Comparing the integration of programming and computational thinking into Danish and Swedish elementary mathematics curriculum resources

Raimundo Elicer[1], Andreas Lindenskov Tamborg[1], Kajsa Bråting[2] and Cecilia Kilhamn[3]

[1] University of Copenhagen, Denmark
[2] Uppsala University, Sweden
[3] University of Gothenburg, Sweden

Computational thinking has become part of the mathematics curriculum in several countries. This has led recently available teaching resources to explicitly integrate computational thinking (CT). In this paper, we investigate and compare how curriculum resources developed in Denmark — digital teaching modules — and Sweden — printed mathematics textbooks — have incorporated CT in mathematics for grades 1–6 (age 7–12). Specifically, we identify and compare the CT and mathematical concepts, actions, and combinations in tasks within these resources. Our analysis reveals that Danish tasks are oriented toward CT concepts related to data, actions related to programming, and mathematical concepts within statistics. This is different from Swedish tasks, which are oriented toward CT concepts related to instructions and commands, actions related to following stepwise procedures, and mathematical concepts related to patterns. Moreover, what is most dominant in one country is almost or completely absent in the other. We conclude the paper by contrasting these two approaches with existing knowledge on computational thinking in school mathematics.

ARTICLE DETAILS

LUMAT Special Issue
Vol 11 No 3 (2023), 77–102

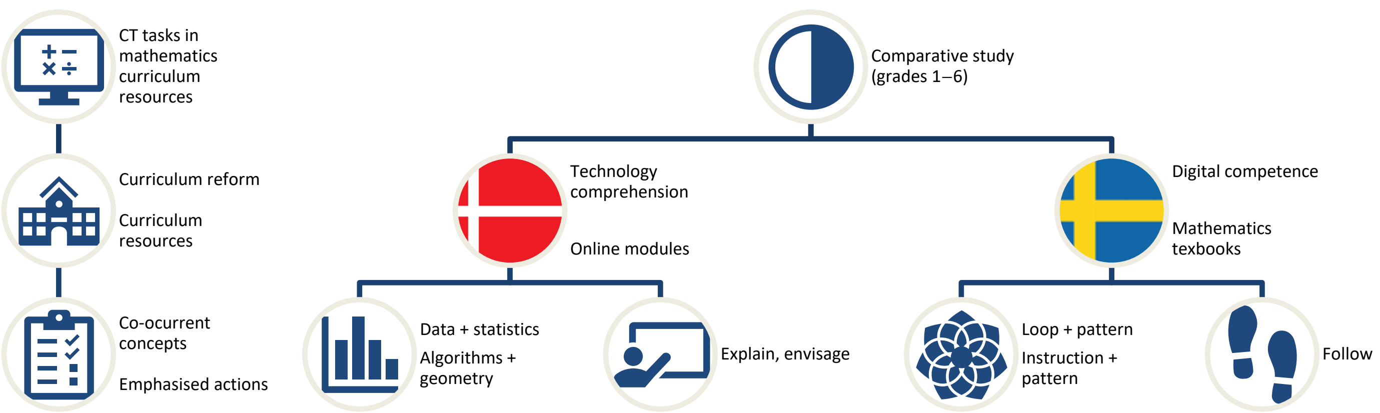
Received 15 February 2023
Accepted 13 November 2023
Published 24 November 2023

Pages: 26
References: 34

Correspondence:
reli@ind.ku.dk

https://doi.org/10.31129/LUMAT.11.3.1940

Keywords: computational thinking, curriculum resources, mathematics, programming



1 Introduction

Computational thinking (CT) is a term coined in the scientific literature and first introduced by Papert (1980) as an educational construct. It roughly refers to a set of knowledge and skills necessary to frame problems in such a way a computer can carry out its solution without creating new problems (Wing, 2006). As a developing concept, it has a myriad of connotations that range from an attempted precise definition (Shute et al., 2017) to a collection of practices (Pérez, 2018; Weintrop et al., 2016). Some scholars argue for CT being more of a way of thinking, rather than being related to computing (Li et al., 2020), a conceptual foundation demonstrable 'with or without the assistance of computers' (Shute et al., 2017) and thus denoting programming as a separate skill. In general, CT is a highly ambiguous term. Palts and Pedaste (2020), for example, recently identified 65 distinct definitions of the term. In this article, we regard computational thinking as a set of constructs empirically observable in human productions particularly in school curricula[1]. CT constructs include programming knowledge and skills, data-handling practices, computational problem solving, modelling, algorithms and simulations. Rather than being guided by a specific definition of CT, this paper takes an outset as what is being referred to as such in Danish and Swedish curricula, which we unfold in the following subsections. It is worth noting that although Wing (2006) emphasized that computational thinking cannot be reduced to programming, many definitions include programming as a sub-element (Bocconi et al., 2022). Therefore, we consider programming to be sub-component of computational thinking.

Since Wing (2006) revived the concept, CT has come to be seen as a teachable competence beyond the domain of computer science, and curricula in many countries have expanded to include elements of CT (Bocconi et al., 2022). However, there are tremendous differences in the implementation strategies adopted by countries, and while some have established new subjects to address CT, e.g., Computing in England (Department for Education, 2013), other countries have revised existing subjects to include elements of CT, such as Sweden and France (Modeste, 2018). In the case of the latter, the CT elements have often been included in the mathematics curriculum.

[1] In this article, following Remillard (2005), we refer to the term curriculum as *formal* curriculum, namely the 'goals and activities outlined by school policies or designed in textbooks' (p. 213), distinct from the curriculum *intended* by teachers' aims and *enacted* in actual classroom practice.

The question of how to establish and exploit synergies between CT and mathematics education is a topic that has been subject to much research and unresolved discussion (see, e.g., Kohen-Vacs et al., 2020). An increasing amount of literature has proposed ways of framing how relationships between CT and mathematics are or should be (e.g., Benton et al., 2017; Gadanidis, 2017). This need for a better understanding has become increasingly important because a growing number of mathematics curriculum resources that include CT have begun to emerge. Yet, more attention has been given to developing ways in which mathematics and CT *could* or *should* be combined than to studying how they *in fact* are combined in available teaching resources (Tamborg et al., 2023). This study focuses on the Danish and Swedish cases, which are of particular interest as they are two neighbouring countries with shared traditions on the aims and approaches to school mathematics (Dahl & Stedøy, 2004). Yet, they have chosen two different ways of implementing CT on a policy level (Helenius & Misfeldt, 2021), whereby CT and mathematics competence descriptions and learning goals tend to be juxtaposed in the Danish case and integrated in the Swedish case (Tamborg et al., 2023). However, it is well known that curriculum policy does not correspond 1:1 to textbook material (Bråting & Kilhamn, 2022). The aim and contribution of this study is to investigate how CT and mathematics are represented and combined in textbook material in Denmark and Sweden and to engage in a discussion of the possible implications of such differences and/or similarities. We conduct this analysis as a comparative quantitative study of curriculum resources available in Denmark and Sweden by investigating what characterises them and how they differ.

