Journal for STEM Education Research (2020) 3:147–166
https://doi.org/10.1007/s41979-020-00044-w

EDITORIAL

On Computational Thinking and STEM Education

Yeping Li[1] · Alan H. Schoenfeld[2] · Andrea A. diSessa[2] · Arthur C. Graesser[3] · Lisa C. Benson[4] · Lyn D. English[5] · Richard A. Duschl[6]

Published online: 19 August 2020

Abstract

The recognized importance of computational thinking has helped to propel the rapid development of related educational efforts and programs over the past decade. Given the multi-faceted nature of computational thinking, which goes beyond programming and computer science, however, approaches and practices for developing students' computational thinking are not always self-explanatory in terms of their foci and feasibility in diverse educational contexts. In this editorial, we first examine relevant publications in computational thinking to identify a trend of integrating computational thinking into disciplinary education. We subsequently build on recent discussions about the concept of computational thinking to (1) frame a review of educational efforts in developing students' computational thinking, (2) discuss opportunities and challenges to further such educational efforts through not only programming and computer science but also other disciplines, and (3) articulate needed research and scholarship to support educational practices.

Keywords Assessment · Computational thinking · Programming · STEM disciplines · STEM education · Teacher education · Trends

Introduction

In our recent editorial (Li et al. 2020a), we discussed computational thinking (CT) as a model of thinking that is important to every student. Contrary to a common perception that CT belongs to programming and computer science (CS), we discussed its multi-faceted nature and highlighted the importance of knowing and understanding the concept as truly transdisciplinary, surpassing programming and CS instruction. Based on our proposal that CT should emphasize thinking as part of the problem-solving process and de-emphasize computing as producing "code," we further discussed how

✉ Yeping Li
yepingli@tamu.edu

Extended author information available on the last page of the article

this notion relates to different facets of CT as described with various approaches taken in the literature.

The meaning of CT has been open to different interpretations (e.g., Barr et al. 2011; Li et al. 2020a; NRC 2010), so it is not surprising that multiple models of CT exist, emphasizing different facets (Shute et al. 2017). There currently is diversity in efforts to design and promote educational approaches and practices that aim to develop students' CT (Grover and Pea 2013; NRC 2011). However, existing approaches and practices for developing students' CT are not always self-explanatory in guiding researchers and practitioners in the selection of relevant CT skills in particular educational contexts. Our purpose in this editorial is to build on recent discussions of CT to provide an overview of educational efforts in developing students' CT, specifically within the context of science, technology, engineering and mathematics (STEM) education.

In the following sections, we first provide a brief overview of trends of research in CT education in light of research publications, followed by a discussion of the complexity and challenges in terms of shifts in disconnecting and re-connecting CT and disciplinary education in STEM. We then focus our review and discussion on efforts in developing students' CT in terms of specific facets of CT, most prominently: CT as a discipline-specific thinking practice and CT as a trans-disciplinary thinking practice. Efforts in CT assessment and teacher education are also discussed as they are inseparable parts of CT education. We conclude by discussing future areas of research and scholarship to support educational practices in developing CT integrated in and through STEM education.

A Brief Overview: Trends of CT Research and Instruction

CT is a relatively new notion to educators, especially in pre-college school settings. Consequently, researchers and educators alike have devoted much time and efforts to debate and define its nature over the past decade (see for example, Grover and Pea 2013; Li et al. 2020a; NRC 2010; Shute et al. 2017). At the same time, many scholars have also noted the need for further research on CT and related education in both undergraduate and pre-college school levels, especially the need for empirical studies (Fennell et al. 2020; Grover and Pea 2013; Kalelioglu et al. 2016). Simple Google searches with the terms "computational thinking research," "computational thinking education," or "computational thinking in education" all returned more than 43,000,000 items. Such voluminous information shows a rapidly evolving and vibrant topic area, on the growth in interest in CT over the past decade.

Several articles published over the past several years have summarized and highlighted related development in CT research (e.g., Fennell et al. 2020; Grover and Pea 2013; Ilic et al. 2018; Kalelioglu et al. 2016; Shute et al. 2017; Tang et al 2020a). For example, Shute et al. (2017) searched several databases (e.g., ERIC and PsycINFO) and selected, reviewed, and discussed 45 articles published since 2006 to come up with a working definition of CT, which emerged from their review primarily as a way of thinking and acting. Ilic et al. (2018) also conducted a review of studies on CT indexed in the Web of Science (WoS) and ERIC databases. Their search revealed 96 studies on CT published between 2006 and 2016. They found that there was an increase in the number of CT studies in recent years, with 80 out of 96 studies (83.3%) published from

2013 to 2016, and that these studies were mainly conducted in CS. In fact, the majority of these articles were published in journals with a focus on computer, computing, and/or technology, with the largest set of publications (11) in *ACM Transactions on Computing Education*. Likewise, Tang et al. (2020a) searched the WoS database and identified 715 CT-related journal publications from 2006 to 2018. Although Tang et al. (2020a) did not examine the nature of journals where these articles were published, they reported an increasing trend of CT-related article publications over the years, especially when examining them in two time periods: 2006–2012 (246 articles) to 2013–2018 (469 articles).

Research on CT does not necessarily mean, or even include, research on CT in STEM education. In a recent review of STEM education research, Li et al. (2020b) identified and selected 798 articles published in 36 journals from 2000 to 2018 with authors' inclusion of self-identified term of STEM, or science, technology, engineering, and mathematics. Out of these 798 articles, however, only six articles had an explicit focus or connection with CT. The results suggest that CT research and instruction were dramatically lacking as evidenced in the number of journal publications related to STEM education. Putting that together with what was reported about CT research publications above (e.g., Ilic et al. 2018), we can determine that, at least as of a few years ago, CT research and instruction were mainly conducted through programming and within CS education, but without a close and clear connection with other disciplinary education.

In a simple Google search with the term "computational thinking book," we found that many books on CT were published over the past decade, either with a focus on research or instructional practice. There is also an increasing trend in book publication on CT over the past several years. Specifically, one book, *Computational Thinking in the STEM Disciplines* (Khine 2018), intended to focus on CT and STEM disciplines. However, only two out of 15 chapters in this book present a clear and direct connection with STEM disciplines. Most of other chapters in the book are on CT in school curricula, teacher education, programming and coding, or practices in various international contexts. It illustrates again the general lack of focus on CT research and instruction in disciplinary education just a few years ago.

In addition to what we can learn from research reviews and book publications, there are also some collective efforts in publishing about CT as special issues of journals. Table 1 shows a list of sample special issues of journals that we identified through a Google search with the term "computational thinking special issue." Three of these five special issues on CT research and instruction were published in journals with a clear focus on computer or technology, a result that echoes what was reported by Ilic et al. (2018). The other two journals are either content subject specific (i.e., *Mathematical Thinking and Learning*) or educational research in general albeit focusing on a region (i.e., *The Asia-Pacific Education Researcher*).

