblems." Thus teachers' existing knowledge of decomposition in mathematics may serve as a helpful leverage point in helping teachers incorporate the CT of decomposition into their teaching.

One interesting thing to note is that teachers in our study typically talked about decomposition of *numbers* rather than decomposition of *problems.* Although the decomposed numbers were discussed as useful for solving problems, teachers did not explicitly talk about decomposing the problem itself. This emphasis on decomposition of numbers is reflected in the Common Core Standards for Mathematics (CCSSI, 2010), as well. For example, a first-grade standard requires students to decompose numbers into tens and ones and a fourth grade standard requires students to decompose non-unit fractions into unit fractions. In order to help teachers leverage their existing instructional emphasis on decomposition in mathematics give students exposure to CT, professional development and curriculum materials should help teachers see the ways in which decompositions of numbers can be applied to decompose problems. Emphasis should be placed not only on decomposing 23 into 20 + 3, but on decomposing 23 * 5 into 20 * 5 and 3 * 5.

Preliminary steps for professional development. To help our teacher partners focus on decomposition of problems, rather than numbers, we first had them complete an activity involving a problem that could be decomposed in multiple ways without specific attention to decomposing a number. Teachers worked in a group to place a set of fraction cards on a number line. In the subsequent discussion, teachers shared the various ways they decomposed the problem: thinking about order first, then spacing; handling the mixed numbers first, then the improper fractions; and so on. We later discussed how decomposing a number such as 23 before multiplying it by 5 allows students to focus their attention on one thing at a time: handling the tens first (20 * 5), then handling the ones (3 * 5). We also created a lesson screening tool to help teachers identify aspects of CT within their existing lessons (Yadav, Larimore, Rich, & Schwarz, 2019). In a later workshop, teachers used this tool to design CT-enhanced lessons. The guiding questions for identifying decomposition were: "Is there a complex task or situation that students could break down? Can the task or situation be broken down in multiple ways?" Our recordings of this workshop suggested these questions help teachers keep decomposition of *problems* in mind, even as they saw and discussed examples of decomposition of numbers.

Algorithmic thinking. In the case of algorithmic thinking, many teachers' responses reflected their familiarity with the word *algorithm* as a predefined procedure. This use is common in mathematics education; indeed, the glossary in the CCSS-M defines *computational algorithm* as follows: "A set of predefined steps applicable to a class of problems that gives the correct result in every case when the steps are carried out correctly" (CCSSI, 2010). Likely because of this emphasis, teachers tended to connect the CT component of algorithmic thinking to cases when their students follow predetermined steps to solve mathematics problems.

While this idea is not entirely incompatible with how algorithmic thinking applies to computer science, it tends to be missing key elements important in computer science, such as algorithm development and algorithm evaluation. The CS K-12 Framework (2016) suggested that by the end of grade 2, students should understand that "People follow and create processes as part of daily life." The inclusion of the word *create* extends algorithmic thinking beyond the instructional contexts discussed by most teachers in this study. Professional development facilitators and curriculum developers seeking to help teachers connect CT to their practice should challenge teachers' ideas of algorithmic thinking as *following* steps, and expand it to include *creation* of those steps. A few teachers did mention the idea of students developing their own steps when discussing algorithmic thinking, suggesting the feasibility of helping elementary school teachers to develop a more expansive definition of algorithmic thinking and applying it in their teaching. Prior research on how to support teachers in facilitating student invention of methods for performing mathematical operations (e.g., Carpenter, Fennema, Franke, Levi, & Empson, 2000) could also inform professional development to help teachers see how student-invented procedures connect to algorithmic thinking.

Preliminary steps for professional development. Due to teachers' strong association of the word *algorithm* to predefined procedures, we elected not to make algorithmic thinking one of our initial focus CT components. Teachers designed their CT-enhanced mathematics and science lessons without attention to algorithmic thinking. After an initial round of implementation, however, we chose to introduce the idea of developing algorithms not within the context procedures for of the four basic arithmetic operations, but rather in the context of navigating a number grid. Teachers first developed directions for moving from one number on the grid to another using only the commands +/- 1 or +/- 10. They then translated the same commands into directions for a robot, using moving and turning. We plan to later connect this idea of developing directions to other contexts, including student-invented strategies for the four operations.

