A Pathway into Computational Thinking in Primary Schools

Aleksandra Djurdjevic-Pahl[1], Claus Pahl[2(✉)], Ilenia Fronza[2], and Nabil El Ioini[2]

[1] Independent Researcher and Instructor, Bolzano, Italy
[2] Free University of Bozen-Bolzano, Bolzano, Italy
claus.pahl@unibz.it

Abstract. Computing is a key skill that cannot be underestimated in todays digitalised world. Computing abilities enable humans of all ages and backgrounds to understand, create and manage computerised environments. Consequently, computing education becomes an important concern. For instance, the national curriculum in the UK states that a high-quality computing education equips pupils to use computational thinking and creativity to understand and change the world. Our aim is to support the early stages of computing education in primary schools. Our proposal is a pathway into Computing Education (CE) through Computational Thinking (CT), starting off from traditional mathematics curricula for primary schools. This is a first step, not involving concrete computer programming or ICT management, but develops the key skills of computational thinking such as logical reasoning or abstraction.

Keywords: Computational thinking · Mathematics · Primary schools

1 Introduction

Computing is a key skill that has a significant value in todays digitalised world. Computing abilities enable people of all ages and backgrounds to understand, create and manage computerised environments and the digital content in them. Consequently, computing education becomes a critical concern for our societies. The national curriculum in the UK states that "a high-quality computing education equips pupils to use computational thinking and creativity to understand and change the world"[1]. Computational thinking (CT) brings problem solving, design and understanding together in a way meaningful to computing. CT research can be considered at a formative stage. CT essentially started in 2006 after J. Wing's seminal article [23]. Two adopted definitions of CT are [1, 23]:

- a conceptual framework and a vocabulary for K-12 educators [14],
- an operational definition identifying concepts, practices and perspectives [4].

[1] https://www.gov.uk/government/publications/national-curriculum-in-england-computing-programmes-of-study/.

T.-T. Wu et al. (Eds.): SETE 2016, LNCS 10108, pp. 165–175, 2017.
DOI: 10.1007/978-3-319-52836-6_19

More recently, more practical concerns such as the further development, promotion or assessment of CT are investigated. Still, there is no agreement regarding accepted CT techniques. What is still left unclear is the actual attainment of students in CT-based eduction, i.e., what do we expect students to know after participating in CT education [12] and how to assess this.

According to Wing, CT can be explained as"the thought processes involved in formulating a problem and expressing its solution(s) in such a way that a computer – human or machine – can effectively carry out". Thus, CT does not focus only on problem solving, but also on problem formulation. Our aim is to support the early stages of computing thinking with problem analysis, formulation and solving in primary schools [3, 5, 8, 9, 11, 20, 21]. Our proposal is a pathway into computing education through CT, starting off from traditional mathematics curricula for primary schools. Our approach is a first step that does not involve concrete computer programming or ICT management, but develops the key skills of computational thinking such as logical reasoning or abstraction.

2 Computational Thinking

Computational Thinking (CT) involves solving problems, designing systems, and understanding human behavior, by drawing on concepts fundamental to Computer Science [23]. The Computer Science Teachers Association (CSTA) and the International Society for Technology in Education (ISTE) suggested an operational definition of CT that provides a framework that aims at K-12 educators [14]. In 2012, an operational definition with three key dimensions includes [4].

– concepts (such as sequences, loops, and conditionals),
– practices (such as testing and reusing),
– perspectives (such as questioning and expressing).

Teachers and students should use an appropriate vocabulary to describe problems and solutions [2]. Students should learn to accept wrong solutions as part of a path for a positive result. Moreover, they should be encouraged to work as a team and to use decomposition, abstraction, negotiations, and consensus building. It is also important to stimulate interest in computational concerns.

Recent work on CT focused on tools to support CT [13]. Graphical programming environments are probably the most commonly used solution, since they allow an early experience of design and construction in a computing context, while avoiding syntax problems. Some recent research looks at CT assessment, i.e., how to evaluate the effectiveness of a CT curriculum by measuring what students have effectively learned [17, 22].

