Universal Journal of Educational Research 6(11): 2669-2678, 2018 http://www.hrpub.org
DOI: 10.13189/ujer.2018.061133

Compute mindlessly. Not! Map consciously

P. Niemelä[1,*], V. Mikkolainen[2], J. Vuorinen[3]

[1]Pervasive Computing, Tampere University of Technology, Tampere, Finland

[2]Department of Education, City of Tampere, Finland

[3]Ideal Learning Oy, Tampere, Finland

Abstract This paper utilizes concept mapping as a tool for conscious and deliberate knowledge building in mathematics and its extension to algorithms. Currently, alleged defects in mathematics education are obvious: instead of conceptual elaboration, everyday praxis relies on routine computations that are likely to lead into alienated concepts with weak connections to prior knowledge. A concept map visualizes the existing conceptual structure, and whenever new information is brought in, it will be placed in the map by clearly explicating its linkage to the previous concepts. In the Finnish mathematics education, such new knowledge is programming content that is integrated into elementary school mathematics in 2014 Finnish National Curriculum. This content is crystallized as the requirements of computational and algorithmic thinking, the utilization of respective data structures, and adequate amount of hands-on practice to internalize good coding conventions. This study examines secondary ($N = 19$) and higher education students ($N = 10$) and their conceptual knowledge of mathematics concentrating on the domain of algorithms in particular. The concept maps drawn by the students are evaluated using the SOLO taxonomy. To conclude, a consensus map of algorithms is represented and linked to the elementary mathematics syllabus.

Keywords Concept Mapping, Visualizations, Meta-cognitive Skills, SOLO Taxonomy

1 Introduction

Oftentimes, school mathematics is interpreted solely as problem solving. This approach highlights the discovery aspects of finding a correct solution, and procedural fluency in carrying out the actual calculations. Mastering the concepts and constructing consistent schemata have not been in the center of the gravity of teaching. In progressing further in mathematics, however, may prove to be problematic without a strong conceptual basis especially concerning such open-ended problems that require intuition and a well-parsed overall view. Deep learning implies such a basis of densely linked concepts constituting a well-organized structure. Internalizing essential concepts implies connecting the concepts with prior knowledge. New concepts are either assimilated without modifications or accommodated with modifications as a part of the schemata [1].

The current views of cognitive constructivism range from weak to strong [2]: two extreme poles are the empiricist position of considering a mind to be shaped by nature in relatively passive fashion, the other extreme being 'radical constructivism' that holds all knowledge perspective and being a result of active construction on the part of learners [3, 4]. Constructivism being the dominant learning theory in the Finnish elementary school curricula, students are compliantly encouraged as active constructors of their own knowledge [5, 6].

Plenty of tools and methods exist for fostering the deliberate associations of new concepts into one's schemata. For example, Ausubel et al. promote advance organizers, i.e., short textual abstracts that enable bridging new and old knowledge, and the authors consider conscious organizing as the main means of meaningful learning [7]. In presenting scientific explanations, Mayer et al. introduce the rule of thumb of three 'C's: conciseness, coherence, and coordination. Accordingly, Ausubel's advance organizers contain all essentials yet briefly; concerning text, less is often more [8]. In moving from textual to ever more concise representations, that is, in symbolic or graphical ones, cognitive tools such as concept and mind maps become exploitable [9]. For example, Novakian concept maps can be constructed by using CMapTools application that Novak's group developed for the purpose [10]; allegedly, it increases accessibility and reduces the redundancy of information [11]. Similarly, Buzanian mindmaps address the issue of organizing one's thoughts and taking notes efficiently. However, the approach is less constrained: no link verbs are required in preference to all possible visual means to associating the concepts, which targets to unlock creativity [12].

In visualizing the conceptual structures, a concept map also exposes potential misconceptions, thus enabling teachers' attempts of aligning them. Hence, the map provides a means to evaluate a student's conceptual learning [13]. Hay and Kinchin specify an evaluation rubrics based on, e.g., the general shape of a map, whether a spoke, a chain, or a network; the levels of a hierarchy; and linking style, where cross-links rank high. By analyzing these factors, the growth in sophistication is demonstrable [14]. The sophistication is also pivotal in distinguishing experts from novices. Williams claims that *'individuals whose knowledge within a particular domain is interconnected and structured will activate large chunks of information when they perform an activity in that knowledge domain'* [13]. In contrast, novices' knowledge can be described as a set of un- or loosely related facts. As an implication of these apparent differences, Royer et al. confirm that the measures of knowledge organization and structure provide the appropriate indexes of skill development [15]. After analyzing the concept maps of the 'function' in mathematics, Williams addresses the general homogeneity embraced the experts' maps that deviates them from the students' ones [13]. She summarizes, *'Experts' maps and their distinct variance from the students' maps lend to the conclusion that concept maps do capture a representative sample conceptual knowledge and can differentiate well among fairly disparate levels of understanding.'*

In general, such modeling and abstraction skills are beneficial not only for the purposes of knowledge building and 'learning to learn' [9], but, moreover, for modeling and abstracting problems in general, the potential applications ranging through various domain areas. This study concentrates on the domains of mathematics and computer science, where map- and graph-like elaborations are common in software education, because of their general applicability. For instance, in software modeling map-like conceptualizations are crucial throughout the process in order to design and communicate an overall architectural component and class structure of the system. Especially, UML diagrams, such as class and sequence diagrams, are exploitable for the purpose. Notably, the class diagram in UML is analogous to concept mapping; however, the linking verbs are limited to these four options: association, aggregation/composition, and generalization.