Automation. In the case of automation, teachers most often made connections to automatic fact recall, or supporting students to memorize basic math facts and produce the answers automatically, without relying on counting or other strategies that use cognitive resources (Baroody, Eiland, Purpura, & Reid, 2013). The ideas of quickly producing answers and saving labor are also captured in the CT-oriented description of automation: “Automation is a labour saving process in which a computer is instructed to execute a set of repetitive tasks quickly” (Bocconi et al., 2016, p. 18). The key difference between automatic fact recall and CT automation is that in the former, students are producing the automated result, and in the latter, computers are producing the automated result. Curriculum developers and professional development facilitators should point out this difference to teachers and provide strategies to help them make the difference clear to their students. For example, teachers could emphasize the idea of saving effort, rather than the idea of memorization, when discussing fact automaticity with students. A few teachers discussed automation as offloading to technology. This idea could serve as a starting point for discussing automated processed executed by a computer rather than a student.

Preliminary steps for professional development. In an effort to leverage the connections teachers saw between CT components and their existing mathematics teaching practices, we began our work with them using an unplugged approach. Thus, all the initial activities we co-developed with teachers did not involve technology, and we have not yet explored the idea of helping teachers connect what students can do automatically to what computers can be programmed to do automatically. As we shift from unplugged to plugged activities, we plan to leverage the ideas some teachers expressed about offloading to technology to discuss automation and how it relates (or does not relate) to fact fluency.

Connecting the Familiar to the Unfamiliar

Teachers in this study made the fewest connections to the CT components of debugging and abstraction. This raises the question of how to help teachers connect these two concepts to something already in their practice. We present some initial thoughts in the sections that follow.

Debugging. We conducted four pilot interviews before beginning our official data collection that shed light on how teachers might better connect debugging to their existing mathematics teaching practices. Three of the four pilot interviewees mentioned having their students examine their own and other students' arithmetic work to find, analyze, and correct errors. This activity seems closely connected to debugging. Interestingly, though, the CT handout to which the pilot teachers responded did not explicitly mention debugging, but did mention that students should "locate and correct errors" as a part of analyzing problems and artifacts. The CT handout to which the main study interviewees responded *did* explicitly mention debugging, but did not use the language "locate and correct errors" in its description. This contrast suggests that curriculum and professional development should provide a definition of debugging that makes the connection between bugs and errors explicit to teachers. The research literature on students' arithmetic methods, which often contains references to "buggy algorithms" (Carpenter et al., 1998, p. 6) may be helpful in this regard.

Preliminary steps for professional development. The CT screener tool mentioned above also provides questions to help teachers identify opportunities for debugging in their existing lessons. The questions are, "Do students have opportunities to reflect upon their work? Do they have opportunities to revise their thinking or make improvements?" Our recordings of the workshops where teachers use the CT screener tool contain some productive conversations among teachers about how to intentionally build in time and opportunities for students to both catch errors and consider how to fix them. Strategies teachers had for supporting debugging included providing completed examples for students to compare to their own work and providing sufficient space for recording work on student pages.

Abstraction. In the case of abstraction, it is less clear how to help teachers connect the idea to their mathematics and science teaching practices. Given that almost all teachers in this study connected the CT component of generalization to their existing mathematics or science teaching practices, curriculum developers and professional development facilitators might consider leveraging connections between generalization and abstraction. For example, several teachers discussed recognizing and applying patterns in mathematics as a form of generalization; the general form of a pattern could be discussed as an abstraction. Further, the practices of developing scientific models (both conceptual models and data models) involves focusing on most relevant details of the phenomena. This relates to abstraction and can be used as a bridge to help teachers bring abstraction into elementary science.