The topics of these special issues range from CT in education in general to CT with disciplinary education. For example, the special issue of *Mathematical Thinking and Learning* is a collection of four articles published in March 2018. As discussed and summarized by the journal's editor, these articles "present stimulating examples of how the implicit links between computational thinking and mathematics learning can be explicated and built upon." (English 2018, p. 2). The special issue of *Journal of Science Education and Technology* is another collection of 12 articles that focus on the

Table 1 Selected special issues of journals on CT published recently

Journal	Year	Special issue's topic
Mathematical Thinking and Learning	Jan 2018	Computational thinking and mathematics learning
Computers in Human Behavior	Mar 2018	Exploring the computational thinking effects in pre-university education
International Journal of Child-Computer Interaction	2018	Computational thinking and coding in childhood
Journal of Science Education and Technology	Feb 2020	Computational thinking from a disciplinary perspective
The Asia-Pacific Education Researcher	Feb 2020	Computational thinking education in the Asian Pacific region

relationship between CT and other disciplines, especially in making connections between established pre-college school subject content areas of STEM and newer CT-integrated disciplines (Lee et al. 2020).

At the time of writing this editorial, there are at least three more special issues of journals on CT that are in preparation. Two of these have a clear focus on CT and disciplinary education: "computational thinking for STEAM and engineering education" in *Computer Applications in Engineering Education*; and "data literacy & computational thinking in engineering education" in *Journal of Computing in Higher Education*.

These recent publications show a trend of research in developing students' CT, from efforts similar to those in the past that focused on programming and CS education (e.g., Ilic et al. 2018) to connecting and integrating CT with disciplinary education, especially STEM education. It is this latter trend that we aim to discuss further in this editorial.

The Complexity and Challenge in Connecting CT and STEM in Education

In comparison to efforts in developing students' CT predominantly through programming and in CS instruction, the slow and late-emerging trend of developing CT in disciplinary education is likely associated with other factors, such as politics and curriculum decisions. For example, in Switzerland, STEM is implemented in the German-speaking region as MINT (mathematik, informatik, naturwissenschaften, and technik) but not yet in the French-speaking part. In the latter, it is thus likely to integrate newly mandated CS education in another discipline with cross-disciplinary projects and CT may be considered as a transversal skill as others such as creativity and communication.[1] It is important for us to know how CT may be connected with STEM when discussing how to develop CT in connection with STEM education.

In the following sections, we discuss two shifts to illustrate the complexity and challenge in connecting CT and STEM in pre-college education. The first shift (disconnecting CT from non-CS STEM in education) has already taken place. It concerns the rapidly rising visibility of CT as important to all students, but enacted mainly in terms of programming and

[1] Email communication with Morgane Chevalier in Switzerland, July 2020.

within CS education. The second shift (re-connecting CT with non-CS STEM in education) started more recently. It concerns moving CT beyond programming and CS, per se, to integration in disciplinary education, especially integration with STEM.

Shift 1: CT Comes to Be Taken as Important Not Only for STEM Professionals But Also for Every Student, But It Is Enacted Instructionally Within a CS Disciplinary Perspective

In STEM professions, it is commonly recognized that computing and CT are important (Denning 2007; Froyd et al. 2012). In fact, as Denning (2009) pointed out, CT has long been used and commonly talked about in many professional fields, such as physics and biology, even without the participation of computer scientists. To many scientists, the use of computing is not only a tool, but also a way of making discoveries, involving problem solving, design, and model building. For example, computational modeling is used in biology research (Brodland 2015), climate risk management (Garner et al. 2016), and robust decision making under deep uncertainty (Lempert 2019). CT is naturally part of STEM, including of course, CS.

However, the significance of CT in STEM fields did not automatically translate into school instruction. CT was not considered to be important to everyone until Wing's call for its special role (Wing 2006). At the same time, as a computer scientist, Wing populated a CS discipline-based notion of CT that presents a foundational connection between CT and CS. Following Wing's conception of CT, it would become necessary to make CS education available for every student in order to develop their CT. There are, indeed, many educational initiatives that have been developed to make CS education available to many more students, simultaneously developing their CT (e.g., Grover and Pea 2013; U.S. National Science Foundation n.d.).

With Wing's particular disciplinary perspective on CT, the *subject fixation* of CT within CS becomes all but inevitable. In fact, her perspective simultaneously makes the connection between CT and other disciplines of professional STEM and STEM education much less direct and unclear. It would even lead to the question whether CS is already part of STEM, so that developing students' CT (under the CS discipline-based perspective) is readily part of STEM education. As we discuss further, below, the relationship between STEM and CS has not been straightforward and clear either in scholarly discussion or in educational practice.

T in STEM Is Not the Same as CS

Although there is "T" in STEM, meaning technology, the position of technology (and also engineering) in STEM has been less clear in comparison to the traditional subjects of mathematics (M) and science (S) in STEM (e.g., Bybee 2000; Daugherty 2009; Nager and Atkinson 2016).

If taking T as the use of computers, one may think that T includes CS as part of STEM. But a focus on the computer as a tool does not serve either T or CS well. As discussed in the previous editorial, CS is no longer taken as a study centering on computers, per se, but rather about processing information, both artificial and natural (Denning 2005). At the same time, T needs to be taken as a field of interdisciplinary study that goes well beyond computers, or even computation. For example, Bybee (2000) argued that "To be clear, the use of computers,

as one of many educational technologies, is essential in this age. However, it should not be confused with the study of technology, which provides students with opportunities to learn about the processes of design, fundamental concepts of technology and engineering, and the limits and possibilities of technology in society" (p. 23). In fact, many argued the importance of CS and considered CS as closely related but not the same as T (e.g., Guzdial and Morrison 2016; Nager and Atkinson 2016). Nager and Atkinson (2016) even argued that CS is not part of STEM, as it is not even represented in the acronym "STEM."

CS Had Not Been Considered as Important as Other Disciplines in STEM Education Until Recently

The importance of CS did not gain much recognition in many education systems including the USA in the past (e.g., Wilson et al. 2010). With the recognized importance of computing in many different fields (PITAC 2005), making CS education for all has gained momentum over the past decade. Specifically in the USA, Congress passed a "STEM Education Act of 2015" with an explicit indication: the term "STEM education" means education in the subjects of science, technology, engineering, and mathematics, including computer science (U.S. Congress 2015). Thus, although CS differs from T in STEM, including CS in STEM education has been officially recognized and promoted at the national level in the USA (U.S. Department of Education n.d.).

Viewing CS as equally important as other disciplines of STEM is now echoed in many influential documents and initiatives in the USA. For example, current administration announced an initiative for STEM education funding, with CS listed together with STEM in the announcement (White House 2017). However, the inclusion of CS in STEM education does not yet help make direct connections between CT and STEM in disciplinary education other than in CS.

Shift 2: Transforming Students' CT Development as Connected Directly with Not Only Programming and CS, But Also Other Disciplines of STEM in Education

Putting CS as part of STEM education helps promote CS as important for all students, but does not provide direct implications for developing CT through non-CS education, respecting the fact that CT can (and maybe should) also be developed through other STEM disciplines. Instead, taking CT as intimately connected with STEM disciplines, and not just CS, reflects what has been the reality in many professional fields of STEM even before Shift 1 discussed above. More importantly, it presents a re-direction in CT education to explicitly and substantially link to STEM education, beyond just CS.