Some sample CT strategies and tools to teach and assess CT will now be described. Visual programming languages allow creating programs by manipulating graphical elements; the notation is usually based on diagrams or blocks to be joined between them [6]. Animation programs, for instance, show some aspects of program execution in an explicit form (e.g., the call stack) to illustrate the concept of execution. Moreover, visual simulation of programs allows students to learn how to read code, understand the flow in it and then trace it.

The state-of-the-art in CT research reveals shortcomings that would benefit from further mainly empirical research [13]:

– Grade and age-appropriate curricula for CT still need to be designed or improved [10]. Most of the studies have been conducted in an undergraduate context. Thus, more empirical work in the K-12 context is needed to understand the problems faced during the first computing experiences. We look at primary schools here, which have been neglected.
– Emphasis has been put on coding so far [7]. This has led instructors to focus on a small aspect of CT, thus neglecting its broader aims. We approach the subject from a mathematics baseline, also drawing in a variety of other areas.

While at secondary school level, coding can form part of the curriculum, at primary school level this opportunity is more restricted. While some programming tools like Scratch Junior exist (https://www.scratchjr.org/), more emphasis should be put here on the foundations that coding is based on. This is reflected in the community by an agreed set of CT capabilities. CT aims at capabilities that help with using computers to solve or model problems [1, 24], but can actually be learned without programming. Key capabilities are (see Fig. 1 for a visualisation): *algorithms*: making steps and rules; *logic and logical reasoning*: predicting and analysing; *decomposition*: breaking down into parts; *patterns*: spotting and using similarities; *abstraction*: removing unnecessary detail; *evaluation*: making judgements.

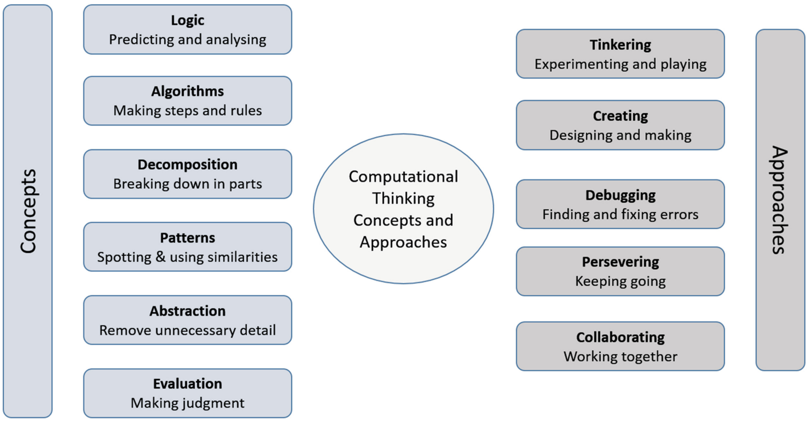

Fig. 1. CT Principles – concepts and approaches.

We show here a pathway into CT that builds these capabilities in a nonprogramming environment. This specifically provides an outline for 4-year curriculum for primary school children from year 1 onwards.

3 Thinking Outside the Box

We describe here a 4-year curriculum for primary school children that is based on our experience of developing and teaching this at a primary school. The material has been taught as a 1-hour per week extra-curricular activity called "Thinking Outside the Box" – a name that is meant to capture the idea of (computational) thinking in the broadest possible sense.

Thinking Outside the Box (TOtB) is a series of classes of mathematics grounded computational thinking activities designed to make pupils think. Generally, the activities are complementary to those being taught in traditional classroom activities and their aim is to help children develop logical thinking, acquire analytical and problem solving skills. The approach is however different.

In traditional primary school education, children often tend to develop a compartmentalized view of mathematics from a very early age. At the same time, they tend to develop a dislike for maths and logic. To avoid something like that happening as much as possible, we propose to use a much wider approach to maths and computational thinking through problem solving based on a dedicated set of materials. The tasks all focus on CT capabilities, but the wider context is in the area of art, language or science as much as is in the area of maths and computing, thus stressing the multi-disciplinary impact and relevance of CT.