We conduct this characterisation building from the suggested computational concepts (know-what) and practices (know-how) by Brennan and Resnick (2012). Although we expect the presence of CT and mathematical concepts in the curriculum resources, we aimed at pinpointing how they are represented and combined. Furthermore, based on Benton et al.'s (2017) framework for actions to embed CT in mathematics, we sought to identify the types of activities students should engage with. Thus, we operationalise our aim by addressing the following research question: *What are the combinations of CT and mathematical concepts and actions involved in Danish and Swedish curriculum resources, and how do they differ?*

We find this research question important and timely because CT is still a new component of mathematics education, which has become mandatory in many places and few teachers are trained in. Curriculum resources are likely to play a pivotal role

in CT teaching in the mathematics classroom. Thus, mathematics teachers' experience of teaching CT is likely to be heavily rooted in the teaching resources that are available (Børne- og undervisnings ministeriet [BUVM], 2021b).

Since Denmark and Sweden are taking different strategies, we begin by briefly describing how CT is related to their mathematics curriculum and the type of curriculum resources that are available.

1.1 Programming as part of the mathematics curriculum in Sweden

In the work leading up to the revised K–9 national curriculum implemented in 2018, the Swedish National Agency of Education undertook the task of strengthening what was referred to as students' digital competence, described as an overarching competence area with no fixed content (Olofsson et al., 2021). The idea was that the content of the national curriculum should be successively renegotiated to include the digital competences that were relevant in the surrounding society. Consequently, this led to revisions of all major syllabi, in which the responsible use of digital media and its social, ethical, and legal aspects was categorised within social science and controlling objects by means of programming became part of the subject technology, while learning programming as such was integrated into mathematics throughout all grade levels (Heintz et al., 2017). In the mathematics syllabus, described in the national curriculum, programming was incorporated under the core content of algebra at all grade levels, described in the following ways[2] (Swedish National Agency of Education, 2018, pp. 56–59): for grades 1–3 (age 7–9): 'How unambiguous, step-by-step instructions can be constructed, described, and followed as a basis for programming. The use of symbols in step-by-step instructions.' For grades 4–6 (age 10–12): 'How algorithms can be created and used in programming. Programming in visual programming environments.' In grades 7–9 text-based programming is also added.

Helenius and Misfeldt (2021) emphasise that programming itself is in focus in the Swedish syllabi, and that the curriculum does not describe how programming can be used as a mathematical tool. Another characteristic of programming in the Swedish curriculum is that it primarily specifies a number of practices that students should be able to perform, whereas programming concepts are more or less absent.

In Sweden, teaching resources are primarily produced by private publishers, not by the National Agency of Education. Since 2018, several textbook producers have

2 This is the official English translation.

made efforts to include CT by developing teaching resources that address programming in mathematics. For grades 1–6, publishers have mainly done so by incorporating CT into the ordinary mathematics textbooks (Bråting & Kilhamn, 2022).

1.2 Technology comprehension in the Danish mathematics curriculum

Denmark has not yet made the final decision on revising the curriculum to include CT. In 2018, BUVM, however, launched a pilot project in which 46 schools across the country were to implement a new subject called Technology Comprehension (BUVM, 2018). The Danish approach sought to gain initial experiences with two different models of implementing technology comprehension to systematically research the effects of these approaches and, ultimately, inform a future, national-scale curriculum revision. The two strategies were 1) technology comprehension as a subject and 2) technology comprehension as an integrated part of existing subjects, such as Danish, mathematics, social sciences, science, physics/chemistry, craft and design, and the arts (BUVM, 2018). Both implementation strategies began with developing a curriculum for a subject in its own right. The individual components of this curriculum were then distributed among the subjects into which it should be integrated. The purpose declaration of technology comprehension emphasised a critical mindset and democratic values, reading 'students shall develop competencies and obtain skills and knowledge so that they constructively and critically can participate in the development of digital artefacts and understand their importance' (BUVM, 2019, authors' translation). This broad focus was also reflected in the four competency areas described: *Digital empowerment*, the critical exploration and analysis of how technology shapes our lives; *Digital design and digital design processes*, framing problems and generating solutions through iterative processes, which can lead to prototypes; *Computational thinking*, the ability to develop solutions to complex problems, the ability to make abstractions regarding phenomena and relationships, and computers' ability to process information; and *Technological agency*, the ability to understand and use digital technology to develop digital artefacts.

The two countries typify and classify aspects that can be considered part of CT differently. In Sweden, programming is the explicit manifestation of digital competence in the mathematics curriculum. In Denmark, CT is considered a competence area, while programming is left as a skill within technological agency. Therefore, we use CT as an inclusive term that gives justice to what programming entails in the Swedish curriculum and makes an explicit addition of programming into CT in Denmark.

In the following section, we briefly outline existing research on CT and mathematics to position the contribution of this paper in relation to the described body of knowledge.

2 Perspectives on relationships between mathematics and CT

During the past decade or so, an increasing body of knowledge on the relationship between CT and mathematics has emerged. This research follows several strands. One perspective has focused on unfolding the theoretical differences between mathematics and CT from an educational perspective and, in addition, developing arguments regarding how they could and why they should be connected. Along these lines, Pérez (2018) has argued that an essential difference between mathematics and CT is their orientations. In his view, mathematics tends to be inwards-oriented in the sense of being abstract and predominantly focusing on understanding disciplinary concepts and terms. In Pérez's (2018) view, CT is, on the contrary, more outward-oriented in that methods, concepts, and ways of thinking always are taught, learned, and applied in relation to practical problems in the real world. He argues that one powerful potential of integrating CT and mathematics is that it allows mathematics education to appeal to a broader and more diverse group of students, without favouring mathematics-advantaged students.

Gadanidis (2017) makes a similar proposition by arguing that, while mathematics and CT share a focus on logical structures and modelling, they operate within distinct epistemological frames. In his view, the frame of mathematics is associated with being a mathematician and engaging in mathematical practices, while the frame of computational thinking emphasises productive actions and their role in task optimisation. Gadanidis (2017) argues that CT can support mathematics education by, among other things, increasing students' agency, supporting abstraction, and enabling automation. Research however also found the overlap between mathematical and computing languages overlap to be a potential source of confusion. For example, Bråting and Kilhamn (2021) showed that symbols from the two domains can carry different meanings (e.g., the equal sign) and that different symbols carry the exact same meaning (e.g., modular arithmetic).

While these contributions indicate both good reasons to integrate CT into mathematics education and the potential pitfalls, they offer only a little advice regarding how to achieve such integrations. Weintrop and colleagues (2016) took this a step fur-

ther and developed a taxonomy of four computational thinking practices for mathematics and science teaching. These practices are classified into data, modelling and simulation, computational problem-solving, and systems thinking (Weintrop et al., 2016). Each of these practices is described in terms of their taxonomical levels, from their most basic components to more advanced ones.

The work described above primarily develops arguments for why and how CT and mathematics *should* and *could* be combined. Fewer studies have conducted empirical descriptive work that investigates and compares what *in fact* is done in curriculum resources from different contexts, here among our recent work. Bråting and Kilhamn (2022) developed an analytical tool to analyse Swedish textbook tasks, adapting Brennan and Resnick's (2012) and Benton et al.'s (2017) frameworks for action. Analysing CT tasks designed for the Danish mathematics curriculum, Elicer and Tamborg (2022) took a grounded approach without any *a priori* defined categories by means of open, comparative, and iterative coding. A corollary to this study is that the categories can be approximated by identifying whether a task includes CT or mathematical concepts and actions and combinations thereof. We build on this work to further characterise and compare these teaching resources at a more systematic level.