This study aims to examine ways of leveraging the procedural routines commonly exploited in mathematics to more conceptual elaborations. For the purpose, visualizations such as maps and graphs have proven to be beneficial yet not fully exploited to deepen a mathematical insight. Thus, the study utilizes mapping as a research method to examine the participants' conceptual knowledge. Like the study conducted by Williams [13] that captures perceptions related to function, we examine how higher education people embrace algorithm. This study asks:

- RQ1: Which mathematics syllabus areas do the elementary level students and higher education participants highlight?
- RQ2: Where would the interviewees situate algorithms in the mathematics syllabus?
- RQ3: Which kind of 'crowd-sourced' algorithm concept map can be constructed based on the answers and how could it be aligned with FNC-2014 mathematics syllabus?

To outline the structure of the article, the document flow is shortly introduced: Ch. 2 represents the data collection method and the analysis phase. In analysis, SOLO taxonomy was utilized as evaluation rubrics. Ch. 3 addresses the current conceptual knowledge of higher education students about syllabus areas of mathematics that is compared with the data collected earlier from elementary school. Next, the students situate algorithms to the best-fit mathematics syllabus areas and draw concept maps of algorithms quantified later as a combined crowd-sourced student perception. Ch. 4 reflects this map in the light of the Finnish National Curriculum 2014 (FNC-2014) and searches the links and affordances between teaching mathematics and computing together. In consequence, this study simultaneously hypothesizes a suitable subset for mathematics teachers' pre- and in-service training. Ch. 5 summarizes the results.

2 Data and Methods

Data was collected in two separate periods, in years of 2016 and 2018, and from two different educational levels, that is, at elementary and higher education. Next, the two data collection iterations are described in a more detail, followed by the explanation of the analysis methods.

2.1 Data from Year 8 math students

In 2016, Year 8 students ($N = 19$) of the Hope International School of Cambodia drew maps during their mathematics lessons. The exercise was a part of learning experiment that targeted constructing a holistic view of the topics of the approaching IGCSE exams in Year 10. The learning experiment targeted to familiarize students with the mathematics syllabus and its concepts by exploiting concept mapping. The experiment included several phases, for example:

1. A scaffolded concept map of 'functions', where concept mapping principles were explained
2. Taboo game with syllabus concepts (i.e., describe a given concept without using the 'taboo' word itself)
3. Syllabus read-through: sketchy maps with and without the syllabus
4. Presentations, a syllabus area per a group
5. A scaffolded concept map of 'Trigonometry'
6. The final concept maps of the current and next syllabus areas
7. Poster combining all the maps together
8. Survey

The IGCSE math syllabus splits into eleven separate areas; the first area handled was 'functions'. First, a teacher demonstrated an example of constructing a map. Next, the students had to identify the main concepts from the text and build a map using these concepts. The text was written to open up the short bullet-list-style of the actual IGCSE syllabus and examination requirements. Because of the conciseness of the original list, the students found difficult to capture the basic idea of the topics, quite a few of which were new. After digesting 'functions', the whole syllabus was to be run through. The first joyful practice was playing the 'Taboo' game, where a student had to explain the concept to his team by avoiding the use of the word itself. Next, it was time to run through the whole syllabus, one colored paper per each syllabus area was filled with syllabus topics followed by linking related concepts and grouping the close-related areas together, such as Geometry and Coordinate Geometry. Each group selected one syllabus area for their presentation, which was videoed. Another mapping example, in this case Trigonometry, was represented and simultaneously the area was revised in general, to start transit back to concept mapping. The remaining syllabus areas were distributed among the groups advised to make the maps as representative as possible with a lot of graphics and symmetry. In the final phase, these concept maps were merged as a revision poster, and the surveys were filled out.

The qualitative data of surveys was exploited as a qualifier of success of this learning experiment. In maps, we were interested in the concepts, whether formal or informal, the consistency of the hierarchy levels and the relatedness of the concepts.

2.2 Data from higher education interviewees

The 2018 study was conducted before summer holidays, by randomly interviewing employees ($N = 10$) of Pervasive Computing Department about their perceptions about algorithms and their best position in the mathematics syllabus. The concept of algorithm was selected, because FNC-2014 mandates teaching 'algorithmic thinking' [6] that is a new and unestablished topic. The interview questions were:

1. Which syllabus areas of mathematics can you still recall?
2. Define algorithm
3. To which mathematics syllabus area would you connect the algorithms?
4. Which are the most essential subtopics of algorithms to be taught?
5. Draw a concept map starting from the algorithm.
6. What would you think about concept mapping as part of mathematics lessons?

Each interviewee was met separately; the interviewer asked the questions and wrote down the responses, except the two last questions: the map was drawn by the interviewee, as well as concept mapping as a learning method was evaluated by the responders themselves.

2.3 Evaluating responses and maps

Textual feedback of the structured interviews was analyzed with content analysis methods, concept maps were evaluated through the prism of SOLO taxonomy, and the validity of the results was estimated by using FNC-2014 mathematics syllabus as a point of reference.

Open-ended and creative question types, such as an assignment of drawing a concept map, reveal a student's conceptual knowledge and perceptions, and these representations may demonstrate learning [16, 17]. According to Gilbert, the development of one's own expressions serves as a valid indicator. In general, expression means talking or writing, for instance explaining to someone one's personal conceptions about the topic.

2.3.1 SOLO taxonomy

Structure of the Observed Learning Outcome (SOLO) systematizes the evaluation of learner's performance [18, 19, 20]. SOLO was chosen as a taxonomy as it provides means to study the overall conceptual understanding of the participants. It postulates a sequence of levels that indicate the growth of the structural complexity: learning progression is categorized into four (or five) different levels represented in the ascending order of sophistication [21]:

1. Pre-structural
   - a task is not attacked appropriately,
   - a student hasn't really understood the point and uses too simple a way of going about it.
2. Uni-/Multi-structural
   - one aspect of the task is picked up and used, but in isolation of the entirety,
   - or several such things are picked up, but treated otherwise similarly as in uni-structural.
3. Relational
   - the coherency of the knowledge increases. This level indicates an adequate understanding about a topic.
4. Extended abstract
   - conceptualization leverages the whole at a higher level of abstraction,
   - the knowledge is generalized and applied to a new topic or area.