Preliminary steps for professional development. The description of abstraction introduced to teachers in the interviews was as follows: “Abstraction is the process of making an artifact more understandable through reducing the unnecessary detail. The skill in abstraction is in choosing the right detail to hide so that the problem becomes easier, without losing anything that is important” (Bocconi et al., 2016). Because teachers did not readily see abstraction within their own teaching practices, we introduced them to abstraction in a context outside of mathematics: creating and using maps. Teachers discussed what is or is not important to show on different kinds of maps. On a street map, for example, street names are important but the trees along the street likely are not. On a subway map, how stations are connected is important, but the exact distances between them are not.

After this discussion, we encouraged teachers to think about how different mathematical representations show or hide important information. For example, we discussed how the symbolic form of a fraction, such as $\frac{2}{3}$, does not show the idea of equal parts, an important aspect of the part-whole interpretation of a fraction. To help teachers see when they could support students in identifying important information in a representation or situation, we included abstraction questions on the CT screener tool that said, “Do students identify key information in the task? Do they use representations or other tools to reduce complexity?” Our recordings of the workshops where teachers used this tool reveal that teachers engaged in rich conversations about what various mathematical representations they already use in their classroom illuminate or hide.

Helping Teachers Address Their Anticipated Challenges

To formulate professional development that supports lasting change in teacher practice, developers must understand how teachers interpret the difficulties of the changes (Guskey, 2002). The time concerns expressed by teachers suggest the importance of using an integrated approach to bringing CT into elementary school. Although teaching mathematics and science through CT may increase the amount of instructional time devoted to these subjects, it will likely add less time than addressing CT as a separate subject. Teachers' concerns about the high-level thinking associated with CT and its appropriateness for elementary students suggests leveraging work that does exist in these areas about what are appropriate activities (e.g., Appendix F of NGSS with grade-appropriate guidelines of practices such as computational thinking and scientific modeling) and conducting additional research on what robust and appropriate CT looks like in elementary school. Our partner teachers are currently trying out many kinds of CT-enhanced mathematics and science lessons in their classrooms. In later work, we plan to report on the kinds of activities that were most successful in various contexts.

LIMITATIONS AND FUTURE DIRECTION

This study has several limitations. First, the sample size and context limit the generalizability of the results. Nonetheless, the results can provide an evidence-based starting point for informing current work and subsequent research. Future research could adapt the themes from this study into a survey to get a broader perspective on how elementary teachers see CT as related to their math and science teaching. Second, structuring the interviews around particular ideas and terminology present on the handouts may have prevented participants from accessing knowledge and teaching practices that relate to CT but are not associated with the terminology or the particular highlighted aspects of CT. As evidenced by our pilot interviews, different wording in the handouts led teachers to make different connections. Additionally, we acknowledge that other descriptions of CT may place greater emphasis on different ideas, which could allow teachers to make more or different connections to their mathematics or science instruction. Third, reliance on self-report from teachers may limit the validity of the results. To address these limitations, future work could also observe teachers' implementing math and science lessons in their classrooms to inform how CT ideas and practices are enacted (even though teachers might not see them as such). This would avoid over-reliance on particular terminology in interview handouts.

As noted above, we used the results of this study to guide the creation of personalized professional development experiences to support elementary school teachers in the sampled district in bringing computational thinking instruction to their students via mathematics and science instruction. We believe the results can also inform curriculum development by helping designers link between mathematics, science, and computational thinking practices and can help administrators and policy directors think about constraints that may affect teachers' adoption of CT practices in their elementary classrooms. In addition, we have been using these findings to bring CT into preservice elementary teacher education instruction so that preservice teachers may leave their certification programs knowing more about computational thinking. Our work will continue to advance the efforts of bringing CT to elementary students and their teachers, with the goal of ultimately broadening and diversifying the CS field and giving all students the computer science education they need to thrive in an increasingly computing-oriented world.

APPENDIX A: INTERVIEW PROTOCOL

Thank you for taking the time to talk to me today. The goal is to get your perspective on computational thinking and how it might apply in your classroom. There are no right or wrong answers, we are just interested in your ideas to help us develop PD activities.