3 Methodology

In what follows, we describe and justify the analytical strategy we will use to address the research question. Because the study takes available curriculum resources for elementary school (grades 1–6) as a point of departure, we first describe the sources of data and the selection process, leading up to specifying the unit of analysis. Second, we describe and argue for the analytical tool and its connections to existing theoretical frameworks. Finally, we display the strategies used to process and summarise the data analysis in light of the research question.

3.1 Data sources and selection of tasks

In Denmark, the subject Technology Comprehension has not yet become part of the mandatory curriculum. However, as part of the pilot project, expert groups developed a series of teaching modules oriented toward each of the two strategies for tech-

nology comprehension (as a subject in its own right and integrated into existing subjects), which are publicly available.[3] These modules include a declaration of competency areas and learning goals, an overarching scenario and problem statement, a sequence of tasks and resource banks. These resources were initially developed in 2019, and some of them were updated in 2021 as a result of the pilot project (BUVM, 2021b). We take these materials to be included in our analysis, since the resources constitute the only teaching materials that integrate technology comprehension in mathematics in compliance with the pilot curriculum.

In principle, there are 18 teaching modules developed for mathematics in grades 1 to 6, but four of them only cover technology comprehension areas related to digital empowerment, digital design and design processes, and sub-areas of technological agency outside of programming (Elicer & Tamborg, 2022). Therefore, a total of 14 teaching units include CT learning goals embedded into mathematics.

The modules are designed to be approached during several lessons, and they follow a general structure consisting of three phases: introduction, challenge and construction, and outro-phase (BUVM, 2021a). In turn, these phases are subdivided into tasks signposted with numerals and headings, sometimes subdivided into parts interpreted as separate tasks. We took these tasks as units of analysis in order to have fair ground of comparison to the Swedish curriculum material, which are mainly organized in smaller tasks. A total of 165 tasks were analysed in the Danish material.

In Sweden, since CT is an integral part of the mathematics curriculum, a fairly high volume of digital and printed resources are already available. In this study, we have chosen to restrict our analysis of the Swedish resources to printed mathematics textbooks because the programming content is then included in a well-known mathematical context and can be expected to be in use for a longer time as compared to digital resources, which are revised more often. We screened 56 Swedish mathematics textbooks for grades 1–6, of which 33 did not contain any tasks explicitly or implicitly labelled as belonging to CT (Bråting & Kilhamn, 2022). The resulting 23 textbooks included CT as chapters or sections titled ‘programming’ and ‘programming and pattern’, amounting to a total of 390 tasks, which are treated as units of analysis. The books belong to the following four series:

3 www.tekforsøget.dk/forlob

- Favorit matematik, grades 1–6, published by Studentlitteratur.
- Mondo matematik, grades 1–6, published by Gleerups.
- Singma matematik, grades 1–5, published by Natur & Kultur.
- Prima matematik, grades 1–6, published by Gleerups.

We explained the relevance and timeliness of comparing the curriculum resources from Denmark and Sweden. We acknowledge that the status of the curricular reforms and genre of their resources differ. However, the selection of tasks deals with these issues to a large extent. For both countries, the sources of data include relatively stable CT curriculum resources explicitly embedded in a mathematical context. All tasks are explicitly or implicitly related to CT, either by including CT concepts and actions or by appearing under a heading that relates them to CT. They are developed for grades 1–6 and contain a handleable number of signposted tasks, available for teachers to make use of in class without further instruction. Thus, we deem our data reasonable for comparison.

3.2 Theoretical underpinnings and analytical tool

Addressing our research question requires identifying CT and mathematical content in tasks that are included in Danish and Swedish mathematics curriculum resources, as well as how CT is combined with mathematics in these two contexts. In order to achieve this, we must identify the domain-specific aspects in the units of analysis. Based on the description of CT practices and concepts in the work of Brennan and Resnick (2012) and design principles for programming activities in mathematics developed by Benton et al. (2017), Bråting and Kilhamn (2022) constructed a framework suitable for analysing textbook tasks. In the following, we describe how we have applied these theoretical underpinnings to support the identification of the domain-specific *actions* and *concepts* involved in the selected tasks.

3.2.1 Concepts

In order to characterize CT resources for mathematics, we need to identify aspects of CT and aspects of mathematics that came to the fore in the tasks. Our analysis is partly based on identifying and distinguishing between two types of concepts, *CT concepts* and *mathematical concepts*. All of these are identified via the explicit and meaningful use of words in the context tasks. That is, we identify the occurrence of a signifier in the form of an explicit term, with the occurrence of the signified concept. Although a

concept is much more than the words used to represent it (Wedman, 2020), in an analysis of written text, it is only through words, symbols and images that the intended concepts are available.

CT concepts are those that do not pertain to school mathematics in its traditional sense and belong to the task with the purpose of exploring and learning a computational idea (Li et al., 2020). Aside from programming concepts — e.g., code, algorithm, and condition — we consider concepts from CT, namely those related to (computer) modelling, data practices, and structures. This first step in identifying CT concepts is inspired by Brennan and Resnick's (2012) framework for CT, whose first dimension consists of a closed set of computational concepts: sequences, loops, parallelism, events, conditionals, operators, and data. However, while these predefinitions of CT concepts are helpful, they are not necessarily exhaustive. In our analysis, we follow a grounded approach in which the concepts described above serve as an important source of inspiration. The Danish teaching resources include a list of technological disciplinary concepts. For example, the task in Figure 1 comes from the module 'Concept of chance', which declares two competency areas from CT, namely programming and user studies and redesign. Therefore, *data* is not a CT concept in this context but, rather, a mathematical one. In Swedish textbooks and syllabus, computational concepts are referred to as programming concepts. For example, the concept *code* in the task displayed in Figure 2 is unambiguously a CT concept.

Mathematical concepts are those traditionally belonging to school mathematics, with an emphasis on the mathematical ideas to be learned in the particular context, including the mathematics sub-area and learning goals. For example, Figure 1 displays a Danish task that instructs students to play a dice game in pairs, register the results and winner of each play, and discuss whether the game is fair. Here, the mathematical (statistical) concepts highlighted are *fairness* and *data*. In the Swedish task shown in Figure 2, geometrical figures are represented in order to be identified. The one mathematical concept involved is *pattern*.

3.2.3. Subject loop

Students must play the game: "The difference between two dice". The game involves rolling two dice. The difference is determined and noted in a table. See *Table for Rolling Two Dice (Worksheet A1)*. Player A wins if the Difference is 2 or 4.

Students must gain experience with the game by playing together two by two with ordinary dice (one blue and one red). A student is defined to be player A and one is defined to be player B.

Dice 1	Dice 2	Difference	A wins
5	1	4	1
3	3	0	0

Figure 3 Table for Rolling Two Dice – Difference

If A wins, 1 is written in the last column. If A loses, 0 is written, so that 1 indicates a positive answer and 0 a negative outcome.

Students get 15 min. to play the game, after which they in the group should discuss whether the game is fair. They have to argue for their answers based on their data.

Students' experiences from the game are gathered in class and reasons why the game is unfair are shared.

