Computational/Algorithmic thinking in the school mathematics

Djordje M. Kadijevich

Institute for Educational Research, Belgrade, Serbia

A keynote talk at the minisymposium “Mathematics in Education”
at the 8th European Congress of Mathematics (https://www.8ecm.si)

Main terms

- Computational thinking (CT) – An ability to recognize aspects of computations in various problem situations, and to deal appropriately with those aspects by applying computer tools; CT calls for decomposition, abstraction, algorithms, …

- Algorithmic thinking (AT) – An ability to comprehend, test, improve, or design an algorithm – a solution to a mathematical problem expressed in a sequence of clearly defined instructions that process some numerical, symbolic or geometric data; AT calls for decomposition, abstraction, ...

- Mathematics educators and mathematicians may prefer to use AT to distinguish its place in mathematics curriculum from components of computer science curriculum

Presentation outline

1. Global context

2. Critical CT/AT issues: Definition, state of research, curricular aspects

   Stephens, M., & Kadijevich, D. M. (2019). Computational/algorithmic thinking. In Lerman, S. (Ed.), *Encyclopedia of mathematics education* (pp. 117–123). Cham, Switzerland: Springer.

   Kadijevich, D. M., & Stephens, M. (2020). *Three important aspects of research on computational/algorithmic thinking*. Paper accepted for presentation at ICME-14 TSG-14 “Teaching and learning of programming and algorithms”

3. Cultivating CT through data practice

   Kadijevich, D. M. (2019). Cultivating computational thinking through data practice. In Passey, D., Bottino, R., Lewin, C., & Sanchez, E. (Eds.), *Empowering learners for life in the digital age* (pp. 24–33). Cham, Switzerland: Springer.

   Kadijevich, D. M. (2019). Interactive displays: Use of interactive charts and dashboards in education. In Tatnall A. (Ed.), *Encyclopedia of education and information technologies* (pp. 968–973). Cham, Switzerland: Springer.

4. CT/AT at ICME-14

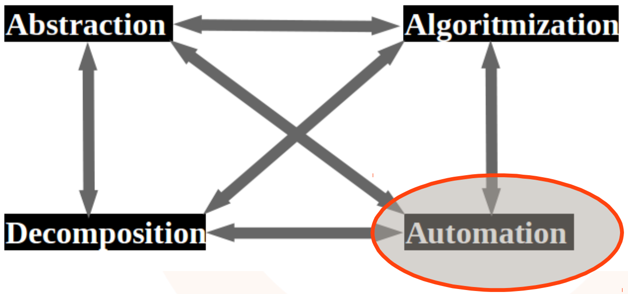
1. Global context

- Data science (DS) trend, CT/AT critical to DS practice; *International Data Science in Schools Project*: integrate computational and statistical thinking in a pre-calculus DS course
- PISA 2021: CT included in modelling steps (formulating, employing, interpreting & evaluating, reasoning); CT skills call for abstraction, decomposition, algorithmic thinking, and automation
- International Computer and Information Literacy Study (ICILS) assessed CT in 2018; Tasks required to analyze problems, break problems into subproblems, find sequences of steps/instructions to solve them, ...

2.1 Critical CT/AT issue: *Definition*

- Various CT definitions reflecting a context (programming, STEM education, K-12 school subjects); Focus on their similarities rather than differences

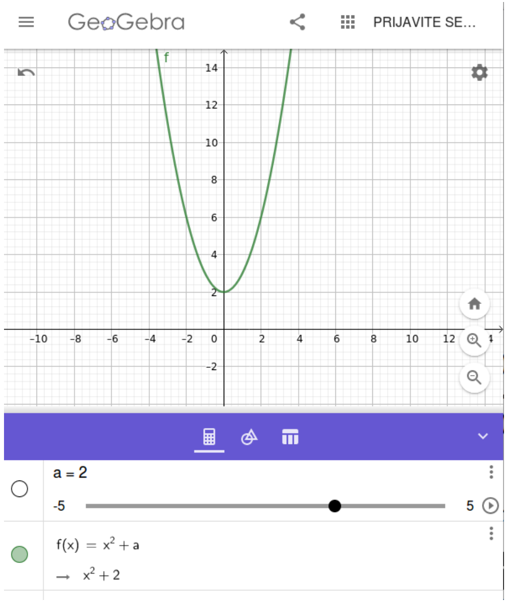
- CT cornerstones: decomposition, abstraction,algorithmisation, and automation; Use automation to separate AT from CT

