CAMBRIDGE Mathematics

IN PARTNERSHIP WITH NUI Galway OÉ Gaillimh

ESPRESSO DOPPIO

ISSUE 24 NOVEMBER 2019

RESEARCH, FILTERED BY CAMBRIDGE MATHEMATICS

TALKING POINT:

WHAT ARE MATHEMATICAL THINKING AND COMPUTATIONAL THINKING AND WHAT IS THE RELATIONSHIP BETWEEN THEM?

IN SUMMARY

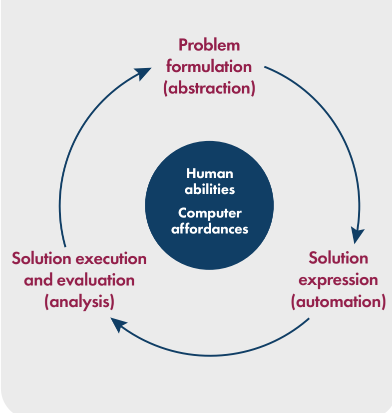
- Mathematical thinking (MT) and computational thinking (CT) are both types of abstract problem solving approaches which may have some overlap
- MT and CT are similar in that they can both be improved upon by practice with reflection; they may support one another
- CT is evolving in comparison with the longer-established MT
- Using both MT and CT might support students in feeling comfortable with trial and error, ambiguity and flexibility. Both support students becoming independent learners and can be learned at any age
- CT is more constrained by hardware and real-world constraints than MT; CT may apply more broadly than MT
- Processes common to both may include decomposition, algorithm design and modelling thinking
- MT can be practised in the context of computational tools and CT in the context of mathematics

Comparing conceptions of mathematical and computational thinking cycles

Mathematical problem solving cycle (iterative)

Adapted from Salado et al (2018)

Computational thinking cycle (iterative)



Adapted from Repenning et al, in Rich & Hodges (2017)

1

Mathematical thinking (MT) and computational thinking (CT) are interrelated and are both highly complex. MT is a process that can be viewed through many different lenses and involves the application of mathematical skills to solve maths problems[1] and is provoked by contradiction, tension and surprise and supported by an atmosphere of questioning, challenging and reflecting[2]. CT is 'an approach to solving problems in a way that can be implemented with a computer' and is distinct from computer science[3]. In comparison with MT, CT is a relatively new area of research without complete consensus on a definition, but it has been suggested it may comprise problem decomposition, abstraction, algorithmic design, debugging, iteration, and generalisation[4]. It also involves an iterative design, refinement and reflection process that is central to creative thinking[5]. Supporting teachers to learn more about CT and how it might support learning in a variety of contexts has been identified as a priority[6].

IMPLICATIONS: Compared with MT, CT is a new domain that requires further research in understanding its place in the classroom

Shared information about MT and CT help students to see the connections between them. It is useful for mathematics teachers to know about CT and for computing teachers to know about MT

'The power of computational thinking is that it applies to every other type of reasoning'

Barr & Stephenson, 1999

'Science and mathematics are becoming computational endeavors'

Weintrop et al, 2015

2

There are similarities between MT and CT. Both are problem-solving methodologies as they involve the recognition of pattern in problem structures. They both also involve processes such as *decomposition* (breaking down problems into smaller steps); *algorithm design* (working out general principles from multiple examples); and *modelling thinking* (translating objects or phenomena from the real world into mathematical equations, and/or computer relations)[7]. There are also more general heuristics and behaviours in problem solving that they have in common, such as abstracted thinking and metacognition, building up comfort with trial and error, ambiguity, flexibility, and being able to consider and evaluate multiple ways to solve problems. Both MT and CT can be developed at any age and, when learners are proficient enough, practised with increasing independence[4].

IMPLICATIONS: Processes common to both may include decomposition, algorithm design and modelling thinking

Applying both MT and CT may support students in building up comfort with trial and error, ambiguity and flexibility

MT and CT are both ways to engage in problem-solving and can both be improved upon by practice with reflection, can be developed at any age and may allow for students to become more independent learners

3

There is increasing interest in including some aspects of computational thinking within mathematics curricula: skills such as pattern recognition and decomposition, designing and using abstraction, using appropriate computing tools and defining algorithms have been identified as part of the mathematical problem solving process.[8]

