

Teachers' Understanding of Synergies Between Computational and Mathematical Thinking after a Summer Professional Development

Zuhal YILMAZ^{1*}, Karmen Williams², Fernando Alegre³, Rose Kendrick⁴, Juana Moreno⁵

^{1,2,3,4,5} Louisiana State University, United States of America

zyilmaz@lsu.edu, kwil419@lsu.edu, falegre@lsu.edu, rkendr3@lsu.edu, moreno@lsu.edu

ABSTRACT

Emergent studies suggest that engaging in computer science education has the potential to support students' learning of key mathematical concepts due to the connection between mathematical and computational thinking (MT & CT). To create a rigorous learning environment that focuses on this connection, teachers must gain an in-depth understanding of the synergies between CT and MT, and learn the ways to integrate the common practices and aspects of both into their practice. Thus, we co-designed a five-week long summer professional development (PD) that cultivated various perspectives about the synergies between CT and MT. We explored the shifts in understanding of CT and its connection to MT with a cohort of ten teachers from multiple subject areas. The results suggest that the teachers displayed an overall improvement in the richness and depth of their descriptions of CT and MT, and provided a variety of examples of synergies between them. Most of the teachers recognized problem-solving, generalization and abstraction, and decomposition as synergies between CT and MT. They also explained more precisely how they used mathematical knowledge in computing activities after the PD. These results suggest that the PD may help teachers to integrate both types of thinking into their classroom practices. We also found that some aspects of MT and CT, such as modeling, did not surface in the data analysis. This finding will be helpful to chart the focus and design of future PDs.

KEYWORDS

computational thinking, mathematical thinking, in-service teacher, computer science, professional development

1. INTRODUCTION

There has been an increased interest in making computer science (CS) education a core subject in K-12 education (Menekse, 2015). Various studies emphasized the potential benefits of integration of CS education on students' mathematical learning (Alegre et al., 2022; Barcelos et al., 2018). However, teachers must first develop an understanding about this integration to create an effective learning environment for their students. Thus, we co-designed a professional development (PD) focusing on understanding what computational and mathematical thinking (CT and MT) include and highlighting the synergies between these two types of thinking.

CT is "the thought processes involved in formulating problems and their solutions so that the solutions are represented in a form that can be efficiently carried out by an information-processing agent." (Wing, 2006). Weintrop et al., (2016) also stated that CT is reformulating problems,

thinking recursively, using decomposition and abstraction, choosing appropriate models, and using heuristic reasoning.

From a mathematics education perspective, the discourse on CT resonates with the fundamental theoretical frameworks of mathematics education like mathematical modeling (Lesh & Fennewald, 2013), problem solving (Schoenfeld & Sloane, 2016), generalization and abstraction (Tall, 1999), and quantitative reasoning (Duval, 1999). Thereby, these similarities inform and guide us to design a PD for teachers that focuses on the synergies between CT and MT.

2. CONNECTING MATHEMATICS AND COMPUTING

Using computing to aid students' mathematics and science learning has a long history. Early attempts to use programming "as a tool were based on unguided discovery" (Alegre et al., 2020, p. 992). This approach is shown to be ineffective for transfer (Mayer, 2004). However, in the past decade, this trend was reversed when the Bootstrap project (Schanzer et al. 2013) started to show evidence of transfer. Most recent studies focus on programming skills with a limited explicit connection to key mathematical concepts (Hickmott et al., 2018). Hickmott et al. (2018) reviewed 393 studies published up to the ten years after Wing (2006)'s article and found that one of the major gaps in the literature was the limited empirical studies that explicitly connected CT and MT. They suggested a lack of mathematics education expertise as the leading factor to numerous studies that "incidentally" make the connection.

Only a few studies (e.g. Barcelos et al., 2018; Brating & Kilhamn, 2021) explore learning of key mathematical ideas (e.g. algebraic thinking, functions, multiple representations) through engaging in computing. These studies suggested that engaging in computing tasks could enhance students' learning of key mathematical ideas.