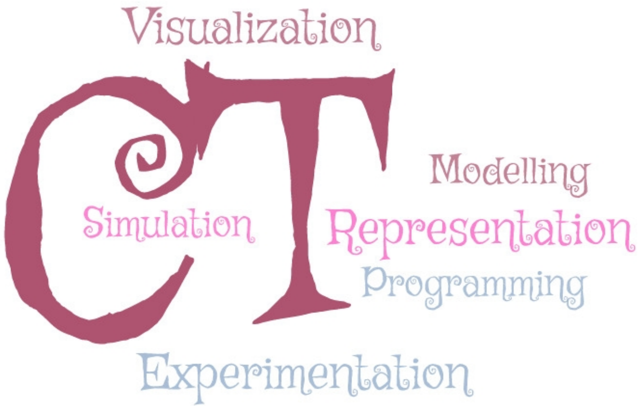


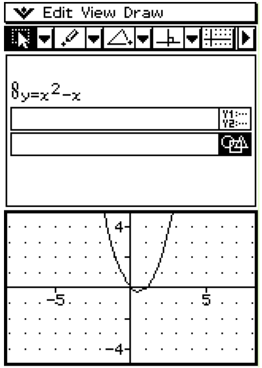
CT *without* computers?; History – developing computing machines through centuries; Today – the human is formulating a problem solution to be carried out by a computer (not by a human)

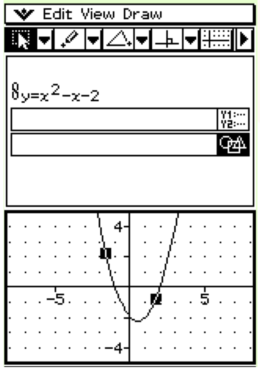
... /

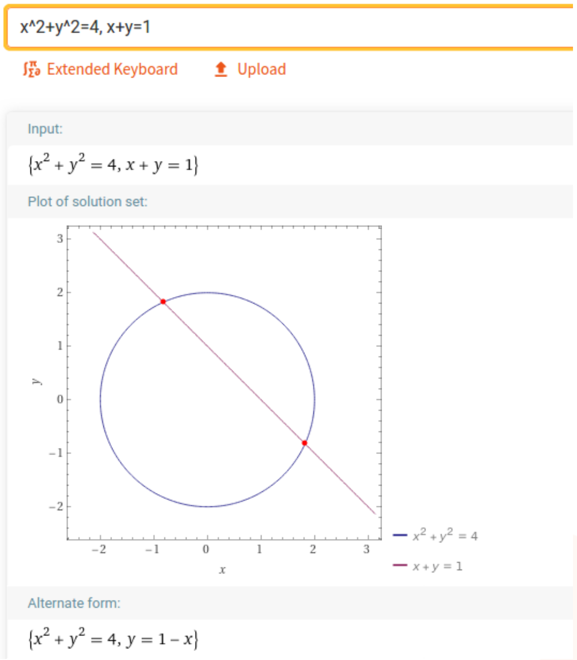
Programming is just a part of CT!