1. Can you tell me a little bit about yourself? Where and what grade level do you teach?

2. If a colleague asked you to describe what computational thinking was, what would you say?

3. **[Present CT handout.]** CT is generally thought of as the thinking practices used by computer scientists. Here is how some organizations and researchers have described some key CT ideas. I will give you a few minutes to read it through. Then I'm going to ask you if you've seen students in your classroom using these CT ideas, and when.
    a. Can you think of any examples of when you've seen your students use these ideas? Describe as many examples as you can.
    b. (If no mention of mathematics in examples) Have you seen your students engaging in this kind of thinking as they do <u>mathematics</u>? Describe as many examples as you can.
    c. (If no mention science in examples) Have you seen your students engaging in this kind of thinking as they do <u>science</u>? Describe as many examples as you can.

4. a. If you wanted to make the CT a particular focus in your mathematics instruction, how would you change the way you taught the activities you described?

b. If you wanted to make the CT a particular focus in your science instruction, how would you change the way you taught the activities you described?

5. What, if any, are the benefits of teaching mathematics and science using these CT ideas?

6. What, if any, are the challenges in teaching mathematics and science using these CT ideas?

   a. How do the challenges you're mentioning play out in your particular situation? For example, what would make CT difficult for your students? What makes implementation of CT difficult in your school?

7. **[Present Math Practices handout.]** Here are some of the practices that are used by mathematicians (according to the Common Core Standards). I'll give you a few minutes to read it through.

   a. Do you see any connections between these and the CT ideas? Give one or two examples.
   b. Do you see any important differences between CT and the math practices? Give one or two examples.

8. **[Present Science Practices handout.]** Here are some of the practices and cross-cutting concepts that are used by scientists (according to the Next Generation Science Standards). I'll give you a few minutes to read it through.

   a. Do you see any connections between these science practices and cross-cutting concepts and the CT ideas? Give one or two examples.
   b. Do you see any important differences between CT and these science ideas? Give one or two examples.

9. In what ways do you think that your students might benefit from learning CT concepts (beyond learning math and science)? How might this help them later in life?

   a. Computer scientists think that these skills are important for students to think computationally and prepare them to take either CS courses or even pursue a CS career in future. What are your thoughts as an elementary school teacher about how this could play out as your students move along K-12?

10. Is there anything else you would like to add to what you have already said?

APPENDIX B: CT HANDOUT

Computational Thinking Core Skills

Source: Bocconi, S., Chioccariello, A., Dettori, G., Ferrari, A., Engelhardt, K. (2016). Developing computational thinking in compulsory education – Implications for policy and practice; EUR 28295 EN

Abstraction. Abstraction is the process of making an artifact more understandable through reducing the unnecessary detail. The skill in abstraction is in choosing the right detail to hide so that the problem becomes easier, without losing anything that is important. A key part of it is in choosing a good representation of a system. Different representations make different things easy to do.

Algorithmic Thinking. Algorithmic thinking is a way of getting to a solution through a clear definition of the steps.

Automation. Automation is a labor saving process in which a computer is instructed to execute a set of repetitive tasks quickly and efficiently compared to the processing power of a human. In this light, computer programs are “automations of abstractions”.

Decomposition. Decomposition is a way of thinking about artifacts in terms of their component parts. The parts can then be understood, solved, developed and evaluated separately. This makes complex problems easier to solve, novel situations better understood and large systems easier to design.

Debugging. Debugging is the systematic application of analysis and evaluation using skills such as testing, tracing, and logical thinking to predict and verify outcomes.

Generalization. Generalization is associated with identifying patterns, similarities and connections, and exploiting those features. It is a way of quickly solving new problems based on previous solutions to problems, and building on prior experience. Asking questions such as “Is this similar to a problem I’ve already solved?” and “How is it different?” are important here, as is the process of recognizing patterns both in the data being used and the processes/strategies being used. Algorithms that solve some specific problems can be adapted to solve a whole class of similar problems.

APPENDIX C: MATH PRACTICES HANDOUT

Mathematical Practices

Source: Common Core State Standards for Mathematics

Make sense of problems and persevere in solving them. Students explain the meaning of a problem and plan a solution. They try special cases and simpler forms of the original problem to help them get started. Students check their answers to problems using a different method, and they continually ask themselves, “Does this make sense?”