Figure 1. Task belonging to the Danish course ‘Concept of chance’

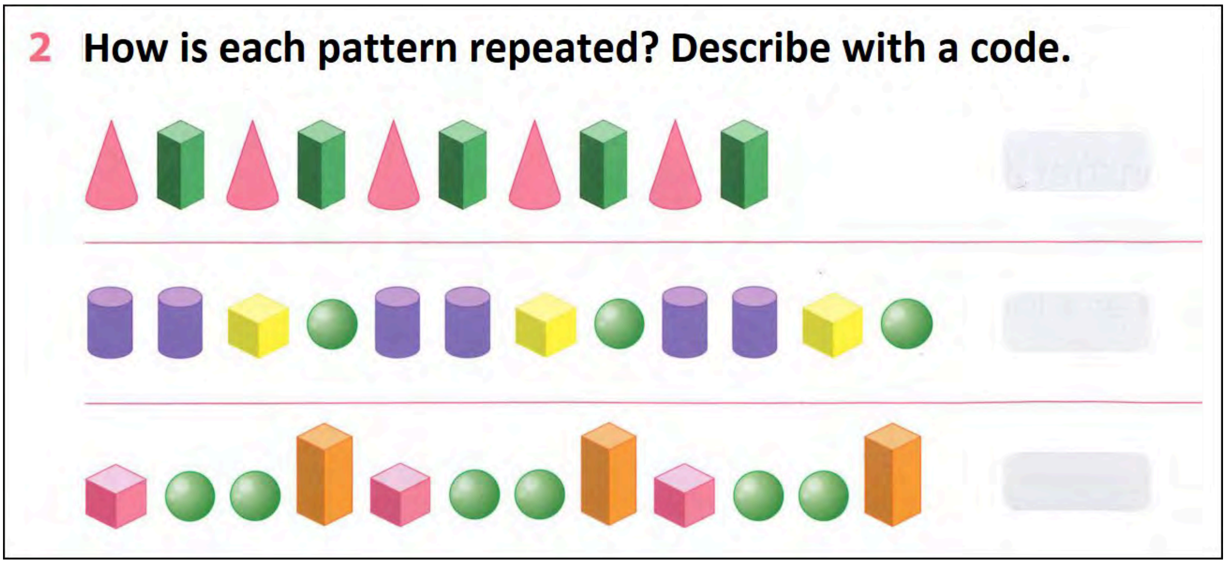


Figure 2. Programming task from Swedish textbook Singma 3B

3.2.2 Actions

In addition to the invoked concepts, we classify the CT-related activities students are explicit asked or expected to engage in when doing the task as *actions*. Though Brennan and Resnick (2012) defined *computational practices* as their second dimension in framing CT, these practices do not highlight CT’s relationship to mathematical learning. As part of the *ScratchMaths* project, to answer the question of what programming in Scratch can do for students’ mathematical learning, Benton et al. (2017) delineated five such activities. These five activities formed what they refer to as the 5E pedagogical framework. One caveat regarding this framework is that it was developed

to *design* programming activities in Scratch, as opposed to *analysing* tasks at face value. In other words, it is a framework for action, which includes ‘prescriptions for pedagogical strategies’ (diSessa & Cobb, 2004, p. 81) and can provide effective heuristics for designing and teaching. We have adapted the action framework of Benton et al. (2017) into six actions suitable for analysing the Swedish data. Detailed examples of each action can be found in Bråting and Kilhamn (2022). The actions are:

a) Follow, i.e., follow a procedure, follow stepwise instructions, or repeat or continue a pattern.
b) Figure out, i.e., work out a procedure, a rule, or a pattern
c) Debug, i.e., find mistakes in a pattern or debug a code
d) Program, i.e., form and create, give instructions, create a pattern, write code, or represent with symbols.
e) Explain, i.e., using words/natural language to explain or describe a procedure, a rule, a pattern, or a concept.
f) Envisage, i.e., predict what will happen or reflect on potential outcomes when conditions or values are changed.

For example, in Figure 1, students should *follow* (a) instructions to play a game and register its outcomes. Although the task includes a discussion and plenum about the game’s fairness based on its results, it is not the instructions for the game that must be figured out (b) or explained (e). The task in Figure 2 is coded as *figure out* (b) because students should work out the pattern that the figures follow. Although the concept of *code* is present, the task only asks them to describe the pattern with a code, not to create an original program or pattern (d).

The analytical tool described above could be seen as a compromise between the open, face-value coding of concepts and predefined practice-oriented categories. It has previously been successfully used to analyse Swedish textbooks (Bråting & Kilhamn, 2022). For comparison’s sake, this is the analytical tool of choice to provide an initial characterisation and overview of tasks in the Swedish and Danish resources. However, addressing our research question requires us to further process these findings in the analysis.

3.3 Data processing and analysis

As stated, our research question focuses on investigating the characteristics of the concepts and actions included in the Danish and Swedish tasks, as well as how these

were combined with mathematical content in these two contexts. In order to address this question, we began by coding the tasks to identify what concepts from mathematics and CT the tasks address and what types of programming actions they include. Our coding of concepts from mathematics and CT was based on the concepts explicitly mentioned in the tasks (e.g., triangle, area, addition, etc. and algorithm, loop, and bug). Next, we coded each task according to what type of actions it included. We conducted all coding manually in a spreadsheet. For each country's case, two researchers coded the tasks — concepts and actions — following three stages: 1) joint coding, 2) parallel coding with a later settling of eventual disagreements, and 3) the separate coding of the remainder. We remained in Stages 1 and 2 for approximately one-third of the total data for each country. This process left us with a descriptive account of the CT and mathematical content addressed in the tasks. After coding the material, we conducted four aggregations or summaries, which we report in the results section.

The first two aggregations result after counting the number of tasks that contain CT and mathematical concepts for each country's material. For comparison's sake, we report the percentage relative to the total number of tasks from each country; 165 from the Danish material and 390 from the Swedish material. The most frequent CT concepts are reported and compared in Table 1. Mathematical concepts are more diverse, and thus, we classified them into arithmetical (operations, number systems), geometrical (polygons, coordinate plane, angles) and statistical (probability, data) concepts. Given the strong emphasis on patterns and number sequences in Sweden, we designated these as a separate category. Mathematical concepts are reported in Table 2.

The next two data processing steps address combinations. First, we compare the co-occurrence of CT and mathematical concepts. For this, we count the number of tasks containing combinations of the CT and mathematical concepts reported in Tables 2 and 3. For example, the task in Figure 1 only refers to data in its mathematical (statistical) sense, namely as registered instances of a random process. Therefore, that task does not report such a co-occurrence. However, the task in Figure 2 includes a mathematical and a CT concept, *pattern* and *code*, respectively, so it is counted as a co-occurrence. This summary and comparison are reported in Table 3.

Finally, we compare the distribution of actions involved in tasks that include mathematical concepts. After filtering out those tasks that do not include mathematical concepts, we count the number of tasks, for each country, that involve each of the six actions of our analytical tool. These are reported in Figure 3 as percentages, for the sake of a fair comparison. It is important to note that tasks may include more than

one CT or mathematical concept, as well as several actions. For this reason, the total does not always add up to 100%. We will return to this matter in the discussion. Below, we describe the results of our analysis described above.

4 Results

4.1 CT concepts

As described above, the first analysis concerns the number of tasks containing different CT concepts. Table 1 below represents the results of this and the relative distribution of CT concepts in the resources from the two countries.

Table 1. Overview of CT concepts and their distribution in Danish and Swedish tasks, absolute numbers and relative percentages.