2.1. PD in CS: Connecting CT and MT

CT is still a relatively new concept, many teachers lack the knowledge and resources necessary to successfully incorporate it into their curricula (Yadav et al., 2016; Wu et al., 2021). Providing effective PDs to support teachers' knowledge of CT and equip them with necessary resources are critical to improve students' learning experiences. Thus, we have reviewed the existing CS PD literature, particularly those focusing on supporting teachers' understanding of the synergies between CT and MT.

In our review, we only encountered a few PD studies (e.g. Hart et al., 2008; Wu et al., 2021) that focused on this connection. Hart et al. (2008) conducted a series of summer workshops to "provide teachers with innovative activities

and ideas that link their secondary school mathematics curriculum with computer science.” (p. 286). Wu et al. (2021) worked with 11 science and mathematics teachers in a co-designed PD focusing on enhancing teachers’ confidence in integrating STEM-CS practices (e.g. modeling and simulation, and data practices) in their curricula. They found that teachers gained confidence and skills in designing STEM-CS curriculum.

Menekşe (2015) conducted a review on CS education PD in the US. She found that there is limited collaboration between researchers and practitioners in designing PD. Also, she found that the majority of computer science PD was shorter than a week and the support was not ongoing. She also found a few PDs focused on integration of CS in mathematics curriculum. To this end, we co-designed a 5- weeks long summer PD with mathematics and CS education researchers and practitioners. This PD focused on supporting teachers’ understanding of CT and MT, and the synergies between both. We seek to answer the following research questions:

- 1- Which aspects of computational thinking were emphasized in teachers’ description of CT before and after the co-designed PD?
- 2- Which aspects of mathematical thinking were emphasized in teachers’ description of MT before and after the co-designed PD?
- 3- How did teachers’ understanding of the synergies between CT and MT change after the co-designed PD intervention?

3. METHOD

Researchers interviewed ten teachers about their thinking and understanding of CT, MT and their synergies before and after the PD.

3.1. Research Context and Participants

Alina and Jessica were facilitators of the five-week long summer PD. Both facilitators taught the Introduction to Computational Thinking (ICT) course (Alegre et al., 2020) to 9th graders. Alina is a visual art teacher with a teaching experience of 12 years, and Jessica is a former mathematics teacher with 5 years of teaching experience. Before the PD, both facilitators were trained by an accomplished mathematics coach who has 21 years of experience and 4 years in CS education. Further, the facilitators and accomplished coach debriefed after each PD day and planned for the next day. Ten teachers from high-need schools engaged in PD for five hours per weekday. T4, T7, and T10 have a background in CS, T5 in robotics and T6 in mathematics. Teaching experience ranges from 0 to 17 years (average of 4.5 years). Three graduate students supported the facilitators by helping teachers with content knowledge and technical issues while they worked in breakout rooms.

The PD content focused on the following areas: problem solving, coding as an expression of ideas, decomposition, automation, generalization and abstraction, importance of order, and reification. The activities required use of mathematical concepts such as the coordinate system, functions, and algebraic expressions. For example, write a program based on this prompt: “Create a triangle that has a right angle at the left”; solve a word problem following these

instructions: “do not calculate the solution in your head. Instead, just write an unevaluated expression in your program”; or replace repetitive parts of the code with a function in a loop. We also implemented diverse pedagogical strategies such as peer programming, code reviews, and working in small and whole groups.

3.2. Data Sources and Analysis

The primary data source of this study is the pre and post interviews of the participants. Each interview lasted 30-45 minutes. The lead researcher created the interview protocol based on essential aspects of CT (e.g. Weintrop et al., 2016; Wing, 2006) and MT (e.g. Schoenfeld & Sloane, 2016; Sternberg, 2012). Two researchers and an experienced ICT course teacher shared their feedback on the protocol. We piloted the protocol with a high school teacher and analyzed the pilot data to select questions which provided in-depth responses, and to improve the clarity of the questions. A few sample interview questions were: 1) How would you describe the connection between MT and CT? Can you give an example? 2) In what ways does summer PD support your understanding of the connection between MT and CT?