A key aim is that children have to deduce themselves in an exploratory and discursive way, often as a group, what their tasks are, rather than being told what to do. For children who are taking part for the second or third year (depending on the class they are in), this is a concept they are familiar with and used to. The type of activities vary greatly depending on the class (year) and its composition. Each class starts from a common point at the beginning of the year and if layering amongst the children occurs, the material is adjusted according to their abilities and needs. While there is a basic, joint level for all of them, but there are also additional tasks if necessary. At the end of the year, they will all finish with a similar skills set. The majority of the work is individual. However, the analysis of the results is a joint effort in which they learn to listen to other students' ideas and reasons and compare them to their own.

4 A 4-Year Curriculum for Primary School CT

The core CT capabilities that are targeted by the suggested primary schools CT curriculum are as follows. We also list the respective activities in thematic areas that address them, cf. Fig. 2:

- abstraction (through arithmetic and spatial coordination)
- logical reasoning (through cryptography, and applied arts, language and nature-based problems)
- patterns (through geometry/symmetry and art)
- decomposition (through spatial coordination)
- algorithms (through cryptography, arithmetic and geometric puzzles)
- evaluation, which is one of the CT capabilities, is a cross-cutting concern.

	Arithmetic	Geometry	Cryptography	Spacial Coordination	Applied Problems
Abstraction	X			X	
Logic			X		X
Patterns		X			
Decomposition				X	
Algorithms	X	X	X		
Evaluation	X	X	X	X	X

Fig. 2. Concerns Mapping: Capabilities (abstraction etc.) addressed through Activities (arithmetic etc.).

While not involving computing per se, the aim is to introduce the children to problems that can be solved through computers and providing them with the core skills to program these computers. Thus, a further key objective is to draw on problems from a range of contexts: arts, languages, nature, maps and spaces.

This aims to bring the idea across that problems from a range of disciplines that can be solved through computational methods. At this early stage of a child's education, this broadness of application domains is more important than the coding/programming that is often seen as the core of CT.

We have associated CT capabilities with subjects suitable to teach them, see Fig. 2. These include standard tools like tangrams or pentaminos. We provide concrete activities around these subjects that involve tools (manipulative in Sect. 4.1) or other forms of exercises (non-manipulative in Sect. 4.2).

4.1 Manipulative Activities

Manipulative work aims to support logical thinking and spatial awareness (including eye-hand coordination). For the latter, objects from one-dimensional to three-dimensional (1D, 2D, 3D) are involved. We now outline the tools we used.

- 1D and 2D shapes: Tangrams are used with additionally created stencils, aiming at the analysis of space. Pentaminos are used, firstly, slotting given shapes into rectangles and, secondly, slotting free chosen shapes into any form. Tantrix™ is a puzzle-based table geometry and strategy game.
- 3D shapes: Soma cubes are used in two ways: firstly, for beginners, to build 3D shapes based on 3D graphical presentations using given set of pieces and, secondly, for advanced learners, to build 3D shape based on 2D graphical presentation using given set of pieces. Katamino™ and 3D Pentamino are used to enhance the 2D skills. This could involve building from 2D plans showing front/back/left/right/top aspects of the model. This can be done using any blocks including interlocking ones.
- LÜK™ is a multi-purpose learning system (from mathematical to language to science and arts) that provides a self-learning approach with selfassessment and correction.

4.2 Non-manipulative Activities

Non-manipulative work is about exploring constitutional parts of CT – pattern, abstractions, decomposition – using, firstly, different application contexts: (i) art (painitngs/drawings by Kandinsky, Klee, Escher, etc.) and (ii) language (looking at the syntax and semantics) Secondly, more traditional technical subjects are used in the following way. (i) Science and humanities: geography for spatial awareness/map reading and coordinate finding; history to reason about time travel; astronomy with problems based on constellations or light speed; biology. (ii) Cryptography: using code crackers to explore the relationship between numbers, letters (or letter shapes and symbols) and ciphers. (iii) Technical drawing: including symmetry work (left, right, split left/right in the same picture (1 line)/up, down (1 line)/under angle (14 lines); drawing from perspective based on an existing drawing; scaling down or up based on existing drawings).