This study exploits SOLO to review the conceptual levels of students' algorithm maps that are classified generically by their quality. In addition, the map contents complement the verbal definitions of algorithms gathered in the interview.

2.3.2 FNC-2014 mathematics syllabus and its affordances for algorithms

The data is reflected based on the FNC-2014 guidelines and requirements. For a consistent proceeding, we draft a hypothetical syllabus of algorithm-supportive mathematics, which targets a definition of a consistent path with well-justified building blocks. FNC-2014 divides into three age categories of Years 1–2, Years 3–6, and Years 7–9 [6]. Within one category, educators and related third parties, such as textbook publishers, are free to organize and schedule the content as they desire. In comparison, however, textbooks from different publishers are noticeably akin to each other and have a number of features in common. This is not any downside: the similarity of book series standardizes teaching and facilitates adapting to a new school environment, e.g., if moving from one city to another.

3 Results

3.1 Identified syllabus areas

The learning experiment with Year 8 consisted of a number of exercises that targeted the acquaintance with the IGCSE exam requirements and mathematics syllabus. In clear, the Year 8 students were quite extensively aided with the mapping: they worked as the teams of 3–4 people equipped with the material. Thus, the resulting maps are not a product of their bright memorizations and deep internalizations but rather their collaboration, see Fig. 1.

In contrast, the university students were not prepared for the mathematics syllabus area recall and hence the areas were biased towards the topics they have lately learned, e.g., during university courses or at work-related tasks. This is exemplified by the interviewee accenting number theory as an essential part of algorithms (Case 6, working on cryptography), or by emphasizing linear algebra as a crucial support for algorithms, because of its extensive use in image and video encodings, and pattern recognition at work.

The occurrence frequencies of syllabus areas were recorded from both data. Primarily, Fig. 2 sorts the occurrence frequencies based on the surveys on higher education, and only secondarily based on the Year 8 data. Top-scorers among higher education are algebra, matrices and geometry; in Year 8 algebra, geometry, and number (arithmetic). Year 8 results qualify to their syllabus with the exception of ‘graph’ that is used ambiguously: it is synonymic to statistics, graphical solving, representing, and solving function graphically. In higher education, matrices and vectors total surprisingly high in sum. In addition, the total of continuous mathematics topics would weigh remarkably more, lest the related topics were not scattered widely under such descriptions as derivation, integrals, differential equation, calculus, analysis, and ‘epsilon-delta mathematics’, mentioned once.

When the question about mathematics syllabus areas was set, a few higher education students were puzzled at the outset. First, they could recall none, especially, if plenty of time had passed since the last mathematics courses. They had to take timeout to recall what they have actually learned, e.g., by listing the courses they had accomplished.

3.2 Algorithms in mathematics

As an emergent area of mathematics, algorithms are in focus in this study. In this setup, we examine especially the higher education participants’ conceptual perceptions. As the algorithms have no established position in the syllabus yet, it is interesting to see how they assimilate/accommodate the new topic into their existing mathematics schema. Consequently, in the interview, the following questions address this issue: ‘To which mathematics syllabus area would you connect the algorithms?’, and ‘Which are the most essential subtopics of algorithms to be taught?’.

We group the replies by the syllabus areas proposed: primarily algebra (three mentions of algebra directly, and three indirectly, two functions, and one equations that belong to algebra, altogether six), then ‘Mathematics for Algorithms’ which focused on logic, after which logic as a topic of its own right, and last, the two interviewees who named no math syllabus area. Mathematics for Algorithms refers to the course MAT-02650 provided by Tampere University of Technology in order to learn the needed math contents for algorithms. The responses word by word in the following:

- To algebra. Algorithms should comprise such topics as formulas and operations derived from algorithms. (Case 7, male 35-44 yrs, M.Sc. Marketing)
- To algebra: functions and variables are closely related to algorithms, subtopics would then be recursion, function, data structures, paradigm, efficiency (O-notation), operation, and worst-case (e.g. in sorts).
(Case 9, female 25-34 years, M.Sc., Pervasive Computing Dept.)
- To algebra, possibly number theory. Algorithms require procedural thinking, iterations, indirection, data structures, abstract data types, design patterns and avoiding antipatterns, different programming languages, in particular O-O-programming, polymorphism.
(Case 6, male 25-34 years, M.Sc., Pervasive Computing Dept.)
- Algorithms should be explained simultaneously with functions, because they share the same idea. For example, different sorting algorithms are straightforward to understand and demonstrate. With a purely theoretical treatment, the image may remain too tedious and limited. I would apply algorithms to image processing and provide practical examples of image and pattern recognition. (Case 5, male 25-34 years, B.Sc., Pervasive Computing Dept.)
- While learning equations. The equation is a simple algorithm. Mathematics for Algorithms is somewhat different. Logically though this course would be a good shot. If this course were to be held in our university, I would keep it as a more theoretical one than current course, TIE-20106. For example, plain pseudocode or Python, while the traditional course should comprise hands-on programming. In my algorithms course, there would be less programming and more effort would be put on examining what algorithms are and how they should be designed. (Case 2, male, 25-34 years, B.Sc., Pervasive Computing Dept.)
- To Mathematics for Algorithms, with the enhancement of matrix and vector calculations. As an application, for example, video coding data can be stored as matrices to be

fed for respective algorithms, e.g., to determine the optimum direction of rotation. Algorithms comprise such areas as logic, number theory, and truth tables. (Case 4, male 18-24 years, B.Sc., Pervasive Computing Dept.)