As discussed in our previous editorial (Li et al. 2020a), this shift in educational efforts requires us to broaden the conception of CT so that it is not restricted to a CS disciplinary perspective. Such broadening can contribute to the framing of our review and discussion about educational efforts in developing students' CT.

Understanding CT Development as Involving Discipline-Specific Facets

Recognizing CT's importance does not immediately provide guidance for how to teach it. In fact, many teachers in pre-college education have difficulties in developing and

implementing particular educational activities for teaching CT to students (Denning 2017; Hsu et al. 2018). The complexity in CT education relates not only to the multi-faceted nature of CT, but also to the diversity of related interventions and educational programs (e.g., Grover and Pea 2013; NRC 2011; Shute et al. 2017). In our view, complexity in understanding the nature of CT and complexity in instructional orientations are closely related, with one leading to another. Researchers, instructional developers, educators can pre-specify facets of CT and then develop or identify activities or tools for instructional use. Conversely, facets of CT may not be consciously pre-specified or chosen, but resulted from implementing educational activities and tools determined by a third party. By using the nature of CT as a lens, we can frame our review and discussion of different educational initiatives and programs in the following two sections: developing CT with a specific disciplinary focus and with a focus on trans-disciplinary thinking practices.

Developing CT as a Discipline-Specific Thinking Practice

Viewing CT as a discipline situated thinking process and practice is common for researchers and educators in STEM disciplines. Because CT is commonly used by professionals not only in CS but also in other STEM disciplines, developing students' CT can potentially take approaches that differ in terms of the view of CT as related to CS in specific or other individual disciplines of STEM.

Developing Discipline-Specific CT in and Through Programming- or Coding-Oriented Activities and CS Education

CT is often viewed as directly related to programming, coding and CS, especially after the publication of Wing's seminal paper (Wing 2006). Efforts to develop students' CT thus tend to develop and use activities, tools or platforms related to programming knowledge and skills in educational interventions and programs (e.g., Barth-Cohen et al. 2018; Hsu et al. 2018; Lye and Koh 2014; Shute et al. 2017). As an example, Barth-Cohen et al. (2018) investigated how fifth graders interpret and navigate information when participating in various coding and problem-solving activities in a programming environment. In their study, the school adopted and used a robotics curriculum, had software installed in school-provided laptops for students, and had one physical robot for instructional use. Students' CT development was examined with an emphasis on their performance in formulating and solving problems in this robotic programming environment.

It should be pointed out that, even with the CS discipline-based view of CT, previous work has proposed various models and frameworks to further conceptualize and operationalize CT (Grover and Pea 2013; Shute et al. 2017). It is thus not surprising to find many contrasting educational activities and programs for teaching and assessing CT. Several review articles summarized how active relevant research and educational efforts are in developing students' CS discipline-based CT. For example, a literature review of CT research by Shute et al. (2017) includes a synthesis of various approaches used to develop CT in K-16 settings. Among 11 studies being reviewed, 10 studies used specific programming tools or platforms for CT training purpose, such as Scratch, C, Alice (Carnegie Mellon University 1999), paper-and-pencil programming, and

various robotics packages. Likewise, in a meta-review of CT education studies published in journals between 2006 and 2017, Hsu et al. (2018) found that CT has mainly been applied to activities in program design and CS. Most of the 120 articles focused on programming skill training and mathematical computing, while some adopted a cross-domain teaching mode to enable students to manage and analyze materials of various domains by computing.

Although programming is commonly used as important activities for developing students' CT, some researchers have pointed out that the acquisition of superficial or language-oriented programming skills is often ineffective in helping students to transit from novice programming to problem-solving-oriented programming (e.g., Buitrago Flórez et al. 2017; Michaelson 2018; Moors and Sheenan 2017). CT development should not be limited to acquiring programming skills, but more on problem solving through computational means. Thus, some researchers proposed alternative approaches to develop problem-solving-oriented CT such as, "systematic CT" (Michaelson 2018) and "creative computational problem solving" (Chevalier et al. 2020). For example, Chevalier et al. (2020) took the view of CT development using a robot as more about the problem-solving process than just programming the robot to solve the problem. They proposed a creative computational problem solving model for teaching CT. With this model, they conducted an experimental study with elementary school students using an educational robot and a programming interface. They imposed pre-designed restrictions to the experimental group for accessing the programming interface or executing the code on the robot, while the control group was given the same task and working environment without any accessing restrictions. They found that students in the control group spent most of their time in programming and evaluating in a trial-and-error loop and could hardly move out of this loop to systematically analyze the problem and test strategies for solving it. In contrast, the experimental group was forced to shift their attention toward understanding the problem, generating ideas and formulating strategies, instead of jumping to programming the robot. Their results suggest the importance of not only having a good understanding of CT (e.g., not focusing exclusively on acquiring programming skills), but also thinking and planning carefully in developing and implementing educational activities to develop students' CT.

In our previous editorial (Li et al. 2020a), we pointed out that computational literacy is a concept proposed and used even before CT (diSessa 2000, 2018). This concept highlights the importance of computation for students' learning beyond programming. The concept is not the same as Wing's (2006) conception of CT in that computation is not viewed as the special province of computer scientists, and everyone does not need to think like a computer scientist. In recent report *Charting a Course for Success: America's Strategy for STEM Education* (Committee on STEM Education 2018), both computational literacy and CT were included and used when laying out the vision for the USA to success in STEM education.

Developing Discipline-Specific CT in and Through Education in Non-CS STEM Disciplines

As discussed and summarized above, there is a growing trend of work that tries to integrate CT with disciplinary education. One approach commonly used is to follow the conception of CT outlined by Wing (2006), and then discusses possible activities and

approaches to develop students' CT within different disciplines (e.g., Barr and Stephenson 2011; Fennell et al. 2020; Lee et al. 2020; Swaid 2015). For example, early efforts in this direction involved the identification of possible inclusion of CT in various school subjects, with explanation about which aspects of CT can be included for a specific school subject (e.g., Barr and Stephenson 2011; Swaid 2015). Typically, there is a lack of empirical investigations (Grover and Pea 2013).

Recent efforts in this direction involve more in specific integration designs and/or empirical follow-up studies (e.g., Fennell et al. 2020; Lee et al. 2020; Tucker-Raymond et al. 2019). For example, in the special issue of *Journal of Science Education and Technology*, researchers reported particular strategies taken to make connections between existing STEM subjects in pre-college education and CT, CT-integrated disciplines such as computational sciences (Lee et al. 2020). As another example, Fennell et al. (2020) pointed out the lack of effective pedagogy in postsecondary engineering education that can help develop students' "computational adaptive expertise," one's ability to flexibly use computational knowledge in novel situation in the context of engineering design (McKenna et al. 2008). They drew upon recent research to formulate a computational apprenticeship framework. It is a constructivist research and practice model to introduce students to meaningful computational practices through a series of discipline-situated programming projects.