PD field notes were used as supportive data sources in the analysis. Thematic analysis of the interviews was used to characterize the different ways the teachers describe and exemplify the synergies between CT and MT. Content analysis of the field notes were used to identify the instances in which teachers connected mathematics and computation in PD activities. Practices (aspects) of CT and MT (e.g. abstraction, generalization, decomposition, problem solving) documented in the literature (e.g. Barcelos et al., 2018; Polya, 1945; Tall, 1999; Weintrop et al., 2016; Wing, 2006) guided the creation of codes. Two researchers independently coded the transcripts and the agreement rate was 84.6%. The researchers discussed the disagreements in the coding until reaching an agreement.

4. FINDINGS

The findings are reported in three separate subsections that focus on CT, MT, and the synergies between CT and MT, respectively. In each subsection, the changes in the teachers’ understanding are documented in two forms: as categorizations of the aspects emphasized in the teachers’ descriptions and as quoted examples.

4.1. Changes in understanding of CT

Analysis of the teachers’ descriptions reveals changes in multiple aspects of CT as shown in Table 1.

Table 1. CT aspects

CT aspect	CT aspect-subcategories	Teacher										T
		1	2	3	4	5	6	7	8	9	10	
Problem Solving	Planning	Pre				X		X			X	3
		Post	X			X	X		X	X		6
	Precision	Pre									X	1
		Post				X				X		2
	Decomposition	Pre				X					X	2
		Post	X			X	X	X	X	X	X	8
Critical Thinking	Pre			X	X			X		X	4	

Importance of Order	Post	X	X	X	X	X	X	X	7
	Pre								0
Using Algorithms	Post			X		X	X	X	5
	Pre					X		X	2
Functions (e.g. Input-output)	Post					X		X	2
	Pre			X		X	X		3
Language Aspects of Coding	Post	X	X		X			X	5
	Pre		X			X		X	3
Automation (Efficiency)	Post		X	X	X			X	5
	Pre								0
	Post				X	X			3

Analysis of the pre-interviews showed that only two teachers (T7, T10) could provide a meaningful description of CT. Both of them have a background in CS. Half of the teachers (T2, T3, T4, T5, T8) provided a vague description and very limited examples of CT, and three (T1, T6, T9) could not describe what CT means. After the PD, these three teachers could describe CT. For instance, T9 described it as: “[CT] means taking a problem and working through that problem step by step to figure out how to get the desired output.”

In the post, all the teachers provided a richer description of CT highlighting aspects such as problem solving, decomposition, coding, and order. Six teachers highlighted at least 4 important aspects of CT. However, details of their CT descriptions and examples still varied significantly.

In the pre-interview, 4 teachers indicated that CT includes problem solving. However, only 3 gave a limited explanation of why and how it includes problem-solving. For instance, T5 and T7 stated that in CT, as in a problem-solving process, they plan how to find the solutions. In the post, 8 teachers stated at least one problem-solving skill as they use CT. Only 4 out of them could explain the skill in-depth. For instance, T6 could not state any skills in the pre-interview. In the post, he stated: “[A] skill of a problem solver involves breaking it down into small components and where you can plug them into a computer to help you automate the system to make solving that problem faster.”

The analysis also showed there is an increase in the number of teachers stating that CT includes coding (from 3 to 5) and critical thinking (from 4 to 7). The five teachers who mentioned coding in the post-interview also mentioned critical thinking. They seemed to perceive CT as a thinking type that requires skills beyond coding.

Even though not all the teachers mentioned abstraction and generalization in their pre or post interviews, they used the concepts during the summer PD. For example, in the 4th week of the PD, the teachers were asked to write a program to draw a square of any size using variables. T1 first drew a 6x6 square with the code shown in Figure 1.

```

4 import Standard
5
6 square = [(3,3), (3,-3), (-3,-3), (-3,3)]
7
8 program = drawingOf(polygon(square))

```

Figure 1. Code to create 6x6 square

Then 2 teachers discussed how they could create a square of any size:

Jessica: Do you notice anything about those points on that list?

T4: They are all 3.

Jessica: Is there any way to create variables so you don't need to write 3 so many times?