- To Mathematics for Algorithms, even if it was not a meaningful entity, because of too trivial things. On the other hand, execution times, complexity classes and solution spaces were dismissed, although they are important. In addition, the course should teach whether an algorithm is solvable and halts, and if it is an approximation. Generic Methods, optimization theory, basic algorithms that relate to different data structures. You should get a big picture when to use each algorithm and how. (Case 3, Male 25-34 years, PhD, Pervasive Computing Dept., previously in Automation and Hydraulic Eng.)

- To logic and set theory, even probability, it belongs to all areas, really. Set theory is justified in demonstrating the set an item belongs to, similar to data structures in programming. In algorithm courses, I would teach sorting, searching, recursion, algorithm efficiency, which algorithms fit in particular situations, the aspects of R&D, and logic, e.g. De Morgan's laws. (Case 10, male 18-24 years, B.Sc., Pervasive Computing Dept.)

- Algorithms should rather be taught as their own entity, however, programming and algorithms as a whole together with mathematics. It should contain the basics, such as sorting, efficiency. Traveling salesman problem, the shortest route, non-halting problems. (Case 1, male 45-50 years, PhD, Pervasive Computing Dept.)

- No appropriate math syllabus area can be defined for algorithms. They should be taught separately, and after the introduction applied to all mathematics areas, e.g., as algorithmic word problems. In learning algorithms, students should attempt to execute algorithms based on the 'recipe', and be able to explain their functionality. Is the recipe valid? How many steps at minimum is needed to perform the task? A student should be able to evaluate the efficiency of an algorithm as well. Common algorithms, such as sorting, decomposing and managing an array: merge-sort, quick-sort. From my own school experiences, I recall a scales exercise, where we balanced it by moving gold coins, and the number of moves was limited - it was a good example. (Case 8, Male 25-34, M.Sc., Pervasive Computing Dept.).