When discussing CT as associated directly with STEM disciplines, and not just CS, it is also reasonable to ask if the nature of CT practices in STEM disciplines differ from CT practices in CS. In fact, some researchers tried to examine the practices of CT by STEM professionals to help inform CT development and assessment in disciplinary education (e.g., Beheshti et al. 2017; Malyn-Smith and Lee 2012; Weintrop et al. 2016). For example, Weintrop et al. (2016) took the approach, in contrast to others, that emphasized topics such as abstraction and algorithms, defining CT as a taxonomy of CT practices in mathematics and science. They drew on the existing CT literature, exemplary CT instructional activities and materials, and interviews with mathematicians and scientists, to develop the taxonomy of 22 CT practices organized into four major categories: data practices, modeling and simulation practices, computational problem solving practices, and systems thinking practices. Weintrop et al. discussed, with specific examples, how such a taxonomy can help provide three main benefits of integrating CT with STEM education: (1) building a reciprocal relationship for learning mathematics and science with CT, (2) establishing a sustainable learning environment that can engage all students, and (3) bringing efforts in developing CT in disciplinary education in line with the increasingly computational nature of scientific and mathematical practices.

Along this line of work, characterizing the nature of CT in professional practices, Beheshti et al. (2017) further interviewed 17 professional practitioners in STEM to identify possible characteristics of CT practices that are important to them. These 17 interviewees had expertise in a range of STEM disciplines such as biochemistry, materials science, chemistry, computer science, and transportation engineering. Based on the same taxonomy of 22 CT practices grouped into four categories (Weintrop et al. 2016), Beheshti et al. (2017) found that each main category of CT practices is identified at similar levels across different disciplines: around 30% use in data analysis, 20% in modeling and simulation, 30–35% in computational problem solving, and 15–20% in systems thinking. At the same time, when focusing on the role of computation in CT

practices of these interviewees' research, they found that STEM researchers working more on experimentation than computation would use data practices more frequently, those with higher use of computation entailed a greater use of systems thinking practices and modeling-and-simulation practices. Theoretical researchers mainly drew on computational problem solving practices. Beheshti et al. (2017) indicated that these results help identify and inform educational efforts in high school mathematics and science learning contexts.

Across different disciplines that students are exposed to, it is also important to consider different families of CT practices that likely exist within them. CT can vary in different disciplines such as chemistry, physics, and logic, as revealed by Beheshti et al. (2017). It is not surprising that CT would become discipline-specific when viewed within different disciplinary education. CT competence, if taken as trans-disciplinary practices, would then require exposure for students to a variety of different disciplines so they can apply the right ones when solving practical problems.

Developing CT as a Trans-disciplinary Thinking Practice

By emphasizing the process of thinking, we proposed that CT involves searching for ways of processing information that are incrementally improvable in their efficiency, correctness, and elegance (Li et al. 2020a). This definition helps broaden the conceptualization of CT to go beyond specific subjects in which CT may operate, thus avoiding a *subject fixation* concerning CT. In fact, some scholars shared similar thoughts in their conceptions of CT. For example, Sengupta et al. (2018) took an epistemological shift to view coding and CT more as a complex form of experience, rather than as mastery over computational logic and symbolic form. Shute et al. (2017) conducted a literature review concerning CT and then defined CT primarily as a way of thinking and acting, with or without the assistance of computers. Then, they further specified CT primarily in computational problem solving with six main aspects: decomposition, abstraction, algorithm design, debugging, iteration, and generalization.

To make CT truly accessible and important to all students, educational initiatives and programs need to be developed and made available in pre-college STEM education with the broadened view of CT as a trans-disciplinary thinking process and practice. The following sections discuss some educational initiatives from the literature and also discuss efforts in facilitating students' learning of STEM with CT.

Developing Trans-disciplinary CT in and Through STEM Education

With the recognition that CT is not only in CS but also in many different subjects and everyday activities, Lu and Fletcher (2009) argued that CT development should be articulated and reflected in students' learning of different subject contents before teaching programming language. They provided specific examples to illustrate their argument for developing (aspects of) CT as a general competence, without the use of programming language. For example, in mathematics when solving algebraic word problems, students can learn and experience the use of a trial-and-error (*blind*) approach, or a *heuristic* approach to identify and use specific strategy. They can learn how to *represent* word problems, and apply algebraic rules to *derive* simpler forms. To Lu and Fletcher (2009), these notions (especially those rendered in *italic*) are similar to CT

as experienced in CS. As another example, they argued that group work in science courses can also be used in developing CT. Specifically, data-exchange interactions among group members "are ideal situations for formally introducing notions of *interface* and *encapsulation*." (p. 264).

Sengupta et al. (2018) took a paradigmatic shift in the epistemology and pedagogy of computing and CT, especially for pre-college STEM education. They built on the *Science as Practice* (Duschl 2008; Lehrer 2009) conception to argue for adopting a perspective in which epistemic and representational work are deeply intertwined. Specifically, they proposed to shift away from technocentrism (Papert 1987) that focuses more on the production of a set of axiomatic computational abstractions, to a view from the perspective of teachers and students of computing and CT more "as discursive, perspectival, material and embodied experiences, among others" (p. 49). They discussed several previous studies as examples grounded in this perspective, and highlighted that computing and CT can take different forms, involving different sequences of learning activities and experiences. For example, teachers can use embodied modeling activity to help students in a third grade science classroom (Dickes et al. 2016). In that learning environment, students engaged iteratively in cycles of embodied modeling of foraging behavior and graphing, and then modeled the same phenomena using multi-agent-based NetLogo simulations. They indicate that such event-based programming and modeling in that study can support students in developing conceptual understanding of complex scientific phenomena, by valuing (rather than ignoring) the roles of uncertainty and interpretive dilemmas that are inherent in students' modeling work (Farris et al. 2019).

Facilitating Students' Learning of STEM with the Integration of CT

Although the term CT was not used earlier in the history of computers and education, researchers have long been thinking about and developing educational activities and programs that combine the learning of programming and computational modeling together with mathematics and science in school education. For example, as a pioneer in developing students' procedural thinking through LOGO programming (Papert 1980), Papert and colleagues also developed LOGO-based learning environments for subject content such as fraction in elementary school curricula (Harel and Papert 1990). The results from experimental studies led them to argue that integrated learning of programming with concepts in another domain can be easier and more effective than learning them separately. They believe that computation and programming have a reflexive synergistic quality to facilitate other knowledge acquisition. Abelson and diSessa (1981) produced a fully computationally infused version of a curriculum in geometry (known as "turtle geometry"), arguing both that this combination eased learning, but also that it changed the personal relationship between students and mathematics, turning "mathematics" into a more personal and experiential process. Some researchers also strived to develop and use other visual programming such as Boxer (e.g., diSessa 2000; diSessa et al. 1991) and block-based programming (Weintrop and Wilensky 2017).