Reason abstractly and quantitatively. Students make sense of quantities and their relationships in problem situations. They abstract a given situation and represent it symbolically—for example, they write numbers to represent collections of objects—then manipulate the symbols, pausing as needed to recall what the symbols represent.

Construct viable arguments and critique the reasoning of others. Students make claims about mathematics and build a logical progression of statements to “prove” those claims, using objects, drawings, diagrams, and actions to illustrate their thinking. Students also compare arguments and explain flaws in others’ reasoning.

Model with mathematics. Students apply mathematics to solve everyday problems. For example, they might write an addition equation to describe a situation. They identify important quantities in problems, and use diagrams tables, and graphs to understand relationships among those quantities and draw conclusions.

Use appropriate tools strategically. Students consider the available tools when solving a problem, which might include pencil and paper, a ruler, or a calculator. They make sound decisions about when each of these tools might be helpful.

Attend to precision. Students communicate precisely to others, using clear definitions. They state the meaning of symbols, specify units, and label diagrams. They calculate accurately and efficiently.

Look for and make use of structure. Students look closely to discern a pattern or structure. For example, they notice that 3 + 7 is the same as 7 + 3. They can see complicated things, such as composite shapes, as single objects or as composed of several objects.

Look for and express regularity in repeated reasoning. Students notice if calculations are repeated, and look for general methods and for shortcuts. For example, they might notice when dividing 25 by 11 that they are repeating the same calculations over and over again, and conclude they have a repeating decimal.

APPENDIX D: SCIENCE PRACTICES HANDOUT

Science and Engineering Practices

Source: Next Generation Science Standards

Asking Questions and Defining Problems. Students at any grade level should be able to ask questions of each other about the texts they read, the features of the phenomena they observe, and the conclusions they draw from their models or scientific investigations. For engineering, they should ask questions to define the problem to be solved and the constraints and specifications for its solution.

Developing and Using Models. Models include diagrams, physical replicas, mathematical representations, analogies, and computer simulations. Models bring certain features into focus while obscuring others. Modeling can begin in the earliest grades, with students' models progressing from concrete "pictures" and/or physical scale models to more abstract representations of relevant relationships.

Planning and Carrying Out Investigations. Students should design investigations that generate data to provide evidence to support claims they make about phenomena. It is always important for students to state the goal of an investigation, predict outcomes, and plan a course of action that will provide the best evidence to support their conclusions.

Analyzing and Interpreting Data. Once collected, data must be presented in a form that can reveal any patterns and relationships and that allows results to be communicated to others. Because raw data have little meaning, a major practice of scientists is to organize and interpret data through tabulating, graphing, or statistical analysis.

Using Mathematics and Computational Thinking. Students are expected to use mathematics to represent physical variables and their relationships and to make quantitative predictions. Students are also expected to engage in computational thinking, which involves strategies for organizing and searching data, creating sequences of steps called algorithms, and using and developing new simulations of systems.

Constructing Explanations and Designing Solutions. The goal of science is to construct explanations for phenomena. Students are expected to construct their own explanations, as well as apply standard explanations they learn from teachers or reading. The goal of engineering is to solve problems. Designing solutions is a systematic process that involves defining the problem, then generating, testing, and improving solutions.

Engaging in Argument from Evidence. Argumentation is a process for reaching agreements about explanations and solutions. In science, arguments based on evidence are essential in identifying the best explanation for a natural phenomenon. In engineering, reasoning and argument are needed to identify the best solution to a problem.

Obtaining, Evaluating, and Interpreting Information. Being a critical consumer of information requires the ability to read or view reports of scientific or technological advances (whether found in the press, on the Internet, or in a town meeting) and to recognize salient ideas, identify sources of errors and flaws, and distinguish observations from inferences, arguments from explanations, and claims from evidence.

Acknowledgement

This work is supported by the National Science Foundation under grant number 1738677. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

References

Astrachan, O., Cuny, J., Stephenson, C., & Wilson, C. (2011, March). The CS10K project: mobilizing the community to transform high school computing. In *Proceedings of the 42nd ACM technical symposium on Computer science education* (pp. 85-86). New York: ACM.