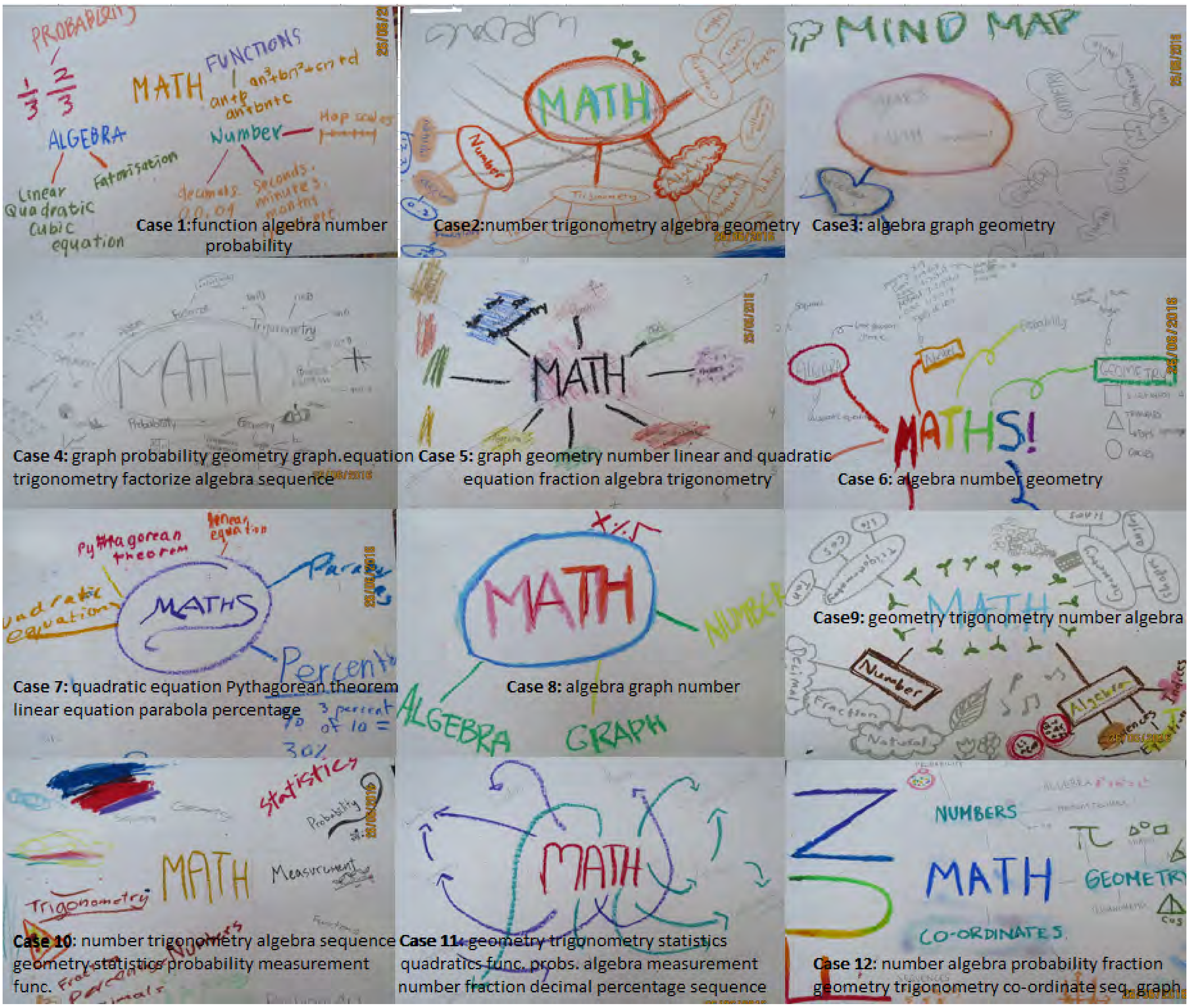
3.3 Algorithm maps evaluated

Among the drawn maps of both groups, concept mapping conventions are not minutely followed (e.g., linking verbs are missing), moreover, consistency is compromised (i.e., general topic first, then more specific at the next hierarchy level). Sophisticated representations are rare in overall, see Figs. 1 and 3. According to the schedule of cognitive development, Year 8 students are novices at abstract thinking compared with participants from higher education, from whom we are permitted to expect much more precise representations. In higher education, the semi-structured interview was, in fact, marshaled to assist in the recall of the most central concepts: the question of algorithm subtopics was stated just before the next step of drawing the map. However, in the drawing phase, a few interviewees suddenly went off at a tangent, exemplified by Cases 1 and 2, 3. Case 1 takes the position of computer science education and highlights the pedagogical viewpoints, his map being interpreted as the following sentences: *'The learning aim of algorithms is to highlight the meaning of them in real-life. Algorithm can be peda[gogical? Editor's addition] One pedagogical method would be visualizing sorts.'* However, these aspects were not prominent in his interview elsewhere. Case 2 took an actor-based standpoint: *'Algorithms are designed by algorithm designer, written in programming languages used by implementer and executed by machines, which are maintained by an admin.'* Based on the answer, it would seem as if the question were to list actors handling algorithms.

Even if the content were highly relevant, a considerable weakness of the map would be in the paucity of important concepts, demonstrated by Cases 4 and 7. Basically, Case 4 suggests following: *'Algorithms employ number theory and logic exercised by truth tables'*, while Case 7 states that *'Algorithm may be a formula'*. The former response is in a resonance with the course 'Mathematics in algorithms'. However, a high-quality map should identify the majority of the relevant concepts, thus these maps remain in the uni-structural level of SOLO taxonomy.

Table 1 summarizes the sophistication levels in SOLO. No idea of the mapping technique or no proper concepts identified implies pre-structural, whereas one hierarchy level only uni-structural, and if multiple levels, then multi-structural. 'Relational' requires cross-linking the nodes of different branches, and 'explicit abstract' generalizations and connections to analogical concepts in different domains. However, most of the maps do not excel in sophistication; especially 'explicit abstract' is absent.

Our closest exemplars in more developed sophistication, Cases 9 and 10, are both evaluated as relational; in addition, they also rank relatively high in relevance. Case 9 can be verbalized as the sentences: *'Algorithm comprises separate commands that form a function which manipulates data structures. The implementation follows heuristics exemplified by divide-and-conquer and recursion. The paradigm influences how the implementation can be done. The efficiency of the implementation has to be measured (O-notation), there are the best and worst cases.'* The narrative is whole and consistent. If any defects have to be named, the map does not itemize any key algorithms, such as well-known searches and sorts. Case 10 narrates a bit more 'algorithmic-math'-biased story that highlights logic: *'Algorithms exploit logic, e.g., Boolean algebra, where De Morgan laws belong as well. R&D utilizes logic and algorithms; SW implements data structures and algorithms exemplified by sorts and finds that can be recursive.'* However, the narrative also introduces some foreign elements, such as R&D and SW that are not necessary, which bears some resemblance to Case 2 and its actors. Linkages are searched and found, and the narrative is whole, however, the absence of efficiency and



Case 1:function algebra number probability

Case2:number trigonometry algebra geometry

Case3: algebra graph geometry

Case 4: graph probability geometry graph.equation trigonometry factorize algebra sequence

Case 5: graph geometry number linear and quadratic equation fraction algebra trigonometry

Case 6: algebra number geometry

Case 7: quadratic equation Pythagorean theorem linear equation parabola percentage

Case 8: algebra graph number

Case9: geometry trigonometry number algebra

Case 10: number trigonometry algebra sequence geometry statistics probability measurement func.

Case 11: geometry trigonometry statistics quadratics func. probs. algebra measurement number fraction decimal percentage sequence

Case 12: number algebra probability fraction geometry trigonometry co-ordinate seq. graph

Figure 1. The mind/concept maps of mathematics drawn by the Year 8 students.

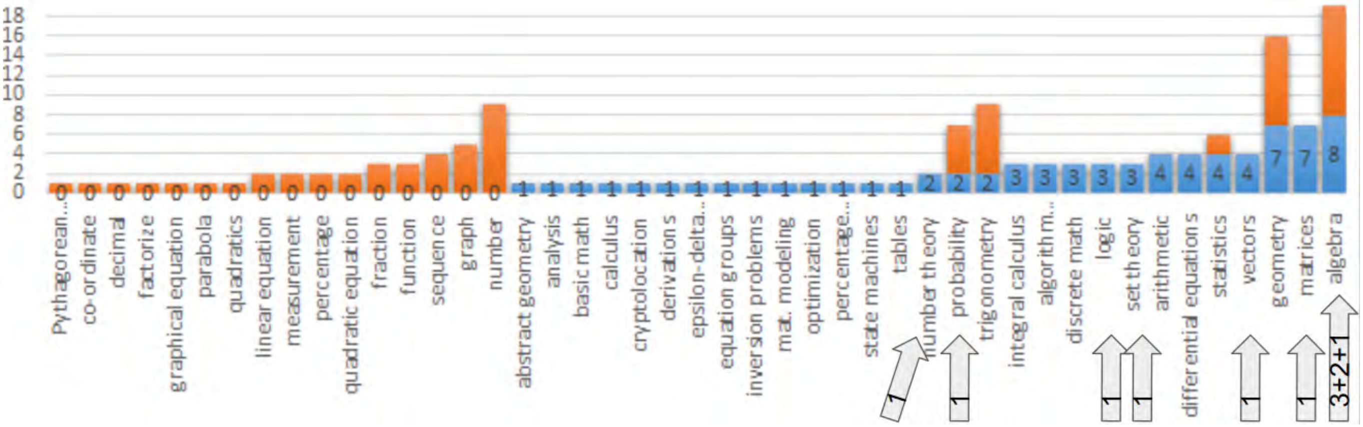


Figure 2. The mathematics syllabus areas identified by Year 8 in orange; and named by the university students in blue. The arrows indicate where the latter would position algorithms. Two of the interviewees could not name any specific syllabus area.