T4: Set a variable and call it pointA = 3 and set another variable pointB = -point A

Then, the teachers started to change the code (Figure 2.)

```

4 import Standard
5
6 pointA = 3
7 pointB = -pointA
8
9 square = [(pointA,pointA), (pointA,pointB), (-3,-3), (-3,3)]
10
11 program = drawingOf(polygon(square))

```

Figure 2. Assigning variables

T1 realized that “seems longer than typing 3”. Other teachers agreed. Then they found a solution calling the variable “pointA” as “a” and “pointB” as “b” (Figure 3).

```

4 import Standard
5
6 a = 7
7 b = -a
8
9 square = [(a,a), (a,b), (b,b), (b,a)]
10
11 program = drawingOf(polygon(square))

```

Figure 3. Generalized code to create any size square.

T4 stated the benefit of doing this is that “you don't need to write all the points. This is a generalized solution to draw any square”.

4.2. Changes in understanding of MT

Analysis of the teachers' descriptions and examples reveals the following aspects of MT shown in Table 2.

Table 2. MT aspects

MT aspect		Teacher									
		1	2	3	4	5	6	7	8	9	10 T
Operations and calculations	Pre			X	X	X	X		X	X	6
	Post	X		X		X			X	X	5
Applying math to real life situations	Pre	X	X	X	X	X	X		X	X	7
	Post	X			X	X	X	X	X	X	7
Problem Solving	Pre			X		X	X				3
	Post		X		X	X	X	X	X	X	7
Process of producing an answer	Pre						X		X	X	3
	Post				X	X	X	X	X	X	6

Table 2 shows no significant change in the number of teachers for the first two categories. However, a conceptual progression in some of the teachers was observed. For instance, three (T6, T9, T10) of the six teachers who perceived MT as carrying out calculations, performing operations in the pre-interview mentioned this aspect in the context of problem-solving situations in the post-interview.

Conceptual progression was also observed in the second category. Although seven teachers stated MT requires

applying mathematics to real life situations in the pre-interview, their examples ($n = 5$) for this aspect lack details. After the PD, the teachers gave more detailed examples of use of MT in real life situations. For instance, in the pre-interview, T1 said that we use MT in grocery shopping, and in the post, the same teacher stated::

MT might be used in grocery shopping, where you need to figure out what the cheapest price for something is. Just because one of them has a lower price on the tag doesn't mean it's the cheapest one, you're going to have to figure out how much per ounce it is, to see if it's actually cheaper.

While T6 and T8 could not give an example in the pre, they gave detailed examples of MT in real life in the post. T8:

You put coffee. You have to know how much coffee grinds that you're going to have to put in that coffee. And if you don't put enough you end up being really watery and not taste good. In a mathematical sense, there's a portion and that portion would be equivalent to some type of number.

Numbers changed significantly in the last two categories. The number of teachers indicating that MT includes problem solving increased from three to seven teachers from pre to post interview. In the post, teachers described the problem-solving process in more detail.

Only T10 stated MT encompasses proof and generalization when we asked what MT means in both interviews. Although other teachers did not state generalization in response to this question in the interviews, 6 teachers, including T10, stated generalization and abstraction is one of the synergies between CT and MT (see section below).

4.3. Synergies between CT and MT

Analysis of the teachers' responses revealed three main synergies between CT and MT as follows: 1) Mathematical concepts used in computation, 2) Engaging in problem solving 3) Practices used in both types of thinking.

4.3.1. Mathematics Concepts Used in Computation

Table 3 shows the distribution of the concepts that teachers stated in both interviews.

Table 3. Mathematics Concepts used in Computation

Categories		Teacher										T
		1	2	3	4	5	6	7	8	9	10	
Functions	Pre											0
	Post	X		X	X	X	X		X	X		7
Operations Calculations	Pre			X								1
	Post	X		X	X				X	X		5
Coordinates	Pre											0
	Post	X	X		X	X			X	X		6
Geometric Shapes	Pre											0
	Post	X		X	X							3

As seen in Table 3, before the PD, teachers saw no use of mathematical concepts in computation. Remarkably, in the post, most of them stated that they used various mathematical concepts such as operations, functions, and coordinates as they engaged in computation.