Ball, D. L., Thames, M. H., & Phelps, G. (2008). Content knowledge for teaching: What makes it special?. *Journal of Teacher Education*, *59*(5), 389-407.

Baroody, A. J., Eiland, M. D., Purpura, D. J., & Reid, E. E. (2013). Can computer-assisted discovery learning foster first graders' fluency with the most basic addition combinations? *American Educational Research Journal, 50*(3), 533-573.

Barr, V., & Stephenson, C. (2011). Bringing computational thinking to K-12. *ACM Inroads*, *2*(1), 48-54.

Battista, M. T., & Clements, D. H. (1986). The effects of Logo and CAI problem-solving environments on problem-solving abilities and mathematics achievement. *Computers in Human Behavior*, *2*(3), 183-193.

Bocconi, S., Chioccariello, A., Dettori, G., Ferrari, A., & Engelhardt, K. (2016). *Developing Computational Thinking in Compulsory Education: Implications for Policy and Practice*. EUR 28295 EN. Retrieved from https://doi.org/10.2791/792158

Bogdan, R. C., & Biklen, S. K. (2007). *Qualitative Research for Education: An Introduction to Theories and Methods.* Boston: Pearson.

Carpenter, T. P., Fennema, E., Franke, M. L., Levi, L., & Empson, S. B. (2000). *Cognitively Guided Instruction: A research-based teacher professional development program for elementary school mathematics*. Research Report from the National Center for Improving Student Learning and Achievement in Mathematics and Science. Retrieved from https://eric.ed.gov/?id=ED470472

Carpenter, T. P., Franke, M. L., Jacobs, V. R., Fennema, E., & Empson, S. B. (1998). A longitudinal study of invention and understanding in children's multidigit addition and subtraction. *Journal for Research in Mathematics Education, 29*(1), 3-20.

College Board. (2017a). *AP Computer Science Principles.* Retrieved from https://apcentral.collegeboard.org/pdf/ap-computer-science-principles-course-and-exam-description.pdf

College Board. (2017b). *AP Program Participation and Performance Data 2017: National Summary Report.* Retrieved from https://research.collegeboard.org/programs/ap/data/participation/ap-2017

Common Core State Standards Initiative (CCSSI). (2010). *Common Core State Standards for Mathematics.* Retrieved from CCSSI website: http://www.corestandards.org/Math/

Davis, E. A., & Krajcik, J. S. (2005). Designing educative curriculum materials to promote teacher learning. *Educational Researcher*, *34*(3), 3–14.

Denning, P. J. (2017). Remaining trouble spots with computational thinking. *Communications of the ACM*, *60*(6), 33–39.

diSessa, A. A. (1993). Toward an epistemology of physics. *Cognition and Instruction*, *10*(2–3), 105–225.

Dwyer, H., Boe, B., Hill, C., Franklin, D., & Harlow, D. (2013). Computational thinking for physics: Programming models of physics phenomenon in elementary school. In Engelhardt, Churukian, & Jones (Eds.), *2013 PERC Proceedings*, 133-136.

Fessakis, G., Gouli, E., & Mavroudi, E. (2013). Problem solving by 5–6 years old kindergarten children in a computer programming environment: A case study. *Computers & Education*, *63*, 87-97.

Flannery, L. P., & Bers, M. U. (2013). Let's dance the "robot hokey-pokey!" Children's programming approaches and achievement throughout early cognitive development. *Journal of Research on Technology in Education*, *46*(1), 81-101.

Forman, E., & Ansell, E. (2001). The multiple voices of a mathematics classroom community. *Educational Studies in Mathematics, 46*(1/3), 115-142.

Goode, J., Chapman, G., & Margolis, J. (2012). Beyond curriculum: The exploring computer science program. *ACM Inroads*, *3*(2), 47–53.

Google. (2014). *Women Who Choose Computer Science—What Really Matters: The Critical Role of Encouragement and Exposure.* Retrieved from https://static.googleusercontent.com/media/ edu.google.com/en//pdfs/women-who-choose-what-really.pdf

Gretter, S., & Yadav, A. (2016). Computational thinking and media & information literacy: An integrated approach to teaching twenty-first century skills. *TechTrends*, *60*(5), 510-516.