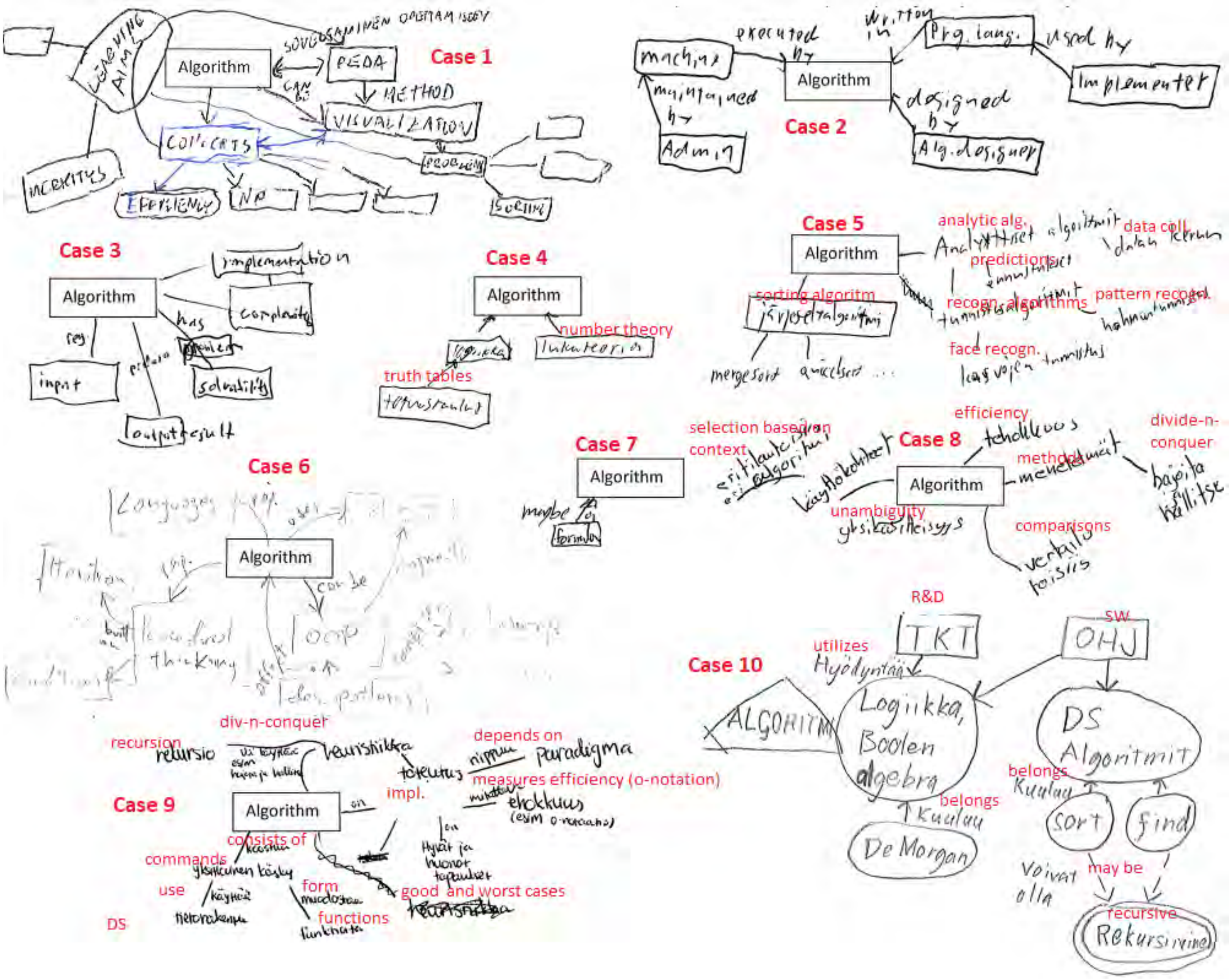


Figure 3. The higher education interviewees' perceptions of algorithms

Table 1. Maps evaluated based on the SOLO taxonomy

		SOLO levels/cases				
Topic	**Edu-level**	pre-	uni-	multistructural	relational	ext.abstract
Math	Year 8		5,7,8,10,11	1,2,3,4,6,9,12		
Algorithm	Higher education		3,7	2,4,8	1,5,6,9,10	