	Danish tasks		**Swedish tasks**	
Instruction, command	12	7%	189	48%
Algorithm	19	12%	74	19%
Loop, iteration, repetition	0	0%	101	26%
Rule	1	1%	51	13%
Code	14	8%	59	16%
Condition	0	0%	37	9%
Bug, debugging	1	1%	17	4%
Data	20	12%	2	1%
Program, programming	33	20%	10	3%
Totals	**165**		**390**	

In Table 1, we see that the CT concepts of programming, algorithm, and data are the most frequent in the Danish tasks. The least represented CT concepts in Danish tasks are loop/iteration/repetition and condition, none of which are present at all. Rule and bug/debugging each appear in 1% of the tasks. As the reader may notice, the sum of the percentages in the Danish column does not add up to 100. This is because the percentages are computed against the total number of tasks, some of which may include more than one CT concept. However, 63 (38%) Danish tasks do not include CT concepts at all.

In contrast, only 34 (9%) of the Swedish tasks analysed include no CT concepts. In these materials, the most frequently occurring CT concepts are instruction/command, loop/iteration/repetition, and code. Instruction/command stands out by appearing in nearly half of the Swedish tasks. Approximately one quarter of the Swedish tasks include the concept of loop/iteration/repetition (the three terms are in here seen as

representing the same concept). The least represented CT concepts are data, program/programming, and bug/debugging. The low representation of data stands out when compared to the Danish case, where it was the second most frequent concept. Moreover, program/programming is the second least represented CT concept in Sweden, while in Denmark, it was the most frequent CT concept.

4.2 Mathematical concepts

The second analysis concerns the mathematics concepts in the tasks. The result of this analysis is summarised in Table 2, which also shows the relative distribution of the mathematical concepts.

Table 2. Overview of mathematical concepts and their distribution in Danish and Swedish tasks in absolute numbers and relative percentages.

	Danish tasks		**Swedish tasks**	
Pattern, sequence	0	0%	131	34%
Geometrical concepts	56	34%	61	16%
Arithmetical concepts	17	10%	26	7%
Statistical concepts	35	21%	2	1%
Totals	**165**		**390**	

In Table 2, we see that patterns and sequences by far are the most frequently occurring mathematical concepts in the Swedish tasks and that geometrical concepts are the second most frequent.

Regarding the Danish tasks, Table 2 shows that geometrical concepts are the most frequent and that statistical concepts are the second most frequent. Patterns and sequences are absent, although it could be argued that several of the Danish tasks address patterns implicitly. One example of this is the task entitled ‘Design the class’s new clock’, in which students are to develop pattern-like figures in GeoGebra, which they can use as the background for a watch. The tasks, however, focus on design processes and do not deal with patterns in the mathematical sense of the word.

It is worth mentioning that, in both cases, the share of tasks that do not engage with mathematical concepts is similar. 150 (38%) of the analysed Swedish tasks and 71 (43%) of the analysed Danish tasks do not contain any explicit terms referring to mathematical concepts.

4.3 Co-occurrence of concepts

Overall, 44 (27%) of the Danish tasks contain both CT and mathematical concepts, as compared to 195 (50)% of the Swedish tasks. This feature is reflected in the structure and pace of the teaching resources. Several tasks in the Danish teaching modules are meant to focus on a particular concept, either mathematical or CT, leaving the connections between disciplines to a later wrap-up task (Elicer & Tamborg, 2022).

In relative terms, 53% of the Danish tasks that include CT concepts also contain mathematical concepts compared to 63% of the Swedish tasks. In other words, when tasks introduce or draw on CT concepts, they are, to a larger extent in Sweden than in Denmark, combined with mathematical concepts within them.

In what follows, we can see how these co-occurrences appear to be disaggregated by the type of CT and mathematical concepts.

Table 3. Co-occurrence of mathematical and CT concepts in both countries. Percentages are relative to the total number of tasks from each country.

	Pattern, sequence		**Geometrical concepts**		**Arithmetical concepts**		**Statistical concepts**	
Country	DK	SE	DK	SE	DK	SE	DK	SE
Instruction, command	0%	12%	3%	7%	1%	3%	1%	0%
Algorithm	0%	3%	7%	8%	1%	2%	0%	0%
Loop, iteration, repetition	0%	18%	0%	4%	0%	0%	0%	0%
Rule	0%	12%	1%	0%	0%	0%	0%	0%
Code	0%	4%	4%	1%	1%	2%	1%	0%
Condition	0%	0%	0%	2%	0%	0%	0%	1%
Bug, debugging	0%	3%	0%	0%	0%	0%	0%	0%
Data	0%	0%	2%	0%	2%	0%	9%	0%
Program, programming	0%	0%	5%	2%	2%	0%	3%	0%

The mathematical and CT concepts that most often co-occur in Denmark are data + statistics, algorithm + geometry, and program + geometry. The CT concepts of loop, condition, and bug and the mathematical concept of pattern are absent from the Danish tasks. The most frequently co-occurring mathematical and CT concepts in the Swedish data are loop + pattern, instruction + pattern, and rule + pattern. The only CT concept that is absent from the Swedish tasks is data. An observation that stands out

from a comparative perspective is that the most frequent co-occurrence in Denmark (data + statistics) is completely absent in the Swedish material. Likewise, the three most frequent co-occurrences in Sweden are completely absent in the Danish material.

4.4 Actions

Figure 3 below illustrates the distribution of actions in the tasks from the two concerned countries. Each task can include more than one action.

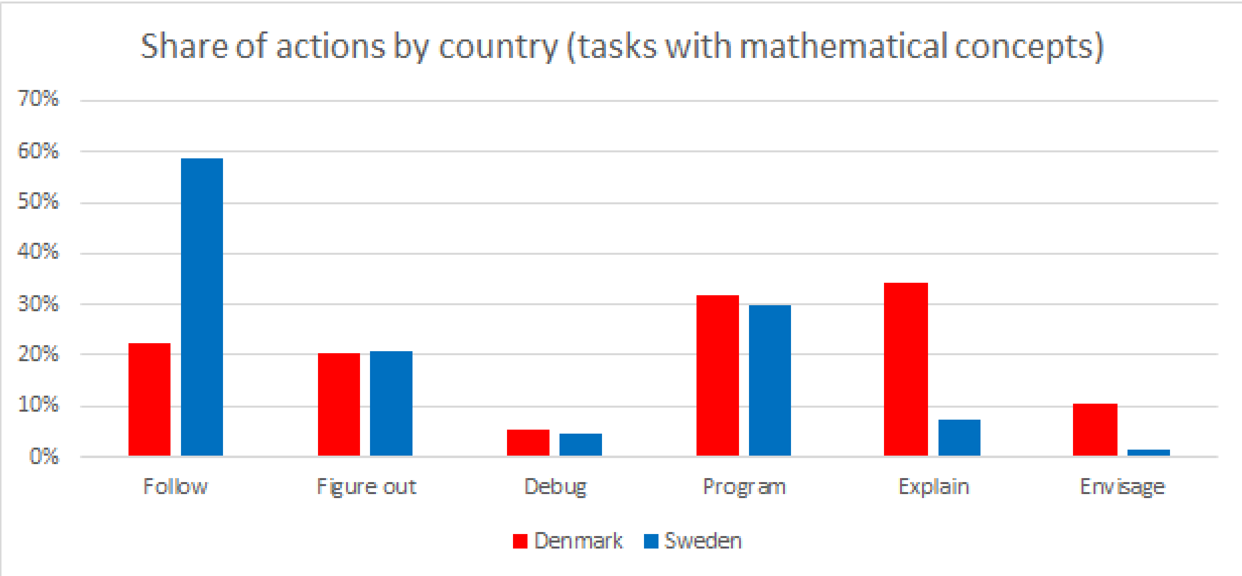


Figure 3. Distribution of the six different actions. Percentages are computed relative to each country's total number of tasks with mathematical concepts: 94 Danish tasks and 250 Swedish tasks.