5 Curriculum by Year

We can map the core manipulative and non-manipulative activities onto a 4-year primary school curriculum. The proposal here is meant to complement a normal curriculum, in particular ongoing maths activities. It is aimed to expose pupils to principles of CT towards educating children to become computationally literate and competent, but also demonstrating topics from a multi-disciplinary context that CT is an ingredient of many other non-maths or non-computing subjects. We detail this now for each year (joining similar 3rd and 4th year).

1st Year activities. First year activities are divided into four areas to explore constitutional parts of maths such as pattern, abstractions, decomposition using:

- The **arithmetic** part includes work on different number puzzles and simple code crackers. The aim of both is to make the children feel comfortable with numbers, move them away from horizontal or vertical addition/subtraction, and to put numbers into a wider context, connecting them with language (letters, words) and art (shapes).
- **Logic reasoning** includes work on sequencing, pattern recognition and pattern prediction and covers art, language and nature. This part of the childrens activity includes often work with manipulative puzzles such as pentaminos and tangrams as well as the use of LÜK.
- **Spatial awareness**, basic map reading and coordinate finding.
- **Geometry** consists of one-line symmetry exercises. The idea behind these exercises is to train eye–hand coordination, fine motor movement and the use of rulers as part of a wider geometry and space concern.

2nd Year activities. Second year activities focus on four areas in particular: arithmetic, comprehension, logic reasoning and symmetry, because these are important for a better understanding of problems.

- **Arithmetic** work includes arithmetic puzzles in different forms as often as possible and combines them with logic reasoning and comprehension exercises where the children have to interpret the text and express it through numbers or the other way around. Introduction to different calculation strategies becomes another important aspect of their arithmetic work at this time.
- **Logic reasoning** includes further work on more complex sequencing, pattern recognition and pattern prediction and covers again art, language and nature. The children also continue to work with manipulative puzzles (pentaminos and tangrams) as well as LÜK.
- **Symmetry**. Image reproduction work consists from two parts: one-line symmetry exercises (which includes left-right, up-down and split (one-line) symmetry) and finishing partially fragmented/degraded image when only a rough outline is given (connecting dots to make an image) based on a smaller version of the original shown in the picture. The aim of these exercises was to work on their concentration, analysis of space and precision. This part of the childrens' work allows us to understand better how they see things around them and how they process the acquired visual information. In a group setting, this aids negotiation and consensus building.

3rd Year and 4th Year. As children in 3rd and 4th year are more advanced, their curricula differ less. The work with third and fourth year is done through a series of problem solving questions as it is a flexible way to explore a variety of topics. Using this platform, the classes explored numbers, patterns, space and change, but again in a wide context of language and art. In addition to this, the children were introduced to cryptography. This aims at exploring pattern, abstractions, decomposition:

- **Logic reasoning** includes further work with tangrams, pentaminos. The children are also introduced to the soma cube. Tasks include building 3D shapes based on 3D graphical presentations using a given set of pieces (in 3rd year) to build 3D shapes based on 2D graphical presentation using a given set of pieces (in 4th year). Another part focuses on cyphers. The children get encrypted tasks which they had to decipher. The clues to deciphering were either embedded in graphical presentations that accompanied the problem or were given at the end of the encrypted text.
- **Geometry** work includes four-line symmetry, finishing partially fragmented or degraded images when only an outline is given (connecting dots to make an image) based on a smaller version of the original shown in the picture.

6 Observations and Lessons Learned

These courses have been taught over a four-year period as extra-curricular activities in a primary school. One group per year, therefore four groups per academic school year. The average number of students per group was 12. The program was developed for children ranging from 6–9 years of age. The ratio of girls to boys on average was close to 50/50. Generally, classes were attended by children of all abilities, both groups with strong mathematical capabilities and those being weaker in the maths classes. One group has passed through all four stages during this process. The aim here was to foster an enjoyment in acquiring CT capabilities, without a pressure for good performance or even any awareness of being monitored and/or assessed in this process. The parents' reason to enrol their children ranged from initiating an interest and developing skills in the topic to supporting already existing interest.