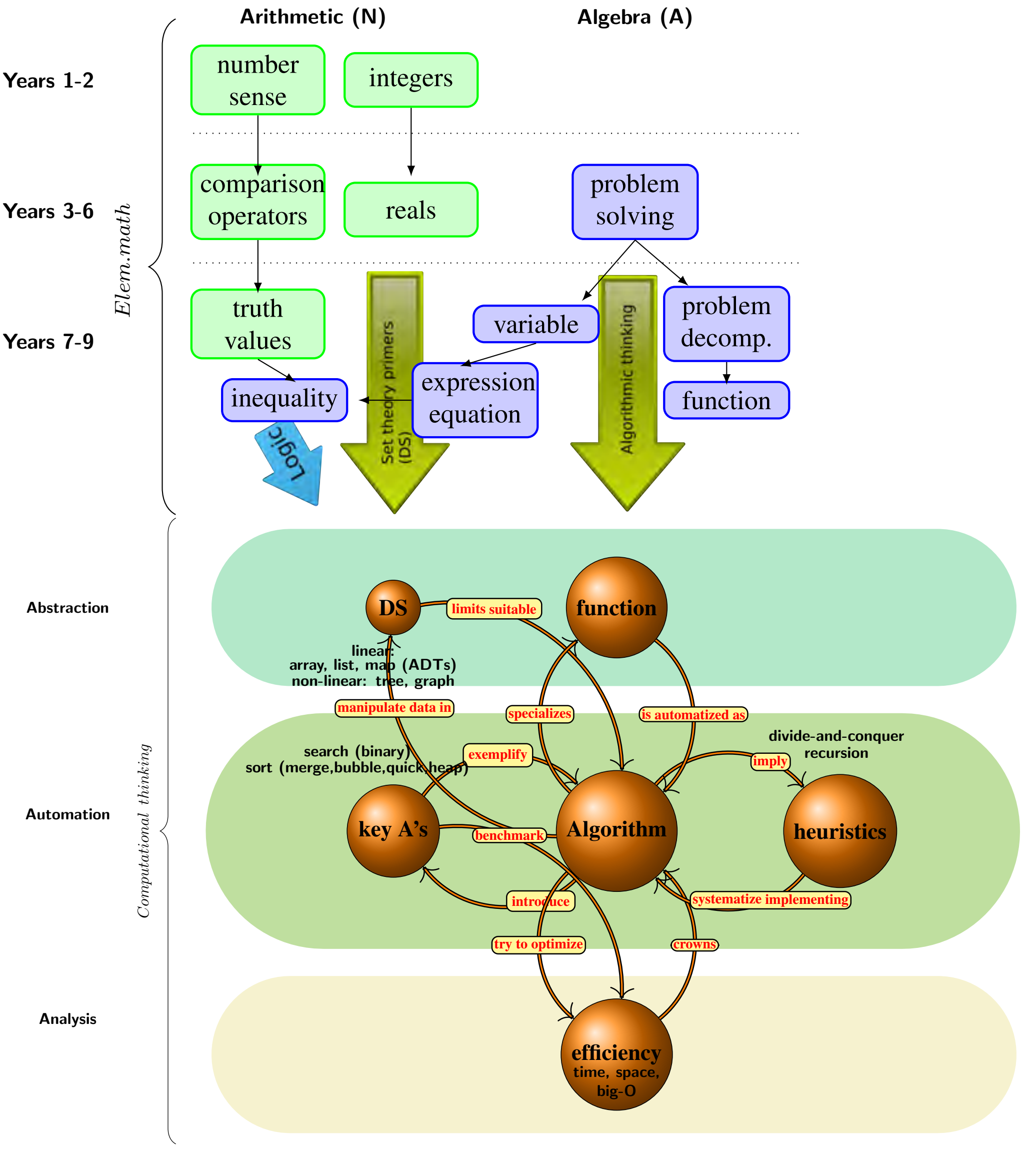


Figure 4. Learning progressions from the FNC-2014 elementary mathematics to algorithmic thinking layered into abstraction, automation, and analysis as suggested in Niemelä et al. [22]

heuristics are the downsides.

To conclude, the quality of the algorithm maps is not too impressive. As interviewers, we take our share of the blame: before drawing, the basics of concept mapping should have been introduced in a more detail. As an additional question type, concept mapping could enrich evaluation that is mainly based on verbal questions. However, in this data, the quality of verbal description and map-based representation contradict in some cases; concept mapping seems not that familiar technique for the participants, or the motivation for drawing is low. However, as part of 'learning to learn', it is beneficial to be aware of different methods in order to know which one to apply and which one is well suited to the respective context.

4 Discussion

In this chapter, we evaluate concept mapping as a learning method, after which the composite map of algorithms is drawn and linked to FNC-2014 mathematics syllabus. The feedback from the students at both education levels was mainly positive. Some quotations selected, Year 8 starts: *'It was helpful because it was visual and it laid out the information nicely.', It was an interesting way of connecting concepts. It is a creative way of showing how everything in math is related and how it all mixes*, and *'I found it interesting and kind of beneficial seeking how the topics and ideas in maths link together. Yep, it was great opportunity to do this stuff. It helped me a lot :) '*

Concept mapping is greeted as a useful visual means to organize one's thoughts to get a better overview. In the Year 8 experiment, these were the only mildly negative comments in addition to some critics regarding the visual look, such as, too much info, and no resting space.

The higher education participants did not illuminate fun and creativeness, but the usefulness of mapping in organizing one's thoughts. In overall, the benefit of multiple representations was emphasized, such as transforming the text as maps and maps back to text, and utilizing bullet point lists, for example: *'I have used concept mapping, but according to my students they are not that useful, if not written out as a text as well.' 'I do not normally draw maps, but bullet point lists instead, that I will complement during the go. On the other hand, I benefit from the visual representations done by others, even if I don't make them by myself.'*, and last, *'I utilized it myself in high school for writing essays: first, free-form boxes and then writing the bullet points. '*

Mapping was mainly seen as a revision tool, exploitable after studying the content. Some proposed a revision app: a click would reveal corresponding exercise types of the area. In addition, comparisons between students' maps were anticipated to be educative.

Most often, the higher education interviewees would integrate algorithms into algebra, logic, or set theory. However, we omit such application specific areas as number theory and linear algebra for the present, as they do not belong to the conceptual core of algorithms. In most of the maps, we spot data structures (DS, that has a connection to set theory), key algorithms such as searches and sorts (merge- and quick-sort mentioned most often), and heuristics. Heuristics systematize the algorithm development, divide-and-conquer was common in comments, also recursion and recursive methods were representative in general.