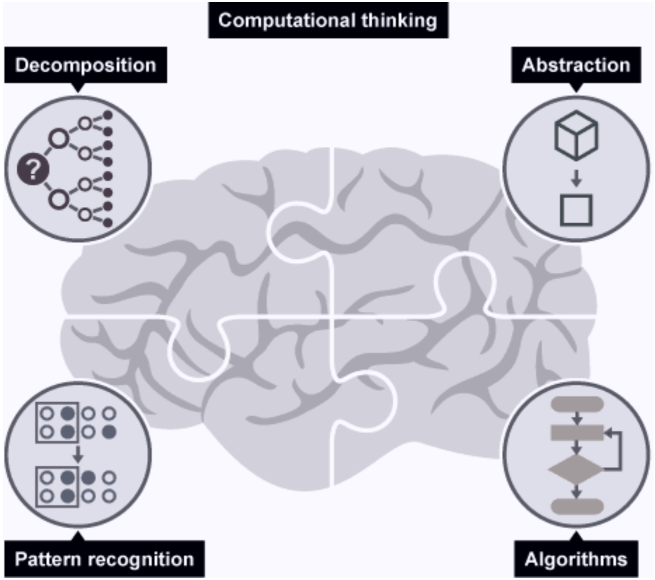




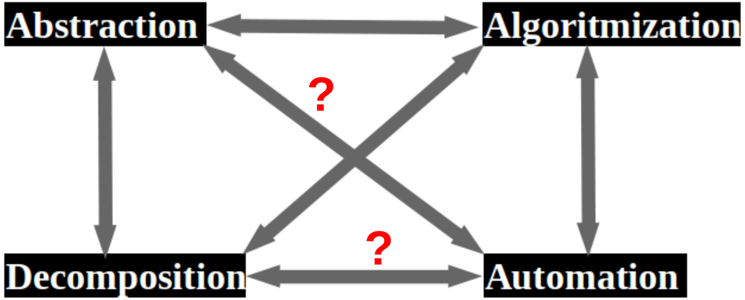




... /



vs



Critical questions:

The status of Automation

Abstraction *vs* Abstra**C**tion; Decomposition *vs* De**C**omposition ... /

AbstraCtion

Known facts about triangle

```
side(a).
side(b).
angle(alpha).
angle(beta).
opposite(alpha, a).
opposite(beta, b).
greater_side(a, b).
```

New fact added

```
greater_angle(alpha, beta).
```

Discovered rule afterwards

```
greater_angle(X, Y) :- angle(X), angle(Y), opposite(X, X1), opposite(Y, Y1),
  side(X1), side(Y1), greater_side(X1, Y1).
```

“The greater side is opposite the greater angle; the greater angle is opposite the greater side.”

“A greater side of a triangle is opposite a greater angle.”

“A greater angle of a triangle is opposite a greater side.”

https://themathpage.com/aBookI/propI-18-19.htm

… video

2.2 Critical CT/AT issue: *State of research*

- Limited research on CT/AT
- About CT: Use programming to explore numbers, operations, and geometry, but also use modeling, simulations, and data analysis to explore functions, probability, and statistics
- About AT: Design algorithms to develop procedural knowledge that is rich in connections; To develop conceptual knowledge, use algorithms to ask advanced questions about their results
- Using CT/AT lens may result in a more focused instruction on AT and its core components (aided by some automation) that are relevant to mathematical thinking (see the abstraction–modeling–problem-solving model)

2.3 Critical CT/AT issue: *Curricular aspects*

- Use a cross-curriculum model for the integration of CT/AT across subject areas (Finland)
- Teach CT/AT within the information/digital technologies curriculum (England, Australia)
- Introduce “programming thinking” in several subject areas, including mathematics (Japan)
- Use a separate subject in the middle years taught by mathematics and information technology teachers (France); Use a senior secondary subject devoted to the study of algorithmics (Victoria, Australia).
- Different models remain unexamined and are by and large untested

… /

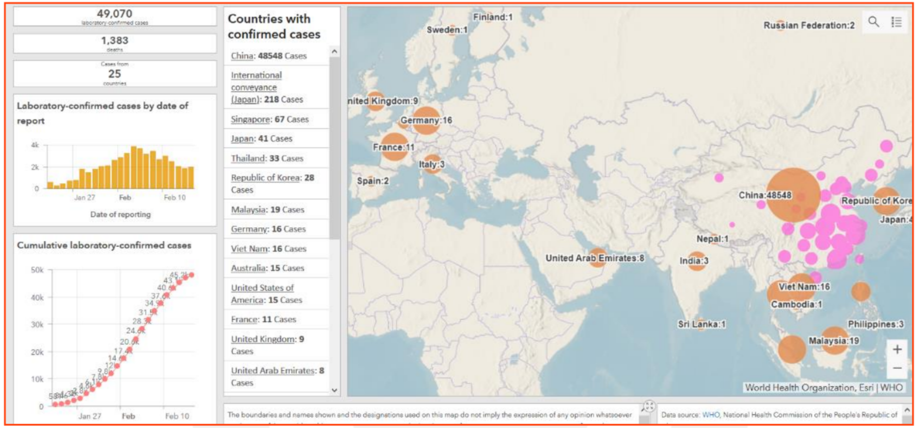
- Key question: Which model to apply in a particular educational context and culture bearing in mind its *pros* and *cons* in fostering creative interfaces between mathematics and computer science?
- Creative interfaces offered by: computational algebra, computational geometry, algorithmic graph theory , …
- Activities: Use algorithms to unpack concepts and procedures; Identify and refine variables and parameters to use a given algorithm; Apply algorithms to provide accessible introductions to modeling, optimization, operations research, and experimental mathematics; Use AT to explore whether a solution exists *vs* how a solution (if it exists) can be found
- Not only apply programming; Support key elements of mathematical reasoning by CT/AT; Use AT/CT to develop this reasoning