The first thing to notice from Figure 3 is the difference in the share of tasks that include *following* a procedure and *explaining* it. The follow action is more than three times as frequent in the Swedish tasks as compared to the Danish tasks. With regard to tasks that include the action *explain*, we see that the Danish tasks include this action more than four times as often as the Swedish tasks. The reasons for this could potentially both be found in the format of the Danish tasks and in the content of the Danish technology comprehension curriculum. The template for the Danish teaching modules consistently includes time allocated to 'setting the scene' and 'wrapping up'. Activities in these sections predominantly consist of open questions for students and teachers to address, either in groups or, more often, in plenary classroom discussions. Particularly in the wrap-up sections, students are often asked to present and explain the outcome or process in which they were engaged. The high proportion of the action

explain in the Danish tasks is also consistent with the emphasis given to democratic participation in digital contexts in the curriculum. For example, the goal of the technology comprehension content area modelling specifies that after grade 3, the student must 'be able to describe the reality represented by a model[4].' Addressing this goal is likely to include the use of natural language to describe a procedure, rule, pattern, or concept, which is how the action *explain* is defined in the analytical tool.

5 Discussion

In what follows, we discuss the findings presented in the results section in light of our research question and earlier research within the field.

5.1 Insights on concepts, data sources and analytical tool

Our results reveal that there are notable differences between what CT concepts are included in tasks designed for elementary school mathematics in the two countries and their relative distribution. To some extent, these differences can problematise the choice of data sources and analytical tool.

A significant share (38%) of Danish tasks do not include CT concepts, compared to a 9% of Swedish tasks. Such a sharp difference is consistent with the different way our data sources embedded CT into mathematics. On the one hand, the Danish resources cover combinations of mathematical, CT and other technology comprehension competency areas throughout the modules, which are declared on the front-matter of each material. For this reason, they do not necessarily cover exclusively CT areas nor do so throughout all the tasks in them (BUVM, 2021a). On the other hand, the selected Swedish mathematics textbooks included CT allocated as specific chapters or sections (Bråting & Kilhamn, 2022). It makes sense that a majority of the tasks within them are much more focused on CT.

The absence of mathematical concepts is significant in both cases, respectively 43% of Danish tasks and 38% of Swedish tasks. This feature reflects back on the coding of only explicit terms, as we argued for in section 3.2.1. Although both sources of data come from explicitly mathematics curriculum resources, our decision to code only for explicit terms in the tasks can hide many concepts implicitly involved in their

[4] https://emu.dk/sites/default/files/2019-02/7568_STIL_M%C3%A5l_Matematik_web_FINAL-a.pdf

interpretation and solution. However, the fact that the share of tasks with no mathematical concepts is fairly similar makes the comparisons in sections 4.3 and 4.4 more grounded. The coding of explicit terms may also explain why the CT concept *program/programming* is prominent among Danish tasks in comparison to *code*, while the opposite is true with Swedish tasks (Table 1). One could argue that these terms are interchangeable in one or both contexts, but we deem these terms to represent different concepts. A program (e.g., a function or algorithm) and the act of programming (closer to generic modelling) can be done with different codes and languages (Caeli & Yadav, 2020). More importantly, their synonymity would be an assumption we cannot make strictly based on the data. A limitation of this study's basis for comparison is the fact that, despite being updated after the pilot study, curriculum resources in Denmark are not yet massively used. Therefore, one necessary extension of this work includes analysing Danish textbooks once the reform rolls out, so that the massive use, stable status and number of tasks are more fairly comparable.

5.2 Co-occurrence and the significance of syllabi

We also see notable differences in how actions and CT concepts are combined with mathematical concepts. In the Swedish tasks, actions are highly skewed toward *following* a procedure (Figure 3) while following stepwise *instructions*, along with the concepts of *loop*, *iteration*, and *repetition*, as the most frequent CT concepts (Table 1). These CT concepts and actions are most frequently combined with mathematical content related to *patterns*. The Danish tasks, on the other hand, most frequently include the actions *explain* and *program* (Figure 3) and CT concepts related to programming and data (Table 1), which most frequently co-occur with statistical concepts (Table 3).

To some extent, these characterisations resemble the curricular decisions made reforming the mathematics curriculum to include CT in the two countries. In the Swedish mathematics syllabus described in the national curriculum, following stepwise instructions is explicitly mentioned in grades 1–3, and for grades 4–6, students must be able to program and use algorithms in visual programming environments. This is reflected in our results, in which loop, iteration, and repetition are all essential components of developing algorithms[5]. Based on the high frequency of mathematical concepts related to patterns in the Swedish data, we can speculate that decisions made at

[5] See, for example, https://www.bbc.co.uk/bitesize/guides/zg46tfr/revision/1

the national curricular level influences paths taken at the level of textbook designers and, consequently, perhaps also in the classroom. When CT was incorporated into mathematics in the Swedish curriculum, a choice had to be made. Either a new subject matter area could have been added, which would have given CT the same status as the traditional subject matter areas, or CT could be inserted into one of the existing subject matter areas. For some reason, algebra was pinpointed as the best place for CT in grades 1–6, not statistics or geometry, which are the most frequent content areas in the Danish tasks. Hence, in the Swedish resources, programming became tightly connected to patterns, a topic already present in algebra.

Similarly, we see a relationship between the Danish tasks and the technology comprehension syllabus. This syllabus has a strong emphasis on students' ability to, e.g., critically explore how technology shapes our lives, frame problems, and use digital technology to develop digital artefacts, all of which are integral to the four technology comprehension competency areas (BUVM, 2019). Such curriculum aims seem to align well with tasks in which students are to program, explain what they have accomplished, and combine data and statistics to inquire into societal phenomena from a mathematical and CT perspective.

This alignment between tasks and the mathematics curriculum in the two countries suggests that the curriculum documents are indeed significant for how curriculum resource developers have engaged in the integration of CT. This may seem obvious, but the alignment has significant implications. It points to important limitations of previous theoretical work on potential ways of establishing synergetic relationships between CT and mathematics. The taxonomy of CT practices in the mathematics classroom developed by Weintrop et al. (2016) and Pérez's (2018) arguments for the usefulness of an orientation toward real-world problems in CT for mathematics education are, no doubt, important contributions. They have broadened our understanding of the fundamental differences between mathematics education and CT and provided tools for navigating this new landscape. However, the results of this study remind us that curriculum resources are often developed to comply with curriculum policy, and the implementation of research into curriculum guidelines and resources may be influenced by a diversity of political factors (Aguilar & Castaneda, 2022). In such contexts, theoretical ideas and suggestions regarding synergetic relations between mathematics and CT are likely to be thought of as useful only to the extent that they align with decisions in the mathematics syllabus. Despite the fact that, for instance, data practices can theoretically constitute an obvious boundary object for

mathematics and CT (Weintrop et al., 2016; Gould, 2021), this is of little to no relevance in the Swedish mathematics syllabus for elementary grades where programming is a mandatory part of algebra. If the analysis had been extended to resources for grades 7–9, the results may have been different because the Swedish curriculum does include the 'assessment of risk and chance based on computer simulations and statistical material' in the area of probability and statistics for these grades.