The use of computational tools and skillsets can deepen the learning and experience of mathematics or provide 'powerful new techniques for employing mathematics to model complex phenomena'[5]. Mathematics also provides a meaningful context (and set of problems) for using computational thinking. CT and MT can intersect – for example, in the application of software in the mathematics and/or computer science classroom[9,10,11]. In applying a mathematical concept to software, both CT and MT are used in decomposing the mathematical problem, thinking in an abstract manner, producing or choosing an algorithm suitable for the problem and debugging any errors that may arise. Examples may be found in probability, statistics, measurement and functions where software applications such as MS Excel, Scratch or graphing calculators will involve CT. Curriculum planning documents, such as the new 2021 PISA framework, are beginning to refer to the use of CT in mathematics; for example as part of mathematical literacy[12].

IMPLICATIONS: MT can be practised in the context of computational tools and skillsets; CT can be applied in the context of mathematics

4

Where MT is confined to mathematical problems with mathematical components, there is increasing evidence that CT applies more broadly, and in a more comprehensible manner, to the complex processes and relationships in the arts as well as the sciences[2,4]. Whilst it is possible to use CT to solve theoretical as well as practical problems[4], more usually CT must take cognisance of the physical constraints of computing hardware and of the real world, whereas MT tends more towards an abstract structure. One simplistic model which considered differences highlighted aspects such as *data mining*, *networking* and *robotics* as singular to CT, and *arithmetic*, *algebra* and *geometry* to MT[13]. As research into school-based CT develops, more will be understood about its relationship with MT and where it should reside in the curriculum.

IMPLICATIONS: CT is more constrained by hardware and real-world constraints than MT, but may apply more broadly

Any comparison between MT and CT should take into account concepts, practices and perspectives in order to be useful

Further research is needed to clarify the implications of CT and its place in national curricula

Lucy Rycroft-Smith & Cornelia Connolly (NUI Galway), 2019

REFERENCES

1. Stacey, K. (2006). What is mathematical thinking and why is it important. *Progress report of the APEC project: collaborative studies on innovations for teaching and learning mathematics in different cultures (II)—Lesson study focusing on mathematical thinking.*
2. Mason, J., Burton, L., & Stacey, K. (2011). *Thinking Mathematically.* (2nd Ed. ed.): Pearson Higher Ed.
3. Barr, V., & Stephenson, C. (2011). Bringing Computational Thinking to K-12: What is Involved and What is the Role of the Computer Science Education Community? *ACM Inroads, 2*(1), 48–54.
4. Shute, V. J., Sun, C., & Asbell-Clarke, J. (2017). Demystifying computational thinking. *Educational Research Review, 22,* 142-158.
5. National Research Council. (2011). *Report of a workshop on the pedagogical aspects of computational thinking.* National Academies Press.
6. Voogt, J., Fisser, P., Good, J., Mishra, P., & Yadav, A. (2015). Computational thinking in compulsory education: Towards an agenda for research and practice. *Education and Information Technologies, 20*(4), 715-728.
7. Liu, J., & Wang, L. (2010). *Computational thinking in discrete mathematics.* Paper presented at the Second International Workshop on Education Technology and Computer Science.
8. Wilensky, U. (1995). Paradox, programming, and learning probability: A case study in a connected mathematics framework. *The Journal of Mathematical Behavior, 14*(2), 253-280.
9. Hickmott, D., Prieto-Rodriguez, E., & Holmes, K. (2018). A scoping review of studies on computational thinking in K–12 mathematics classrooms. *Digital Experiences in Mathematics Education, 4*(1), 48-69.
10. Sung, W., Ahn, J., & Black, J. B. (2017). Introducing computational thinking to young learners: Practicing computational perspectives through embodiment in mathematics education. *Technology, Knowledge and Learning, 22*(3), 443-463.
11. Rohaeti, E. E., & Bernard, M. (2018). The Students' Mathematical Understanding Ability thought scientific-assisted Apporach of Geogebra Software. *Infinity Journal, 7*(2), 165-172.
12. PISA 2021: Mathematics Framework. (n.d.). Retrieved October 22, 2019, from https://pisa.e-wd.org/
13. Sneider, C., Stephenson, C., Schafer, B., Flick, L., Computational Thinking in High School Science Classrooms: Exploring the Science "Framework" and "NGSS", *The Science Teacher, 81*(5), p. 53-59.

Cambridge University Press | University of Cambridge Faculty of Mathematics | University of Cambridge Faculty of Education | Cambridge Assessment