Since current connotations of CT are relatively recent, integrating it with disciplinary thinking and learning are also recent, especially when CT is viewed as a transdisciplinary thinking practice. We can identify two possible directions with the existence of limited research in this area. One direction is to take the view of CT as a trans-

disciplinary thinking practice, and then identify and design ways of integrating CT with discipline-specific STEM content learning. When focusing on a specific discipline, educational design and effort can place more attention on improving students' learning of disciplinary content in STEM with the integration of CT. When looking at CT integration across different disciplines, educational design and effort can place specific attention to developing students' CT as a general competence. In this way, development of CT instruction can potentially let students apply the proper practices when solving problems in the future. Several publications cited in the last section can provide illustrations (e.g., Lu and Fletcher 2009; Sengupta et al. 2018). To develop and use a systematic approach in this direction, it would also be valuable to explore and identify essential productive practices of trans-discipline CT and their connections with individual disciplines. Such essential productive practices can then be both an identifying feature and a bridge between CT and the disciplines. As an example, similar effort was taken in mathematics about productive patterns of mathematical thinking (Schoenfeld 2017).

Another direction is to take CT as naturally integrated in different STEM disciplines, and then reconceptualize and shift STEM content learning from the traditional subject format to computational-based STEM content learning. Abelson and diSessa's "turtle geometry," cited above, and more recent work by diSessa and others (see cited in diSessa 2018) are examples. In this direction, CT may be taken as a core skill for all students with a specific set of CT integration elements identified and used for bridging (e.g., Malyn-Smith et al. 2018). The special issue on "computational thinking from a disciplinary perspective" published recently in the *Journal of Science Education and Technology* contains some further information on this approach (Lee et al. 2020).

If CT is still viewed as discipline-specific and connected specifically with programming, we can identify a growing number of studies that investigate and document the effect of facilitating students' learning of STEM integrated with CT. For example, Sengupta et al. (2013) proposed the use of agent-based computation to integrate CT with existing pre-college science curricula. Specifically, following Wing's notion of CT, they proposed a theoretical framework with four components and a sample learning environment: CTSiM (computational thinking in simulation and modeling), a visual-programming based learning environment for middle school science. The results from their pilot study with sixth grade students in two content units (kinematics and ecology) suggested significant student learning gains, as measured by students' pre- and post-test scores. As another example, Miller et al. (2020) presented an approach of integrating CT, as a set of practices delineated by Grover and Pea (2013, 2018), in promoting students' learning in science education. Specifically, they demonstrated its feasibility and benefits through a case study of a group of US students in a fifth-grade project-based learning of science. Their case study provided qualitative analyses of how various practices associated with CT helped these students in developing shared understanding of the particle nature of matter.

Assessment as an Important Dimension of CT, CT Research, and Instruction

Assessment has been an important dimension of CT itself since its early stage of development. The idea of Generate and Test was advanced early in information

processing theory, such as the T.O.T.E. for "Test - Operate - Test - Exit" proposed by Miller et al. (1960) for having monitoring devices that control the acquisition of the stimulus-response relationship populated in behaviorism.

A computational procedure or algorithm takes input and *generates* an output, followed by an evaluation of the output in the *test* phase. Proficiency in CT requires some fluency in choosing or developing alternative procedures and algorithms (with different sets of these in different disciplines), as well as criteria for evaluating the output to assess its relevance, appropriateness, utility, efficiency, and so on. The test component often requires multiple representations and perspectives. When a mathematical procedure is performed, those manipulations in the procedure may involve a sensory-motor skill such as evaluating whether a robot performance matches expectations, whereas magnitude estimation of the output is a separate skill. Thus, CT is likely to involve multiple cognitive processes and representations rather than mechanically applying a single computational component. Implementing the test component of generate and test, or T.O.T.E., remains an important ingredient in CT.

In CT research and instruction, researchers have realized the importance of CT assessment (e.g., Grover and Pea 2013; Shute et al. 2017; Werner et al. 2012). Grover and Pea (2013) summarized specific techniques and methods as developed and used in different projects, such as the use of student-created or predesigned programming artifacts to evaluate students' understanding and use of CT components (Werner et al. 2012), and students' debugging prebuilt faulty e-textile projects to evaluate students' engineering and programming skills (Fields et al. 2012). Some important efforts in developing or promoting CT assessment have also been carried out by education professional organizations, assessment service entities, or nonprofit scientific research organizations such as SRI education (e.g., Bienkowski et al. 2015).

Recent development in CT education has led to further attention to operationalizing CT. A common approach is to provide a list of component skills, possibly conceptualized as a model. The existence of diverse definitions and models of CT might make it difficult to get a consensus on CT assessment, and consequently make comparisons across different assessments also difficult if not impossible. Shute et al. (2017) summarized some research efforts in CT assessment, including the use of questionnaires and surveys for measuring knowledge of and attitudes toward CT, interviews and observations with participants to understand their CT skill development, particular activity-based assessment (e.g., Scratch-based, or game-based), and the development and validation of CT scales for generic usage. Shute et al. (2017) illustrated assessments of the conceptual foundation of several facets of their competency-based model of CT (decomposition, abstraction, algorithms, debugging, iteration, and generalization) with sample activities."

In a literature review of empirical studies about CT assessment, Tang et al. (2020b) identified a total of 96 journal articles published before August 2019 through searching ERIC, PsycINFO, and Google Scholar. Their review showed some findings consistent with what others reported (e.g., Shute et al. 2017) and also identified aspects that call for further research and development efforts such as the following: CT assessment being mainly focused on assessing students' programming or computing skills but not others; the need for reliability and validity evidence for CT assessments; and the need for CT assessments for high school, college, and teacher education in addition to K-8 school levels. At the same time, it is important to point out that CT assessment is

receiving ever-increased attention from many researchers. Readers can find an increasing number of publications in this topic area, including design and content validation of CT tests for beginners (Relkin et al. 2020; Zapata-Cáceres et al. 2020).

Teacher Education in CT and CT-Integrated Disciplinary Education

Efforts to develop students' CT in school education require special attention to teachers who need to be prepared and supported to take on the challenge. As CT itself is relatively new in pre-college education and undergraduate education, it is not surprising that much more research is needed to understand how to prepare and support teachers in CT (Angeli and Giannakos 2020; Barr and Stephenson 2011; Yadav et al. 2014, 2017).

CT is not a discipline by itself. Compared to disciplinary training for teachers, there are unique challenges in finding ways of helping teachers to develop students' CT. One common approach is to view teachers' instructional expertise as knowledge and skill based, similar to the case in disciplinary education, such as mathematics instruction (Li and Kaiser 2011). With this perspective, some researchers have tried to find out what CT knowledge and skills teachers may need *for themselves*, and then what they may need to integrate and implement CT in subject content teaching *for their students* (Mouza et al. 2017; Yadav et al. 2017). For example, Mouza et al. (2017) redesigned an educational technology course for pre-service teachers (PSTs) to prepare PSTs to integrate CT in school curricula. Based on the technological pedagogical content knowledge (TPACK) framework, Mouza et al. tried to explicate the framework with CT-related concepts, computing tools, and practices. They hypothesized that teachers who need to integrate CT in disciplinary content teaching would need to have two sets of knowledge in relation to their framework: (1) technology knowledge related to CT (TK-CT), and (2) disciplinary content knowledge and pedagogical strategies (both general and content-specific) in relation to CT. Together, they called this body of knowledge TPACK-CT. In their study with 21 PSTs in this redesigned course, they designed specific instruction to model for PSTs the use of TK-CT across course activities as integrated with specific disciplinary content and pedagogy (TPACK-CT). PSTs also had opportunities to design and implement their own CT integrated disciplinary content lessons in K-8 classrooms. Based on a self-reported survey and case reports from PSTs, Mouza et al. indicated the course's positive effect on PSTs' TPACK-CT. At the same time, they indicated that some PSTs experienced difficulty and could not design CT-integrated lessons as expected.