A unique feature of the PD was the use of computing keywords that prioritize connections to math over computer architecture. In this vein, the programming language used in the PD implements loops using a function called "distributed". T10 with CS background highlighted this connection as:

Applying an operation to a list of objects in your code, you have a function. That's called distributed, it takes what would normally be a for next loop, and puts it into and frames it in a way that it immediately invokes the distributive law of multiplication. So that is useful. And it reinforces ideas about how functions are composed in a mathematical expression, as well as being useful for coding too.

Similarly, T4 explained how he used math in automating repetition when asked to make a sun with 16 equidistance rays:

When I rotate my rays around my sun, I know how many angles are in a circle, 360, how many rays do I need to get, 16. Then 360 divided by 16 tells me what the angle difference between each ray is. And then, [I used] a distributed function [to create each ray], which is very similar to putting x outside of a parenthesis of two plus three, knowing that that x has to be distributed to $2x$ plus $3x$.

4.3.2. Engaging in Problem Solving in CT and MT

Engaging in the problem solving steps (Polya, 1945) of "understand the problem, devise a plan, carry out the plan (solve) and look back (check and interpret)" was the most frequently stated synergy between CT and MT after the summer PD. Table 4 shows large changes in all the categories.

Table 4. Synergy of Engaging in Problem Solving

Categories		Teachers										T
		1	2	3	4	5	6	7	8	9	10	
Understand the Problem	Pre										X	1
	Post	X				X	X		X	X	X	6
Devise a Plan	Pre										X	1
	Post	X				X	X		X	X	X	6
Carry out the Plan	Pre										X	1
	Post	X	X			X	X		X	X		6
Check back	Pre										X	1
	Post	X	X		X					X	X	5
Persevere in Problem Solving	Pre											0
	Post		X		X			X				3
Generate Solutions in Multiple Ways	Pre											0
	Post					X	X					2

While six out of 10 teachers provided rich explanations for this synergy in their post-interviews, only one teacher with a CS background mentioned it in the pre-interview (See Table 4). However, in the post, these six teachers also provided examples for the problem-solving steps. For instance, T1 explained how they engaged in the first three steps of the problem-solving process together with decomposition strategies:

You need to use CT, sometimes to get a clearer understanding of a math problem. You need to figure out what the goal is [Understand the problem] and how you're going to get there [Devise a Plan], and then do calculations [Carry out the Plan]. It emphasized the importance of breaking things down step by step. That's what you have to do to figure out and to make mathematical decisions.

Similarly, T10's stated in the post-interview:

You have some large problems, and you have to solve various pieces of it first [Carry out the Plan], and then come back to the larger problem with those results [Check back].

As seen in Table 4, after the PD, three teachers (T2, T4, T7) indicated a critical practice used in both types of thinking: persevere in solving problems (NCTM, 2022). T2 stated:

When we did our code reviews with each other, because someone was struggling with the final image. What we would do is instead of giving them the answers, we would question them, ...so that they can solve the problem on their own. ... Like in solving a mathematics problem.

This quote of T2 highlighted the importance of scaffolding to support productive struggle and encourage perseverance while solving a problem that requires use of CT. T2 also stated that this process is similar in math problem solving.

Only 2 teachers stated that producing solutions in multiple ways is another synergy between CT and MT in the post-interview. For instance, T5 explained this synergy as:

Projects made you figure out a unique way with the limited knowledge that you have, because we have learned solid circles or how to draw circles or how to make any kind of oval-like shape. We will try to draw animals using only polygons and lines. It makes you think of unique ways to solve that problem with the limited information you have.

During the PD, all the teachers created, for instance, unique animal designs, sunny scenes, and pictures using polygons, lines and points. They acknowledged that there is more than one way to create the outcome just like in mathematics.

4.3.3. Practices used in both CT and MT

Analysis of the teachers' responses showed that the following practices are used both in CT and MT (Table 5).