3. Cultivating CT through data practice

- In ICILS 2018, CT was operationalized by tasks that called for programming tasks as well as structuring and manipulating data sets;
- Two CT definitions: Core CT concepts: data collection, data analysis, data representation, problem decomposition, abstraction, algorithm and procedures, automation, parallelization, and simulation / Main CT practice in STEM context: *data practices* (e.g., collecting, visualizing), *modeling and simulation practices* (e.g., building and using computational models), …
- Working with data can activate various CT components, such as abstraction, decomposition, and pattern recognition
- Data science trend: Uncover useful scientific, professional, or educational patterns through simple data modelling using *interactive displays* for example

… /

Interactive displays – Dashboards



- These displays are digital artifacts that comprise one or more interactive reports (charts, tables, or summary measures)
- Dashboards are sets of two or more interactive reports, mostly interactive charts, whose content updates automatically whenever changes in data or variables considered occur
- Powerful tool for visualizing data, i.e. for explorative data analysis
- Used in various industries and areas (e.g., learning analytics); Summarize performance indicators of various subjects and objects (e.g., students, vehicles, patients, publications)

Activities, underlying skills, learning opportunities

- Activities: Asking questions, Visualizing data, and Answering questions; Data preparation and Validating modelling may be added later
- Skills: Choosing relations to examine, identifying dependent and independent variables (Asking questions); Selecting charts and measure to use (Visualizing data); Recognizing regularities in charts produced, and connecting regularities to questions asked (Answering questions)
- Opportunities: Cultivating a modeling/data inquiry cycle, Supporting the development of important disciplinary notions (e.g., variable and functional dependence); Promoting a basic understanding of CT strategies, such as decomposition as well as rapid prototyping, top-down, and bottom-up approaches

Challenges, reasons, scaffolds

- Challenges: Using appropriate sets of variables to answer questions; Selecting appropriate charts and measures; Using context properly to interpret findings; ...
- Reasons: Complexity of this data practice as a design task; Complex interactions of knowledge from different domains; ...
- Scaffolds: Connect the activities i.e. their underlying skills (e.g. variables selection with charts production; charts production with regularities recognition); Provide a description of the underlying context; Develop problem structuring skills while improving knowledge of context under scrutiny, and vice versa; ...

Environment to use, path to follow, progress to assess

- Environment: Use rich computational environments that support various CT assets (e.g., ZOHO Analytics)
- Path: Use displays to understand or evaluate data practice (DP) completed – modify displays to debug or extend DP done – create displays to perform full DP by yourself
- Assessment: Use students' portfolios about dashboards evaluated, improved, or fully developed; Evaluate success of pursuing each DP activity and linking them in terms of underlying skills

Focus of CT pedagogy (didactics) and recent developments

"Interactive visualizations or simulations are at the heart of computational thinking … A second focus for pedagogical environments is the modeling and trouble-shooting of data sets. … A third focus is searching for patterns in large data sets."

National Research Council (2011). *Report of a workshop of pedagogical aspects of computational thinking* (p. 17). Washington, DC: The National Academies Press.

"Together with theory and experimentation, a third pillar of scientific inquiry of complex systems has emerged in the form of a combination of modelling, simulation, optimization, and visualization."

EMS (European Mathematical Society) (2011). *Position paper on the European Commission's contributions to European research* (p. 2). The Author.

Niss (2016): Consider curriculum as a 6-component vector (goals, content, materials, forms of teaching, student activities, assessment)

4. CT/AT at ICME-14

TSG “Teaching and learning of programming and algorithms”

DG “Computational and algorithmic thinking, programming and coding in the school mathematics curriculum: Sharing ideas and implications for practice”

Organizers: Max Stephens, Australia; Djordje M. Kadijevich, Serbia; Zhang Qinqiong, China

Areas to consider: 1) current or proposed curriculum provisions/developments from their home country; 2) relevant classroom/teaching activities; and 3) resources to support teachers

Contribution format: For each area, a 300-400-word account (including your name, country and affiliation, plus one/two references) should be sent by June 30 to Max Stephens at m.stephens@unimelb.edu.au