Helping in-service teachers to learn and implement CT-integrated content lessons could present a variety of challenges, as professional development (PD) can vary dramatically based on teachers' needs and school contexts. Yadav et al. (2018) tried to learn what changes may happen in teachers' understanding of CT over a year-long PD course. The PD focused on the use of CT ideas and practices for supporting students' learning in science and mathematics. To measure how teachers' thinking about CT emerged, Yadav et al. used two open-ended teaching vignettes. Based on teachers' responses, they noticed shifts in teachers' understanding of CT from general to more elaborated ideas. The results suggest the possibility of helping teachers to learn and develop an understanding of CT and its integration in elementary science. They

also pointed out the need to examine how these teachers may be able to translate their CT competence into their classroom instruction.

As illustrated above in these studies, current research related to teacher preparation and training in CT and CT education is still at the exploratory stage. It will take time, effort, commitment, and collaboration for researchers to learn a sufficient amount about specific challenges teachers face and what it might take to address them.

Opportunities and Challenges in Broadening the Perspective About CT and the Need for New Research and Scholarship

We started with a brief overview of research trends in CT education. It helped us to uncover and delineate the dramatic development in CT research and instruction over the past decade. The recent trend of integrating and developing CT with disciplinary education in STEM suggests the importance of broadening the notion of CT beyond programming in the context of CS. At the same time, two shifts have happened in *disconnecting* and then *re-connecting* CT with STEM in education. These shifts imply that appropriately broadening our perspective on CT brings both opportunities and challenges in research and practice.

Some opportunities and challenges become visible in our review of CT research and instruction, through the lens of CT's disciplinary connections. As existing efforts in CT research and instruction dominantly take a CS discipline-based view of CT, opportunities for developing both CS and CT in school education will grow with a more encompassing view of the relationship between CT and disciplines (U.S. National Science Foundation, n.d.). But a great challenge remains to make CT consequential and accessible to all students. It becomes important to broaden the view of CT as a transdisciplinary thinking practice, as we elaborated in the last editorial (Li et al. 2020a). At the same time, what we can learn through the review above is that integrating CT with disciplinary education in STEM has scarcely been explored in undergraduate and precollege education. There are abundant opportunities for exploration in both research and instruction, not because we can build on a solid, existing foundation, but because we know so little. Similarly, challenges still exist in bringing this broadened perspective into productive outcomes in CT-integrated research and instruction. For example, Angeli and Giannakos (2020) recently summarized some challenges such as specifying CT competencies for different grades or student development levels, developing and using pedagogical approaches for teaching CT, teacher training in CT and CT education, and CT assessment. Each of these challenges will indeed require tremendous effort, in particular with respect to assessment and teacher education in CT and CT instruction. The same is the case for specifying CT competencies that serve as a foundation for curriculum, instruction, and assessment in CT. Similar efforts on learning trajectories in mathematics and their application in early childhood mathematics interventions may provide us hints for what it may take for such a development (Clements and Sarama 2011; Daro et al. 2011). Nevertheless, recent trends in integrating CT into STEM education suggest an enthusiasm by many researchers and educators for pursuing opportunities while embracing challenges.

It is also important to point out that, in our view, CT with a broadened conception presents a model of thinking that is important for all students. STEM (including CS)

education is uniquely positioned to develop students' (models of) thinking (Li et al. 2019). Integrating CT in disciplinary education of STEM is a new topic, but it is very important in educating new generations of students in the twenty-first century. The topic is also enticingly open, calling for broad, cross-disciplinary research collaborations. It is certainly a frontier topic for which this journal welcomes manuscript submissions (Li 2018).

Acknowledgments We would like to thank Lauren A. Barth-Cohen, Morgane Chevalier, and Pratim Sengupta for their valuable feedback on an earlier version of this editorial.

References

Abelson, H., & diSessa, A. A. (1981). *Turtle geometry: The computer as a medium for exploring mathematics*. Cambridge: MIT Press.

Angeli, C., & Giannakos, M. (2020). Computational thinking education: Issues and challenges. *Computers in Human Behavior, 105*, 106185.

Barr, V. & Stephenson, C. (2011). Bringing computational thinking to K-12: What is involved and what is the role of the computer science education community? *ACM Inroads, 2*(1), 48–54.

Barr, D., Harrison, J., & Conery, L. (2011). *Computational thinking: A digital age skill for everyone* (pp. 20–23). Learning and Leading with Technology.

Barth-Cohen, L. A., Jiang, S., Shen, J., Chen, G., & Eltoukhy, M. (2018). Interpreting and navigating multiple representations for computational thinking in a robotics programing environment. *J STEM Educ Res, 1*(1), 119–147.

Beheshti E, Weintrop, D., Swanson, H., Orton, K., Horn, M. S., Jona, K, & Wilensky, U. (2017). *Computational thinking in practice: How STEM professionals use CT in their work*. Available at https://ccl.northwestern.edu/2017/Beheshtietal.pdf. Accessed on April 2, 2020.

Bienkowski, M., Snow, E., Rutstein, D. W., & Grover, S. (2015). Assessment design patterns for computational thinking practices in secondary computer science: A first look (SRI technical report). Menlo Park, CA: SRI International. Available at http://pact.sri.com/resources.html Accessed on March 28, 2020.

Brodland, G. W. (2015). How computational models can help unlock biological systems. *Seminars in Cell & Developmental Biology, 47-48*, 62–73.

Buitrago Flórez, F., Casallas, R., Hernández, M., Reyes, A., Restrepo, S., & Danies, G. (2017). Changing a generation's way of thinking: Teaching computational thinking through programming. *Review of Educational Research, 87*(4), 834–860. https://doi.org/10.3102/0034654317710096.

Bybee, R. W. (2000). Achieving technological literacy: A national imperative. *The Technology Teacher, 60*(1), 23–28.

Carnegie Mellon University (1999), Alice, https://www.alice.org.

Chevalier, M., Giang, C., Piatti, A., & Mondada, F. (2020). Fostering computational thinking through educational robotics: A model for creative computational problem solving. *International Journal of STEM Education, 7*, 39. https://doi.org/10.1186/s40594-020-00238-z.

Clements, D. H., & Sarama, J. (2011). Early childhood mathematics intervention. *Science, 333*, 968–970.

Committee on STEM Education, National Science & Technology Council, the White House. (2018). *Charting a course for success: America's strategy for STEM education*. Washington, DC. Available at https://www.whitehouse.gov/wp-content/uploads/2018/12/STEM-Education-Strategic-Plan-2018.pdf Accessed on February 18, 2020.

Daro, P., Mosher, F. A., & Corcoran, T. (2011). Learning trajectories in mathematics: A foundation for standards, curriculum, assessment, and instruction. *CPRE Research Reports*. Available at http://repository.upenn.edu/cpre_researchreports/60. Accessed on July 26, 2020.