Table 5. Common practices of CT and MT

Categories		Teacher										T
		1	2	3	4	5	6	7	8	9	10	
Generalization and abstraction	Pre										X	1
	Post	X	X		X			X		X	X	6
Automation (Efficiency)	Pre										X	1
	Post				X			X				2
Debugging	Pre										X	1
	Post		X		X			X			X	4
Decomposition	Pre											0
	Post	X	X	X	X	X			X		X	7

Importance of Order	Pre											0
	Post							X		X	X	2

In the pre-interview, only T10 indicated generalization and abstraction as one of the practices of CT and MT. In contrast, in the post-interview the majority of the teachers (n = 6) acknowledged generalization and abstraction as one of the common practices of CT and MT. For instance, T1 stated:

That's kind of a generalization. When you see something that is repeated in that code, you need to generalize it and kind of simplify it. So that goes with math too. You have to ...make things easier to understand for the outside viewer.

This quote showed how the “look for and express regularity in repeated reasoning” (NCTM, 2022) mathematical practice can also be used in CT and how it is connected to generalization.

In addition, all 6 teachers provided concrete examples when they used generalization and abstraction in the tasks that used CT and MT (See figure 3 as an example). During the PD, 9 out of 10 teachers explicitly noticed the regularities in the code and defined functions for the regularities.

Although the majority of the teachers thought generalization and abstraction as a common practice, two teachers stated efficiency and automation as a common practice in the post-interview. These teachers did not explain why they thought it is a common practice of CT and MT. T7 stated:

Trying to come up with a quick way of solving. So, problem solving when you're testing things,...to get the things out.

Another common practice of CT and MT stated in the post-interview was decomposition (n = 7) and debugging (n = 4). For instance, T2 said:

Learned how to think more in a mathematical sense, like using math to solve coding issues. I never would have thought that you could use math to figure out why your code is wrong. The second thing would be breaking apart code, like into pieces. In order to solve the problem, like taking it step by step until you figure out what exactly is wrong.

Since debugging is a skill based on concepts such as separation of concerns and decomposition, it is difficult to determine exactly which are the underlying concepts T2's comments are alluding to. Other aspects of troubleshooting, such as logical reasoning, were not mentioned.

The last common practice of CT and MT was the importance of order, and it was mentioned by two teachers in the post. However, during the summer PD, all the teachers observed the results of different orders, such as how the order in which the vertices of a geometric shape are joined affects the outcomes, or how order in code matters for creating layered objects. T7 explained this ordering practice in the post:

Because I think of PEMDAS, you have to use your order of operations. Same way with CT. I'm coding or creating an algorithm, I may need to put it in the right order, or it's not going to be right.

Four teachers (T1, T2, T4, T7) mentioned modeling as one of the synergies in the post-interview. Since they did not explain this synergy or give an example of modeling in

which they used CT and MT, we did not classify this as one of the categories in Table 5.

5. RESULTS AND DISCUSSION

In a five-week PD focused on CT along with connections with MT, a significant shift in awareness of the centrality of problem solving in both types of thinking is observed from teachers of all backgrounds. Problem solving was mainly associated with the practices of decomposition and generalization/abstraction. However, decomposition seemed to be an unfamiliar concept to most teachers before the PD. The fact that the PD made this concept familiar to them is probably the reason that it was more explicitly stated than generalization during the post interviews.

The teachers demonstrated a progression in their understanding of CT and MT at varying levels. The dominant aspects emphasized in the post interview responses reflect the concepts stressed throughout the PD. The emphasis on decomposition, importance of order and generalization as aspects of CT, and calculations and applying mathematics to real-life situations as aspects of MT in teachers' responses were connected to how it is highlighted in the curriculum and by the PD facilitators. These results suggest a possible classification of CT concepts into a basic group (problem solving, decomposition and abstraction) and a more advanced group (precision, logical reasoning, automation and algorithms). A five-week PD seems to be suitable for learning the concepts in the former, but more time may be needed to internalize the concepts in the latter. Future work will investigate how the teachers' awareness of the concepts in the second group changes after a year of using them in the classroom.