Griffith, P. L., Ruan, J., Stepp, J., & Kimmel, S. J. (2014). The design and implementation of effective professional development in elementary and early childhood settings. In L. E. Martin, S. Kragler, D. J. Quatroche, & K. L. Bauserman (Eds.), *The Handbook of Professional Development in Education: Successful Models and Practices, PreK-12* (pp. 189–204). New York: Guilford Press.

Grossman, P. L. (1989). A study in contrast: Sources of pedagogical content knowledge for secondary English. *Journal of Teacher Education*, *40*(5), 24–31.

Grover, S., & Pea, R. (2013). Computational thinking in K–12 : A review of the state of the field. *Educational Researcher*, *42*(1), 38–43.

Guskey, T. (2002). Professional development and teacher change. *Teachers and Teaching: Theory and Practice, 8*(3), 381-391.

Harlow, D. B., Bianchini, J. A., Swanson, L. H., & Dwyer, H. A. (2013). Potential teachers' appropriate and inappropriate application of pedagogical resources in a model-based physics course: A "Knowledge in Pieces" perspective on teacher learning. *Journal of Research in Science Teaching, 50*(9), 1098-1126.

Hestness, E., Ketelhut, D. J., McGinnis, J. R., & Plane, J. (2018). Professional knowledge building within an elementary teacher professional development experience on computational thinking in science education. *Journal of Technology and Teacher Education, 26*(3), 411-435.

Israel, M., Pearson, J. N., Tapia, T., Wherfel, Q. M., & Reese, G. (2015). Supporting all learners in school-wide computational thinking: A cross-case qualitative analysis. *Computers & Education*, *82*, 263-279.

K-12 CS Framework. (2016). *K-12 Computer Science Framework Statements.* Retrieved from https://k12cs.org/

Krist, C., Elby, A., Good, J., Gupta, A., Sohr, E. R., & Yadav, A. (2017, April). *Integrating computational thinking strategies that support science inquiry: A case study from a summer PD*. Paper presented at the Annual Meeting of American Educational Research Association, San Antonio, TX.

Lichtman, M. (2006). Learning about others through interviewing. In *Qualitative Research in Education: A User's Guide* (pp. 115–135). Thousand Oaks, CA: Sage Publications.

Loucks-Horsley, S., Stiles, K. E., Mundry, S., Love, N., & Hewson, P. W. (2010). *Designing Professional Development for Teachers of Science and Mathematics* (Third Edition). Thousand Oaks, CA: Corwin.

Margolis, J., Estrella, R., Goode, J., Holme, J. J., & Nao, K. (2008). *Stuck in the Shallow End: Education, Race, and Computing.* Cambridge, MA: MIT Press.

Minner, D. D., Levy, A. J., & Century, J. (2010). Inquiry-based science instruction-what is it and does it matter? Results from a research synthesis years 1984 to 2002. *Journal of Research in Science Teaching*, *47*(4), 474–496.

Nager, A, & Atkinson, R. D. (2016). The Case for Improving U.S. Computer Science Education. Retrieved from the Information Technology & Innovation Foundation website: https://itif.org/publications/2016/05/31/case-improving-us-computer-science-education

National Center for Education Statistics (NCES). (2015). Digest of Education Statistics, Table 325.35: Degrees in computer and information sciences conferred by postsecondary institutions, by level of degree and sex of student: 1970-71 through 2014-15. Retrieved from https://nces.ed.gov/programs/digest/d16/tables/dt16_325.35.asp?current=yes

National Research Council. (2001). *Adding It Up: Helping Children Learn Mathematics.* Washington, DC: National Academies Press.

Next Generation Science Standards. (2013). *Next Generation Science Standards.* Retrieved from https://www.nextgenscience.org/search-standards

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

Patton, M. Q. (1987). Depth interviewing. In *How to Use Qualitative Methods in Evaluation* (pp. 108–143). Thousand Oaks, CA: Sage Publications.