Daugherty, M. K. (2009). The "T" and "E" in STEM. In ITEEA (Ed.), *The overlooked STEM imperatives: Technology and engineering* (pp. 18–25). ITEEA: Reston.

Denning, P. J. (2005). Is computer science science? *Communications of the ACM, 48*(4), 27–31.

Denning, P. J. (2007). Computing is a natural science. *Communications of the ACM, 50*(7), 13–18.

Denning, P. J. (2009). The profession of IT beyond computational thinking. *Communications of the ACM, 52*, 28–30.

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

Dickes, A. C., Sengupta, P., Farris, A. V., & Basu, S. (2016). Development of mechanistic reasoning and multilevel explanations of ecology in third grade using agent-based models. *Science Education, 100*(4), 734–776.

diSessa, A. A. (2000). *Changing minds: Computers, learning, and literacy*. Cambridge: MIT Press.

diSessa, A. A. (2018). Computational literacy and "The Big Picture" concerning computers in mathematics education. *Mathematical Thinking and Learning, 20*(1), 3–31. https://doi.org/10.1080/10986065.2018.1403544.

diSessa, A. A., Abelson, H., & Ploger, D. (1991). An overview of boxer. *Journal of Mathematical Behavior, 10*(1), 3–15.

Duschl, R. (2008). Science education in three-part harmony: Balancing conceptual, epistemic, and social learning goals. *Review of Research in Education, 32*(1), 268–291.

English, L. (2018). On MTL's second milestone: Exploring computational thinking and mathematics learning. *Mathematical Thinking and Learning, 20*(1), 1–2.

Farris, A. V., Dickes, A. C., & Sengupta, P. (2019). Learning to interpret measurement and motion in fourth grade computational modeling. *Science & Education, 28*(8), 927–956.

Fennell, H. W., Lyon, J. A., Madamanchi, A., & Magana, A. J. (2020). Toward computational apprenticeship: Bringing a constructivist agenda to computational pedagogy. *Journal of Engineering Education*. https://doi.org/10.1002/jee.20316.

Fields, D. A., Searle, K. A., Kafai, Y. B., & Min, H. S. (2012). Debuggems to assess student learning in e-textiles. In *Proceedings of the 43rd SIGCSE Technical Symposium on Computer Science Education*. New York: ACM Press.

Froyd, J. E., Wankat, P. C., & Smith, K. A. (2012). Five major shifts in 100 years of engineering education. *Proceedings of the IEEE, 100*(Special Centennial Issue), 1344–1360. https://doi.org/10.1109/JPROC.2012.2190167.

Garner, G., Reed, P., & Keller, K. (2016). Climate risk management requires explicit representation of societal trade-offs. *Climatic Change, 134*, 713–723. https://doi.org/10.1007/s10584-016-1607-3.

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

Grover, S., & Pea, R. (2018). Computational thinking: A competency whose time has come. In S. Sentance, E. Barendsen, & C. Schulte (Eds.), *Computer science education: Perspectives on teaching and learning in school* (pp. 19–38). London: Bloomsbury Academic.

Guzdial, M., & Morrison, B. (2016). Growing computer science education into a STEM education discipline. *Communications of the ACM, 59*(11), 31–33.

Harel, I., & Papert, S. (1990). *Software design as a learning environment*. Epistemology & Learning Memo No. 7. Available at http://dailypapert.com/wp-content/uploads/2015/10/Software-Design-as-a-Learning-Environment.pdf. Accessed on April 2, 2020.

Hsu, T.-C., Chang, S.-C., & Hung, Y.-T. (2018). How to learn and how to teach computational thinking: Suggestions based on a review of the literature. *Computers & Education, 126*, 296–310.

Ilic, U., Haseski, H. I., & Tugtekin, U. (2018). Publication trends over 10 years of computational thinking research. *Contemporary Educational Technology, 9*(2), 131–153.

Kalelioglu, F., Gülbahar, Y., & Kukul, V. (2016). A framework for computational thinking based on a systematic research review. *Modern Computing, 4*(3), 583–596.

Khine, M. S. (2018). *Computational thinking in the STEM disciplines: Foundations and research highlights*. Cham: Springer.

Lee, I., Grover, S., Martin, F., Pillai, S., & Malyn-Smith, J. (2020). Computational thinking from a disciplinary perspective: Integrating computational thinking in K-12 science, technology, engineering, and mathematics education. *Journal of Science Education and Technology, 29*(1), 1–8.

Lehrer, R. (2009). Designing to develop disciplinary dispositions: Modeling natural systems. *American Psychologist, 64*(8), 759–771.

Lempert, R. J. (2019). Robust decision making (RDM). In V. Marchau, W. Walker, P. Bloemen, & S. Popper (Eds.), *Decision making under deep uncertainty* (pp. 23–51). Springer: Cham. https://doi.org/10.1007/978-3-030-05252-2_2.

Li, Y. (2018). Journal for STEM education research – Promoting the development of interdisciplinary research in STEM education. *Journal for STEM Education Research, 1*(1–2), 1–6. https://doi.org/10.1007/s41979-018-0009-z.

Li, Y., & Kaiser, G. (2011). *Expertise in mathematics instruction: An international perspective*. New York: Springer.

Li, Y., Schoenfeld, A. H., diSessa, A. A., Grasser, A. C., Benson, L. C., English, L. D., & Duschl, R. A. (2019). On thinking and STEM education. *Journal for STEM Education Research, 2*(1), 1–13. https://doi.org/10.1007/s41979-019-00014-x.

Li, Y., Schoenfeld, A. H., diSessa, A. A., Grasser, A. C., Benson, L. C., English, L. D., & Duschl, R. A. (2020a). Computational thinking is more about thinking than computing. *Journal for STEM Education Research, 3*(1), 1–18. https://doi.org/10.1007/s41979-020-00030-2.

Li, Y., Wang, K., Xiao, Y., & Froyd, J. E. (2020b). Research and trends in STEM education: A systematic review of journal publications. *International Journal of STEM Education, 7*, 11. https://doi.org/10.1186/s40594-020-00207-6.

Lu, J. J., & Fletcher, G. H. (2009). Thinking about computational thinking. *ACM SIGCSE Bulletin, 41*(1), 260–264.

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

Malyn-Smith, J. and Lee, I. A. (2012). Application of the occupational analysis of computational thinking-enabled STEM professionals as a program assessment tool. *Journal of Computational Science Education, 3*(1), 2–10.

Malyn-Smith, J., Lee, I. A., Martin, F., Grover, S., Evans, M. A., & Pillai, S. (2018). Developing a framework for computational thinking from a disciplinary perspective. In S. C. Kong, D. Andone, G. Biswas, T. Crick, H. U. Hoppe, T. C. Hsu, R. H. Huang, K. Y. Li, C. K. Looi, M. Milrad, J. Sheldon, J. L. Shih, K. F. Sin, M. Tissenbaum, & J. Vahrenhold (Eds.), *Proceedings of the International Conference on Computational Thinking Education 2018*. Hong Kong: The Education University of Hong Kong.