The setting also left out an important aspect of education, which is the assessment. As this was taught to children who participated voluntarily and as this was not part of the standard curriculum (and thus not to be graded), no formal evaluation was carried out. Thus, attainment evaluation was not part of the course settings, but of course this might be a concern for courses fully integrated into a schools curriculum. However, CT assessment in general, beyond primary schools, remains a very immature area that requires more research, as we already noted in Sect. 2.

We looked, however, at the overall experience with factors such as engagement and joyfulness as the key evaluation criterion. Increasing confidence is another aspect. We also comment on progress and attainment evaluation.

The **childrens perception** is positive. All children liked the classes and found them great for the following reasons: (i) TOtB brought out a playful side of mathematical and computational topics by exploring patterns and numbers through pictures, letters, words and shapes for the youngest children, (ii) TOtB brought out a sense of adventure in trying to solve different tasks for the older children, (iii) they learnt that there is often more than one way to solve a problem and that by trying different approaches they can find quicker way to solve tasks. This is despite the fact that the classes did create intentional challenges:

- They (as a group) had to conclude themselves what the tasks were about, i.e., they were not told what to do.
- In order to solve tasks, they had to look/search for a pattern/(s) that would not be immediately obvious.
- They had to find connections between different things they would normally not connect, e.g. numbers, letters, shapes, etc.
- They had to be precise/accurate in what they do in order to solve the tasks.

In order to elicit some progress results, we asked for the **parents' perception** informally. The parents were very positive about the general impact of the course. A general observation they had was that children became more focused and quicker in solving tasks. This is confirmed by the **instructor's observations**:

- The children got increasingly more focused in what they were doing and were more quickly able to identify the approach and solve the problem.
- They were able to understand problems better by organising the information into smaller pieces/blocks.
- By understanding smaller blocks, they were able to understand better how these were interconnected into one whole.

Other observations relate to **long-term progress**. In each year/class, about 80% of children would continue with the classes in the following year.

- During any year, their general awareness increased, which resulted in noticeable progress in the childrens ability to understand the tasks given. As a consequence, they were able to solve the tasks quicker and more accurately.
- In particular first class pupils became more confident with not only the numerical aspect of maths, but also the language part as they were constantly encouraged to bring the two together. Working at this intersection of domains from the very beginning has been crucial for the progress they made.
- The strongest effect was observed in children who took part for all 4 years, as the classes were a medium-to-long-term project where children are systematically encouraged to think, ask questions and question what they find, analyse and make connections amongst the things that might not be at first glance related to one another and use what they find to solve problems.

Our wide approach and the variety of problems have made mathematical and computational concerns more interesting and less dry.

A limitation that applies to CT in general is the lack of reliable assessment methods. Specifically in the context of children progress and attainment are difficult to measure. Due to the lack of specifically non-intrusive methods, we have restrained from evaluating these concerns more formally than the qualitative assessment presented. Not using formal assessment also made monitoring systematically and adjusting material to the needs of the children more difficult.

7 Conclusions

Our main observation from our experience is that CT can be started from an early stage, beginning with the first year in primary school. As maths is an element in generally all curricula, we complement and expand on maths on a pathway into computational knowledge and skills towards computing education.

We see this as a pathway towards more computing (and in this sense programming) activities, but we left out early exposure to computers in order to deepen the core CT capabilities first. The emphasis has been on the core CT capabilities from algorithmic thinking and logic to evaluation, make available through a mix of arithmetric, geometry and space related, cryptographic and language related and applied problems.

It has also been an important objective to demonstrate that CT can be applied to non-maths and non-computing subjects, thus emphasising the relevance of CT beyond computing as a technical subject.

Assessment has already been discussed as an issue that needs further investigation [16]. Another direction is an online support system that allows children to interact and that adapts to their abilities [15, 18, 19].