Fig. 4 divides the subtopics of algorithm in the layers of abstraction, automation, and analysis that together constitute the bigger entity of computational thinking [22]. Next, we deliberate draw linkages between algorithms and the FNC-2014 contents of algebra, set theory primers (as preparations for data structures), and logic. Even if these contents might be enough for elementary-level students, their teachers must be more widely knowledgeable concerning computational theories, and the same applies to higher education people as well.

The brief scan through corresponding ACM courses reveals the absence of such theoretical topics as 'computability' (Turing machine, halting) and 'proving' (e.g., loop invariant and recursion with induction-like proofs) [23]. In contrast, the higher education participants (Case 1, 3, 4, 5, and 10) suggest more practical topics to substitute theory, such as applying algorithms in the domains of machine learning and image and video processing. In these domains, data is often handled as matrices. Thus, linear algebra starts to manifest itself in practically-oriented education as an indispensable support for algorithms.

5 Conclusions

RQ1: Which mathematics syllabus areas do the elementary level students and higher education participants highlight? Year 8 highlights numbers (arithmetic), algebra and geometry. The higher education interviewees emphasize linear algebra as well (matrices/vectors), and other advanced discrete and continuous mathematics topics.

RQ2: Where would the interviewees situate algorithms in the mathematics syllabus? To algebra, single votes point out logic, set theory, number theory, and probability.

RQ3: Which kind of 'crowd-sourced' algorithm concept map can be constructed and how could it be aligned with FNC-2014 mathematics syllabus? As the subtopic, the map contains data structures, key algorithms, heuristics, and efficiency considerations. Algebra, functions and equations are thought to provide exploitable affordances in introducing algorithms. Data structures are prompted by number zones and variables leading to the concept of type first; logic by inequalities and truth-values, however, logic in overall should be strengthened to give a better support for programming.

Future studies should focus on learning outcomes with concept mapping in mathematics. Furthermore, a proper review of the crowd-sourced algorithm concept map should examine closer ACM algorithm courses and their subtopics, application areas and exercises in order to capture the absence of any other necessary subtopics in the map in addition to those ones identified already, i.e., computing theory and proving.

Acknowledgments

Thanks to the Academy of Finland (grant number 303694; *Skills, education and the future of work*) for their financial support.

REFERENCES

[1] Piaget J, Duckworth E. Genetic epistemology. American Behavioral Scientist. 1970;13(3):459–480.

[2] Derry SJ. Cognitive schema theory in the constructivist debate. Educational Psychologist. 1996;31(3-4):163–174.

[3] Von Glasersfeld E. Radical Constructivism: A Way of Knowing and Learning. Studies in Mathematics Education Series: 6. ERIC; 1995.

[4] Von Glasersfeld E. An exposition of constructivism: Why some like it radical. In: Facets of systems science. Springer; 1991. p. 229–238.

[5] Finnish National Board of Education. Finnish National Curriculum 2004; 2004.

[6] Finnish National Board of Education. Finnish National Curriculum 2014; 2014.

[7] Ausubel DP, Fitzgerald D. Organizer, general background, and antecedent learning variables in sequential verbal learning. Journal of educational psychology. 1962;53(6):243.

[8] Mayer RE, Bove W, Bryman A, Mars R, Tapangco L. When less is more: Meaningful learning from visual and verbal summaries of science textbook lessons. Journal of educational psychology. 1996;88(1):64.

[9] Novak JD, Gowin DB. Learning how to learn. Cambridge University Press; 1984.

[10] Novak JD, Cañas AJ. The theory underlying concept maps and how to construct and use them. Institute for Human and Machine Cognition; 2008.

[11] Kinchin IM. Novakian concept mapping in university and professional education. Knowledge Management & E-Learning: An International Journal (KM&EL). 2015;7(1):1–5.

[12] Buzan T, Buzan B. The mind map book: How to use radiant thinking to maximize your brain's untapped potential. Plume New York; 1996.

[13] Williams CG. Using concept maps to assess conceptual knowledge of function. Journal for Research in Mathematics Education. 1998;p. 414–421.

[14] Hay D, Kinchin I, Lygo-Baker S. Making learning visible: the role of concept mapping in higher education. Studies in higher education. 2008;33(3):295–311.

[15] Royer JM, Cisero CA, Carlo MS. Techniques and procedures for assessing cognitive skills. Review of Educational Research. 1993;63(2):201–243.

[16] Gilbert J, K R Justi, Ferreira P. Modelling, visualization, and the development of an understanding of the levels of representation in chemical education. Paper presented in ESERA 2007 Conference August 21 - 25 at Malmö University, Malmö Sweden; 2007. .

[17] Nersessian NJ. Mental modeling in conceptual change. In: Stella V, editor. International handbook of research on conceptual change. Routledge; 2008. p. 391–416.

[18] Biggs JB, Collis KF. Evaluation the quality of learning: the SOLO taxonomy (structure of the observed learning outcome). Academic Press; 1982.

[19] Biggs J, Collis KF. SOLO taxonomy. Education News. 1980;17(5):19–23.

[20] Sheard J, Carbone A, Lister R, Simon B, Thompson E, Whalley JL. Going SOLO to assess novice programmers. In: Acm sigcse bulletin. vol. 40. ACM; 2008. p. 209–213.

[21] Biggs J. Assessing for Learning: Some Dimensions Underlying New Approaches to Educational Assessment. Alberta Journal of Educational Research. 1995;41(1):1–17.

[22] Niemelä P, Partanen T, Harsu M, Leppänen L, Ihantola P. Computational Thinking as an Emergent Learning Trajectory of Mathematics. In: Koli Calling International Conference on Computing Education Research. vol. 17; 2017. .

[23] ACM&IEEE. Computer Science Curricula 2013: Curriculum Guidelines for Undergraduate Degree Programs in Computer Science, December 20, 2013; 2013. Available from: `https://www.acm.org/binaries/content/assets/education/cs2013_web_final.pdf`.