McKenna, A., Linsenmeier, R., & Glucksberg, M. (2008). Characterizing computational adaptive expertise. *Proceedings of the ASEE Annual Conference & Exposition*, Pittsburgh, PA. Available at https://peer.asee.org/4415. Accessed on July 22, 2020.

Michaelson, G. (2018). Microworlds, objects first, computational thinking and programming. In M. S. Khine (Ed.), *Computational thinking in the STEM disciplines* (pp. 31–48). Cham: Springer.

Miller, E., Severance, S., & Krajcik, J. (2020). Connecting computational thinking and science in U.S. elementary classroom. In J. Anderson & Y. Li (Eds.), *Integrated approaches to STEM education: An international perspective*. Cham: Springer.

Miller, G. A., Galanter, E., & Pribram, K. H. (1960). *Plans and the structure of behavior*. New York: Holt, Rinehart & Winston.

Moors, L., & Sheenan, R. (2017). Aiding the transition from novice to traditional programming environments. In Proceedings of IDC 2017: ACM interaction design and children conference (pp. 509–514). Stanford University.

Mouza, C., Yang, H., Pan, Y.-C., Yilmaz Ozden, S., & Pollock, L. (2017). Resetting educational technology coursework for pre-service teachers: A computational thinking approach to the development of technological pedagogical content knowledge (TPACK). *Australasian Journal of Educational Technology, 33*(3), 61–76.

Nager, A. & Atkinson, R. D. (2016). The case for improving U.S. computer science education. Information Technology & Innovation Foundation report. Available at http://www2.itif.org/2016-computer-science-education.pdf?_ga=2.92886875.757421025.1591995042-70633712.1591995042. Accessed on June 2, 2020.

National Research Council (NRC). (2010). *Report of a workshop on the scope and nature of computational thinking*. Washington, DC: The National Academies Press.

National Research Council (NRC). (2011). *Report of a workshop on the pedagogical aspects of computational thinking*. Washington, DC: The National Academies Press.

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

Papert, S. (1987). Computer criticism vs. technocentric thinking. *Educational Researcher, 16*(1), 22–30.

President's Information Technology Advisory Committee (PITAC). (2005). *Computational science: Ensuring America's competitiveness* (Report to the President, June 2005). Washington, DC: National Coordination Office for Information Technology Research and Development (NCO/IT R&D). Available at https://www.nitrd.gov/pitac/reports/20050609_computational/computational.pdf. Accessed on February 2, 2020.

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

Schoenfeld, A. H. (2017). Teaching for robust understanding of essential mathematics. In T. McDougal (Ed.), *Essential mathematics for the next generation: What and how students should learn* (pp. 104–129). Tokyo: Tokyo Gagukei University.

Sengupta, P., Kinnebrew, J. S., Basu, S., Biswas, G., & Clark, D. (2013). Integrating computational thinking with K-12 science education using agent-based computation: A theoretical framework. *Education and Information Technologies, 18*(2), 351–380.

Sengupta, P., Dickes, A., & Farris, A. (2018). Toward a phenomenology of computational thinking in STEM education. In M. S. Khine (Ed.), *Computational thinking in the STEM disciplines: Foundations and research highlights* (pp. 49–72). Cham: Springer.

Shute, V. J., Sun, C., & Asbell-Clarke, J. (2017). Demystifying computational thinking. *Educational Research Review, 22*, 142–158.

Swaid, S. I. (2015). Bringing computational thinking to STEM education. *Procedia Manufacturing, 3*, 3657–3662.

Tang, K.-Y., Chou, T.-L., & Tsai, C.-C. (2020a). A content analysis of computational thinking research: An international publication trends and research typology. *The Asia-Pacific Education Researcher, 29*(1), 9–19.

Tang, X., Yin, Y., Lin, Q., Hadad, R., & Zhai, X. (2020b). Assessing computational thinking: A systematic review of empirical studies. *Computers & Education, 148*, 103798.

Tucker-Raymond, E., Puttick, G., Cassidy, M., Harteveld, C., & Troiano, G. M. (2019). "I broke your game!": Critique among middle schoolers designing computer games about climate change. *International Journal of STEM Education, 6*, 41. https://doi.org/10.1186/s40594-019-0194-z.

U.S. Congress. (2015). *Public Law 114–59: STEM Education Act of 2015*. Washington D.C. Available at https://www.congress.gov/114/plaws/publ59/PLAW-114publ59.pdf Accessed on February 20, 2020.

U.S. Department of Education. (n.d.). *Science, Technology, Engineering, and Math, including Computer Science*. https://www.ed.gov/stem.

U.S. National Science Foundation. (n.d.). *CS for All*. https://www.nsf.gov/news/special_reports/csed/csforall.jsp.

Weintrop, D., & Wilensky, U. (2017). Comparing block-based and text-based programming in high school computer science classrooms. *ACM Transactions on Computing Education, 18*(1), Article 3.

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*, 127–147.

Werner, L., Denner, J., Campe, S., & Kawamoto, D. C. (2012). The Fairy performance assessment: Measuring computational thinking in middle school. In *Proceedings of the 43rd ACM Technical Symposium on Computer Science Education (SIGCSE '12)* (pp. 215–220). New York: ACM.

White House. (2017). *President Trump signs memorandum for STEM education funding*. https://www.whitehouse.gov/articles/president-trump-signs-memorandum-stem-education-funding/. Accessed on February 20, 2020.

Wilson, C., Sudol, L. A., Stephenson, C., & Stehlik, M. (2010). *Running on empty: The failure to teach K-12 computer science in the digital age*. New York: The Association for Computing Machinery and the Computer Science Teachers Association.

Wing, J. M. (2006). Computational thinking. *Communications of the ACM, 49*(3), 33–36.

Yadav, A., Krist, C., Good, J., & Caeli, E. N. (2018). Computational thinking in elementary classrooms: Measuring teacher understanding of computational ideas for teaching science. *Computer Science Education, 28*(4), 371–400.

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

Yadav, A., Stephenson, C., & Hong, H. (2017). Computational thinking for teacher education. *Communications of the ACM, 80*(4), 55–62.

Zapata-Cáceres, M., Martín-Barroso, E., & Román-González, M. (2020). Computational thinking test for beginners: Design and content validation. In 2020 IEEE Global Engineering Education Conference (EDUCON) (pp. 1905-1914). IEEE.

Affiliations

Yeping Li[1] · Alan H. Schoenfeld[2] · Andrea A. diSessa[2] · Arthur C. Graesser[3] · Lisa C. Benson[4] · Lyn D. English[5] · Richard A. Duschl[6]

Alan H. Schoenfeld
alans@berkeley.edu

Andrea A. diSessa
disessa@berkeley.edu

Arthur C. Graesser
graesser@memphis.edu

Lisa C. Benson
lbenson@clemson.edu

Lyn D. English
l.english@qut.edu.au

Richard A. Duschl
rduschl@smu.edu

1 Texas A&M University, College Station, TX, USA

2 University of California-Berkeley, Berkeley, CA, USA

3 University of Memphis, Memphis, TN, USA

4 Clemson University, Clemson, SC, USA

5 Queensland University of Technology, Brisbane, Australia

6 Southern Methodist University, Dallas, TX, USA