The teachers' understanding of the synergies between CT and MT improved after the PD, in particular perceiving that both types of thinking types used in problem-solving. However, responses indicate some gaps in their understanding. Only a few teachers mentioned modeling as one of the synergies and yet, these teachers still had difficulty articulating the connections in detail. These results will support researchers in charting the focus and design of future PDs by considering which aspects of CT and MT will be explicitly explored with teachers during the PD. Also, there is a need for conducting a follow up future study on how the teachers make these connections in their classroom practices after the PD.

6. ACKNOWLEDGMENTS

This research project is partly funded by NSF award CNS-1923573 and the U.S. Dep. Of Education award U411CI90287.

7. REFERENCES

- Alegre, F., Underwood, J., Moreno, J., & Alegre, M. (2020, February). Introduction to CT: a new high school curriculum using codeworld. In *Proceedings of the 51st ACM Technical Symposium on Computer Science Education* (pp. 992-998).
- Alegre, F., Yilmaz, Z., Moreno, J & Kendrick, R. (2022, November). Culturally relevant computing tasks: Evidence of synergies between students' mathematical and computational thinking. *44th Annual Conference of the North American Chapter of the International Group for the Psychology of Mathematics Education*, Nashville, TN.
- Barcelos, T. S., Muñoz-Soto, R., Villarroel, R., Merino, E., & Silveira, I. F. (2018). Mathematics Learning through Computational Thinking Activities: A Systematic Literature Review. *J. Univers. Comput. Sci.*, 24(7), 815-845.
- Bråting, K., & Kilhamn, C. (2021). Exploring the intersection of algebraic and computational thinking. *Mathematical Thinking and Learning*, 23(2), 170-185.
- Duval, R. (1999). Representation, vision and visualization: cognitive functions in mathematical thinking. *Basic Issues for Learning*.
- Hart, M., Early, J. P., & Brylow, D. (2008, March). A novel approach to K-12 CS education: Linking mathematics and computer science. In *Proceedings of the 39th SIGCSE technical symposium on Computer Science Education* (pp. 286-290).
- Hickmott, D., Prieto-Rodriguez, E., & Holmes, K. (2018). A scoping review of studies on computational thinking in K-12 mathematics classrooms. *Digital Experiences in Mathematics Education*, 4(1), 48-69.
- Lesh, R., & Fennewald, T. (2013). Introduction to part I modeling: What is it? Why do it? In *Modeling students' mathematical modeling competencies* (pp. 5-10). Springer, Dordrecht.
- Mayer, R. E. (2004). Should there be a three-strikes rule against pure discovery learning? *American psychologist*, 59(1), 14.
- Menekse, M. (2015). Computer science teacher professional development in the United States: A review of studies published between 2004 and 2014. *Computer Science Education*, 25(4), 325-350.
- Polya, G. (1945). *How to Solve It*. Princeton University Press.
- Schanzer, E., Fisler, K., & Krishnamurthi, S. (2013). Bootstrap: Going beyond programming in after-school computer science. In *SPLASH education symposium*.
- Schoenfeld, A. H., & Sloane, A. H. (2016). *Mathematical thinking and problem solving*. Routledge.
- Tall, D. (Ed.). (1991). *Advanced mathematical thinking* (Vol. 11). Springer Science & Business Media.
- Weintrop, D., Beheshti, E., Horn, M., Orton, K., Jona, K., Trouille, L., & Wilensky, U. (2016). Defining CT for mathematics and science classrooms. *Journal of Science Education and Technology*, 25(1), 127-147.
- Wing, J. M. (2006). Computational thinking. *Communications of the ACM*, 49(3), 33-35.
- Wu, S., Peel, A., Bain, C., Horn, M.S., Wilensky, U. (2021). Different paths, same direction: Do teachers learn CT in stem practices through professional development. *Proceedings of the 5th APSCE International Computational Thinking and STEM in Education Conference*, (pp. 52-57).
- Yadav, A., Gretter, S., Hambruch, S., & Sands, P. (2016). Expanding computer science education in schools: understanding teacher experiences and challenges. *Computer Science Education*, 26(4), 235-254.