Pollock, L., Mouza, C., Czik, A., Little, A., Coffey, D., & Buttram, J. (2017). From professional development to the classroom: Findings from CS K-12 teachers. In *Proceedings of the 2017 ACM SIGCSE Technical Symposium on Computer Science Education* (pp. 477–482). New York: ACM.

Reding, T. E., & Dorn, B. (2017). Understanding the " teacher experience " in primary and secondary CS professional development. In *Proceedings of the 2017 ACM Conference on International Computing Education Research* (pp. 155–163). New York: ACM.

Rich, K. M., Spaepen, E., Strickland, C., & Moran, C. (2019). Synergies and differences in mathematical and computational thinking: Implications for integrated instruction. *Interactive Learning Environments.* http://doi.org/10.1080/10494820.2019.1612445

Sands, P., Yadav, A., & Good, J. (2018). Computational thinking in K-12: In-service teacher perceptions of computational thinking. In M. S Khine. (Ed.). *Computational Thinking in the STEM Disciplines*. Springer.

Seiter, L. (2015, February). Using SOLO to classify the programming responses of primary grade students. In *Proceedings of the 46th ACM Technical Symposium on Computer Science Education* (pp. 540-545). New York: ACM.

Seiter, L., & Foreman, B. (2013, August). Modeling the learning progressions of computational thinking of primary grade students. In *Proceedings of the ninth annual international ACM conference on International computing education research* (pp. 59-66). New York: ACM.

Simon, M. A. (1995). Reconstructing mathematics pedagogy from a constructivist perspective. *Journal for Research in Mathematics Education*, *26*(2), 114–145.

Shulman, L. S. (1986). Those who understand: Knowledge growth in teaching. *Educational Researcher*, *15*(2), 4–14. https://doi.org/10.3102/0013189X015002004

Toom, A. (2017). Teachers' professional and pedagogical competencies: A complex divide between teacher work, teacher knowledge, and teacher education. In D. J. Clandinin & J. Husu (Eds.). *The SAGE Handbook of Research on Teacher Education* (pp. 803-819). London: SAGE Publications Ltd.

Weintrop, D., Beheshti, E., Horn, M., Orton, K., Jona, K., Trouille, L., & Wilensky, U. (2016). Defining computational thinking for mathematics and science classrooms. *Journal of Science Education and Technology*, *25*(1), 127–147.

Weiss, R. S. (1994). Introduction. In *Learning from Strangers: The Art and Method of Qualitative Interview Studies* (pp. 1–14). New York: Free Press.

White House. (2016). "Computer Science for All." Blog post dated January 30, 2016. Retrieved from https://obamawhitehouse.archives.gov/blog/2016/01/30/computer-science-all

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

Yadav, A., Gretter, S., Good, J., & McLean, T. (2017). Computational thinking in teacher education. In *Emerging Research, Practice, and Policy on Computational Thinking* (pp. 205-220). Springer International Publishing.

Yadav, A., Hong, H., & Stephenson, C. (2016). Computational thinking for all: Pedagogical approaches to embedding 21st century problem solving in K-12 classrooms. *TechTrends*, *60*(6), 565-568.

Yadav A., Krist, C. Good. J., & Caeli, E. (2018). Computational thinking in elementary classrooms: Measuring teacher understanding of computational ideas for teaching science. *Computer Science Education*. DOI: 10.1080/08993408.2018.1560550

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

Yadav, A., Larimore, R., Rich, K. M., & Schwarz, C. (2019). Integrating computational thinking in elementary classrooms: Introducing a toolkit to support teachers. In K. Graziano (Ed.), *Proceedings of the Society for Information Technology & Teacher Education International Conference 2019* (pp. 347–350).

Yadav, A., Stephenson, C., & Hong, H. (2017). Computational thinking for teacher education. *Communications of the ACM, 60*(4), 55-62.

Yadav, A., Zhou, N., Mayfield, C., Hambrusch, S., & Korb, J. T. (2011.) Introducing computational thinking in education courses. In *Proceedings of the 42nd ACM technical symposium on Computer science education* (pp. 465-470). New York: ACM. https://doi.org/10.1145/1953163.1953297