References

1. Aho, A.V.: Computation and computational thinking. Comput. J. **55**(7), 832–835 (2012)
2. Barr, V., Stephenson, C.: 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 (2011)
3. Berry, M.: Computational thinking in primary schools (2014). http://milesberry.net/03/computational-thinking-in-primaryschools/
4. Brennan, K., Resnick, M.: New frameworks for studying and assessing the development of computational thinking. In: 2012 Annual Meeting of the American Educational Research Association (AERA12), pp. 1–25 (2012)
5. Computing at School: Computational Thinking (2016). http://community.computingatschool.org.uk/resources/252
6. Charlton, P.: Computational thinking and computer science in schools (2012). https://www.lkldev.ioe.ac.uk/lklinnovation/wp-content/uploads/2013/01/Time-to-ReLoadWhatTheResearchSaysBriefing27April2012.pdf
7. Crow, D.: Why every child should learn to code (2014). http://www.theguardian.com/technology/2014/feb/07/
8. Computer Science Teachers Association – CSTA Computational Thinking Task Force: Computational Thinking Resources (2016). http://csta.acm.org/Curriculum/sub/CompThinking.html
9. Curzon, P., Dorling, M., Ng, T., Selby, C., Woollard, J.: Developing computational thinking in the classroom: a framework. in: computing at school (2014)
10. Fronza, I., El Ioini, N., Corral, L.: Students want to create apps: leveraging computational thinking to teach mobile software development. In: 16th Annual Conference on Information Technology Education (SIGITE 2015), pp. 21–26 (2015)
11. Google for education: exploring computational thinking (2016). www.google.com/edu/computational-thinking/index.html
12. Grover, S.: Systems of Assessments for deeper learning of Computational Thinking in K-12. In: Annual Meeting American Educational Research Association, pp. 1–9 (2015)
13. Grover, S., Pea, R.: Computational thinking in K-12: a review of the state of the field. Educ. Res. **42**(1), 38–43 (2013)
14. ISTE and CSTA: Computational Thinking – Teacher Resources (2nd edn.) (2011). http://csta.acm.org/Curriculum/sub/CompThinking.html
15. Kenny, C., Pahl, C.: Intelligent and adaptive tutoring for active learning and training environments. Interact. Learn. Environ. **17**(2), 181–195 (2009)
16. Holohan, E., Melia, M., McMullen, D., Pahl, C.: The generation of e-learning exercise problems from subject ontologies. In: 6th International Conference on Advanced Learning Technologies ICALT 2006, pp. 967–969 (2006)
17. Koh, K.H., Basawapatna, A., Bennett, V., Repenning, A.: Towards the automatic recognition of computational thinking for adaptive visual language learning. In: Symposium on Visual Languages and Human-Centric Computing (2010)
18. Pahl, C., Kenny, C.: Interactive correction and recommendation for computer language learning and training. IEEE Trans. Knowl. Data Eng. **21**(6), 854–866 (2009)
19. Murray, S., Ryan, J., Pahl, C.: A tool-mediated cognitive apprenticeship approach for a computer engineering course. In: 3rd IEEE International Conference on Advanced Learning Technologies ICALT 2003, pp. 2–6 (2003)

20. Mooney, A., Duffin, J., Naughton, T., Monahan, R., Power, J., Maguire, P.: PACT: an initiative to introduce computational thinking to second-level education in Ireland. In: International Conference on Engaging Pedagogy (2014)
21. Settle, A., Franke, B., Hansen, R., Spaltro, F., Jurisson, C., Rennert-May, C., Wildeman, B.: Infusing computational thinking into the middle and high-school curriculum. In: Conference on Innovation and Technology in Computer Science Education (2012)
22. Werner, L., Denner, J., Campe, S., Kawamoto, D.C.: The fairy performance assessment: measuring computational thinking in middle school. In: 43rd ACM Technical Symposium on Computer Science Education (SIGCSE12). 215–220 (2012)
23. Wing, J.M.: Computational Thinking. Comm. ACM **49**, 3 (2006)
24. Wing, J.M.: Computational thinking benefits society (2014). http://socialissues.cs.toronto.edu