The mathematics curricula for elementary grades in Sweden and Denmark point to two ways in which curriculum policy may constrain the integration of CT into teaching resources, namely content itself and the level of specificity in relation to mathematical content. The two contexts differ in that the CT elements in the Danish mathematics curriculum are rather generic and not content-specific, while in the Swedish mathematics curriculum, CT components are more technically narrowed, leaving other issues to other subjects and educational levels (Helenius & Misfeldt, 2021). This could help explain the perhaps most outstanding result of our study: the most frequently co-occurring CT and mathematical concepts in the Danish resources (data + statistics), completely absent in the Swedish resources, are not as dominant compared to other frequent co-occurrences (Table 3). At the same time, the most frequently co-occurring CT and mathematical concepts in the Swedish resources (instruction/loop/rule + pattern), completely absent in the Danish case, standout drastically from other combinations (Table 3).

5.3 Bridging and resource structure

As our results showed, the available Danish resources tend to have a more even share of tasks involving mathematical concepts (Table 2). However, Swedish textbooks display a larger overall share of tasks that combine CT and mathematical concepts than Danish resources (Table 3). In fact, according to Figure 3, the same can be said regarding actions when mathematical concepts are involved. Although Danish teaching modules always included tasks combining concepts from both domains of knowledge, there were multiple instances in which smaller sub-tasks only included concepts from either mathematics or CT. These dissimilarities between Danish and Swedish tasks in the way they combine CT and mathematics can also be explained in terms of how they are situated in the structure of teaching resources, beyond the data sources we discussed in section 5.1.

At a structural level, the Danish tasks are organised as teaching sequences divided into several smaller steps, slowly progressing toward an end-goal. For example, the

task in Figure 1 — with only mathematical (statistical) concepts — is one of many introductory activities about the notion of chance in the module. Later, the idea of the fairness of a game is connected to the difficulty of a game that students should program and adjust. This is different from the Swedish tasks, which are smaller and more independent. They are not, as such, embedded in the context of a larger inquiry. The task in Figure 2 — with co-occurrent mathematical and CT concepts — is self-sufficient and not at all interdependent with those in the same textbook chapter. In that sense, the results may reflect two distinct approaches in terms of connecting mathematics and CT, either *between* tasks or *within* tasks.

In our use of the term actions, we refer to CT actions in the context of mathematics teaching resources. We have thus not distinguished between mathematical and CT actions. Identifying such connections between domains within a task could be labelled as a CT and mathematical action simultaneously. In their study, Bråting and Kilhamn (2022) acknowledge that the analytical tool does not account for some tasks in which the actions' fields of origin are ambiguous. Benton and colleagues (2017) call this action 'bridging' computational and mathematical ideas. Some Danish tasks also pertain to this overarching category, which Elicer and Tamborg (2022) call 'operational integration'. In a few words, they represent a missing category of tasks in which opportunities to link mathematical and computational concepts become clearer. In a sense, such tasks seem to justify the integration of CT and mathematics by necessitating concepts from both fields within the same action.

6 Conclusion

Our research question aims to characterise the analysed resources by identifying CT and mathematical concepts, combinations thereof and actions involved in the tasks.

In terms of disciplinary concepts, at the K–6 level, resources illustrate a contrast between a notion of CT that focuses mostly on programming and algorithms and another that takes a broader view, one including the handling of data and computer modelling. Danish tasks involve a relatively even distribution of mathematical — arithmetical, geometrical, and statistical — concepts, while Swedish tasks are highly skewed toward patterns and sequences, which are absent in the Danish tasks.

As for combinations of mathematical concepts with CT concepts and actions, what is roughly most available in one is absent in the other. Danish tasks rely on the interplay between data as a CT and a statistical concept in its mathematics curriculum resources. This aspect is in line with the recent trend on developing a common data-

scientific literacy as a merge between statistical, mathematical, and computational thinking (Gould, 2021). In turn, Swedish tasks focus on the combination of stepwise instructions, patterns, and number sequences; a strong focus on programming and early algebra as per the first wave of CT (Clements & Sarama, 1997).

In the introduction, we argued for the need of studies of curriculum resources that not only consider how CT and mathematics *could* and *should* be integrated, but how the two subjects *are actually* integrated in available resources. Given the novelty of CT's integration into mainstream curricula, tasks are what is available for teachers to implement the potential synergies this innovation may bring and, in turn, for students to make use of. On that line, our study provides two main contributions.

First, we have documented that two neighbouring countries with an otherwise shared tradition of mathematics teaching integrate CT into mathematics quite differently; a contrast that permeates its curriculum resources. From a comparative point of view, these empirical findings can inform future discussions about shortcomings and potentials in both approaches when policy decisions are translated into teaching materials. Therefore, our results can be sources of awareness and inspiration in the search for alternative curricular strategies than the ones currently adopted to integrate CT into mathematics.

Second, we constructed an analytical tool to conduct the first contribution building on the state of the art. Despite its acknowledged limitations, it has proven applicable in two highly different contexts with regards to implementation strategies and status, and types of curriculum resources. Considering that the interest in CT is a global trend, we envision our analytical tool as suitable to characterise and position curriculum resources from other contexts in relation to one another. The main focus of this paper has been devoted to curriculum material, thereby leaving out teachers' enactment of the resources in the mathematics classroom. This delineation implies constraints in terms of what this study can offer as insights on the implications for classroom practices. As argued previously, the novelty of CT in K-9 and teachers, limited knowledge of its associated concepts is likely to mean that they will largely lean on curriculum material. However, there remains an empirical question and, as more resources on CT and mathematics will continue to emerge, it is an important direction for future research to study how teachers choose and enact CT curriculum material in the classroom and with what learning outcomes for students.

References

Aguilar, M. S., & Castaneda, A. (2022). Out of the public eye: Researching political factors that influence the implementation of research knowledge as part of educational reforms and mathematics textbooks. *Implementation and Replication Studies in Mathematics Education*, *2*(1), 107–129. https://doi.org/10.1163/26670127-bja10001

Benton, L., Hoyles, C., Kalas, I., & Noss, R. (2017). Bridging primary programming and mathematics: Some findings of design research in England. *Digital Experiences in Mathematics Education*, *3*(2), 115–138. https://doi.org/10.1007/s40751-017-0028-x

Bocconi, S., Chioccariello, A., Kampylis, P., Wastiau, P., Engelhardt, K., Earp, J., Horvath, M., Malagoli, C., Cachia, R., Giannoutsou, N., & Punie, Y. (2022). *Reviewing computational thinking in compulsory education: State of play and practices from computing education* (I. dos Santos, R. Cachia, N. Giannoutsou, & Y. Punie (Eds.)). Publications Office of the European Union. https://doi.org/https://doi.org/10.2760/126955

Bråting, K., & Kilhamn, C. (2021). Exploring the intersection of algebraic and computational thinking. *Mathematical Thinking and Learning*, *23*(2), 170–185. https://doi.org/10.1080/10986065.2020.1779012

Bråting, K., & Kilhamn, C. (2022). The integration of programming in Swedish school mathematics: Investigating elementary mathematics textbooks. *Scandinavian Journal of Educational Research*, *66*(4), 594–609. http://doi.org/10.1080/00313831.2021.1897879

Brennan, K., & Resnick, M. (2012). New frameworks for studying and assessing the development of computational thinking. In *Proceedings of the 2012 Annual Meeting of the American Educational Research Association* (pp. 1–25). American Educational Research Association.

Børne- og Undervisningsministeriet (2018). Handlingsplan for teknologi i undervisningen. https://www.uvm.dk/publikationer/folkeskolen/2018-handlingsplan-for-teknologi-i-undervisningen

Børne og Undervisningsministeriet (2019). *Fælles mål for teknologiforståelse* [Common goals for technology comprehension]. https://emu.dk/sites/default/files/2019-02/GSK.%20F%C3%A6lles%20M%C3%A5l.%20Tilg%C3%A6ngelig.%20Teknologiforst%C3%A5else.pdf

Børne- og Undervisningsministeriet (2021a). *Didaktiske prototyper - Format og vejledning* [Didactical prototypes – Format and guidelines]. https://tekforsøget.dk/wp-content/uploads/2021/06/Format-og-vejledning-til-didaktiske-prototyper-maj-2021.pdf

Børne- og Undervisningsministeriet (2021b). *Forsøg med teknologiforståelse i folkeskolens obligatoriske undervisning: Slutevaluering* [Experiment with technology comprehension in compulsory education: Final evaluation]. https://www.uvm.dk/-/media/filer/uvm/aktuelt/pdf21/okt/211004-slutevaluering-teknologoforstaaelse.pdf

Caeli, E. N., & Yadav, A. (2020). *TechTrends*, *64*(1), 29–36. https://doi.org/10.1007/s11528-019-00410-5

Clements, D. H., & Sarama, J. (1997). Research on Logo: A decade of progress. *Computers in the Schools*, *14*(1–2), 9–46. https://doi.org/10.1300/J025v14n01_02

Dahl, B., & Stedøy, I. M. (2004). A Nordic community: Ideas of education and democracy in mathematics. In I. M. Stedøy (Ed.), Mathematics Education - The Nordic Way (pp. 1–10). TAPIR.

Department for Education. (2013). National curriculum in England: Computing programmes of study. National Curriculum in England. https://www.gov.uk/government/publications/national-curriculum-in-england-computing-programmes-of-study/national-curriculum-in-england-computing-programmes-of-study

diSessa, A. A., & Cobb, P. (2004). Ontological innovation and the role of theory in design experiments. Journal of the Learning Sciences, 13(1), 77–103. https://doi.org/10.1207/s15327809jls1301_4

Elicer, R., & Tamborg, A. L. (2022). Nature of the relations between programming and computational thinking and mathematics in Danish teaching resources. In U. T. Jankvist, R. Elicer, A. Clark-Wilson, H.-G. Weigand, & M. Thomsen (Eds.), *Proceedings of the 15th International Conference on Technology in Mathematics Teaching* (pp. 45–52). Aarhus University.

Gadanidis, G. (2017). Five affordances of computational thinking to support elementary mathematics education. *Journal of Computers in Mathematics and Science Teaching*, *36*(2), 143–151.

Gould, R. (2021). Towards data scientific thinking. *Teaching Statistics*, *43*(S1), S11–S22. https://doi.org/10.1111/test.12267

Heintz, F., Mannila, L., Nordén, L. Å., Parnes, P., & Regnell, B. (2017). Introducing programming and digital competence in Swedish K-9 education. In V. Diagené & A. Hellas (Eds.), *International Conference on Informatics in Schools: Situation, Evolution, and Perspectives* (pp. 117–128). Springer. https://doi.org/10.1007/978-3-319-71483-7_10

Helenius, O., & Misfeldt, M. (2021). Programmeringens väg in i skolan: en jämförelse mellan Danmark och Sverige. [Programming's way into school – a comparison between Denmark and Sweden]. In K. Bråting, C. Kilhamn, & L. Rolandsson (Eds.), *Programmering i skolmatematiken: möjligheter och utmaningar* (pp. 39–56). Studentlitteratur.

Kohen-Vacs, D., Kynigos, C., & Milrad, M. (2020). On the integration of learning mathematics and programming. In S.-C. Kong, H. U. Hoppe, T.-C. Hsu, R.-H. Huang, B.-C. Kuo, R. K.-Y. Li, C.-K. Looi, M. Milrad, J.-L. Shih, K.-F. Sin, K.-S. Song, M. Specht, F. Sullivan, & J. Vahrenhold (Eds.), *Proceedings of International Conference on Computational Thinking Education 2020* (pp. 53–56). The Education University of Hong Kong.

Li, Y., Schoenfeld, A. H., diSessa, A. A., Graesser, A. C., Benson, L. C., English, L. D., & Duschl, R. A. (2020). 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

Modeste, S. (2018). Relations between mathematics and computer science in the French secondary school: A developing curriculum. In Y. Shimizu & R. Vithal (Eds.), *ICMI Study 24, School Mathematics Curriculum Reforms: Challenges, Changes and Opportunities* (pp. 277–284). International Commission on Mathematical Instruction and University of Tsukuba.

Olofsson, A. D., Lindberg, J. O., Young Pedersen, A., Arstorp, A. T., Dalsgaard, C., Einum, E., & Willermark, S. (2021). Digital competence across boundaries - beyond a common Nordic model of the digitalisation of K-12 schools? *Education Inquiry*, *12*(4), 317–328. https://doi.org/10.1080/20004508.2021.1976454

Palts, T., & Pedaste, M. (2020). A model for developing computational thinking skills. *Informatics in Education*, *19*(1), 113–128. https://doi.org/10.15388/infedu.2020.06

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

Pérez, A. (2018). A framework for computational thinking dispositions in mathematics education. *Journal for Research in Mathematics Education*, *49*(4), 424–461. https://doi.org/10.5951/jresematheduc.49.4.0424

Remillard, J. T. (2005). Examining key concepts in research on teachers' use of mathematics curricula. *Review of Educational Research*, *75*(2), 211–246. https://doi.org/10.3102/00346543075002211

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

Swedish National Agency of Education (2018). *Curriculum for the compulsory school, preschool class and school-age educare 2011*. Elanders Sverige AB.

Tamborg, A. L., Elicer, R., Bråting, K., Geraniou, E., Jankvist, U.T., Misfeldt, M. (2023). The politics of computational thinking and programming in mathematics education: Comparing curricula and resources in England, Sweden, and Denmark. In B. Pepin, G. Gueudet, & J. Choppin (Eds.), *Handbook of Digital Resources in Mathematics Education*. Springer. https://doi.org/10.1007/978-3-030-95060-6_55-1

Wedman, L. (2020). *The concept concept in mathematics education: A concept analysis*. [Doctoral thesis, University of Gothenburg]. http://hdl.handle.net/2077/64096

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

Wing, J. M. (2006). Computational thinking. *Communications of the ACM*, *49*(3), 33–35. https://doi.org/10.1145/1118178.1118215