



HAL
open science

Graph problems as a means for accessing the abstraction skills

Janka Medová, Gregor Milicic, Matthias Ludwig

► **To cite this version:**

Janka Medová, Gregor Milicic, Matthias Ludwig. Graph problems as a means for accessing the abstraction skills. Twelfth Congress of the European Society for Research in Mathematics Education (CERME12), Feb 2022, Bozen-Bolzano, Italy. hal-03748478

HAL Id: hal-03748478

<https://hal.science/hal-03748478>

Submitted on 9 Aug 2022

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Graph problems as a means for accessing the abstraction skills

Janka Medová¹, Gregor Milicic² and Matthias Ludwig²

¹Constantine the Philosopher University in Nitra, Faculty of Natural Sciences and Informatics, Tr. A. Hlinku 1, 949 74 Nitra, Slovakia; jmedova@ukf.sk

²Goethe University Frankfurt, Institut für Didaktik der Mathematik und der Informatik, Robert-Mayer-Str. 6-8, 60325 Frankfurt, Germany; ludwig@math.uni-frankfurt.de

Computational thinking is an important 21st century skill. The ability to design own algorithm, i.e. algorithmic thinking is its integral part. Graph algorithms seem to be a promising mathematical content contributing to development of algorithmic thinking. However, in order to apply the corresponding skills to the problem at hand, first a corresponding representations has to be found. This step of abstraction is crucial for the application of skills to unknown situations and can be seen as a prerequisite for the algorithmic thinking. Solutions of the representation problem of 58 undergraduate students were analysed. Most students chose the diagram as a representation of the situation, only three students used the adjacency matrix and no students chose the incidence matrix or adjacency list, the other known representations. This may indicate that more activities are needed for enhancing students' ability to represent the graph, either by matrices or by a diagram.

Keywords: Algorithmic thinking, computational thinking, abstraction, graphs, graph theory.

Introduction

With the continuous digitalization of our daily life the development of corresponding digital competencies is crucial for future generations. The importance of “[...] 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” (Wing, 2017, pp. 8) is more and more recognized. Thus the corresponding competencies are increasingly integrated into the current curricula referred under the term of Computational Thinking (CT) (Bocconi et al., 2016) including concepts such as logical reasoning, abstraction, decomposition, generalisation or algorithmic thinking, understood as “[...] the ability to think in terms of sequences and rules as a way of solving problems or understanding situations.” (Csizmadia et al., 2015, p. 7).

The abstraction is the one crucial aspects of the CT describing the process of “... reducing the unnecessary detail” of a situation, problem, or artefact (Bocconi et al., 2016, pp. 18). Certain proficiency in abstraction is a crucial aspect during the process of problem solving, for mathematics as well as for computer science (Ferrari, 2003). Without the ability to simplify a given problem or situation in the process of abstraction resulting to a corresponding mathematical or computational representation, the remaining problem-solving process is bound to fail, nor can other maybe suitable solving strategies or algorithms be applied, as the corresponding requirements cannot be checked. This may lead to an unfeasible choice of problem-solving process for the given problem, resulting inevitably in an incorrect solution.

Graph problems offer a good starting point to address and foster abstraction (Milicic et al. 2021) as some applications of graph algorithms are easy to be explained and offer the suitable environment

for mathematical investigations with low demands on previous knowledge, e.g. constructing Eulerian trail (Geschke et al., 2005), minimum spanning tree (Vidermanová & Melušová, 2011) or estimating chromatic number of graph (Prayitno et al., 2022). In order to obtain a graph from the application or a problem from the real world, firstly the redundant and unnecessary information have to be identified and removed. Redundant information are details of the given problem not contributing in any way to the problem-solving process and can therefore be omitted. Unnecessary information are details not related to the problem which should be solved. Graph problems offer therefore many possibilities to address and foster abstraction as the first step and following, algorithmic thinking subsequently. Despite existing algorithmic thinking skills, the process of abstraction can still be challenging for students (Wetzel et al., 2020), emphasizing the importance of addressing especially this aspect of CT in the school environment.

Like in other fields of mathematics, e.g., arithmetic or algebra, using multiple different representations and linking between them can enhance conceptual understanding of given topic (Hodnik Čadež, 2018; Griffin, 2004). Moreover, the graph representations have to be handled differently while implementing graph algorithms in programming languages and also lead to different space complexity of the produced instantiations.

The earlier findings (e.g. Hazzan & Hadar, 2005) shown that students often overestimate the value of diagrammatic representation of graph. This results to decrease of the level of abstraction while dealing with graph theory concepts and may cause the difficulties related to the recognition of the details of graph algorithms (Dagdilelis & Satratzemi, 1998). However, the recent study of Prayitno et al. (2022) reveals that students are able to come up with both, diagrammatic and matrix representation of the graph while solving novel problems stemming from graph theory and both types of representation can lead to correct solutions of the problem in algorithmic graph theory.

In this paper we present the initial results of an exploratory study with university students. It was conducted in order to identify any possible challenges students face when they are asked to abstract a given graph problem and condense the corresponding information in the suitable representation. As graphs have different representations, it is not clear which of them is preferred by students. We formulated the following three research questions:

1. To what extent are the undergraduate students able to use abstraction in order to represent the relation given by the computer model by the means of graph theory?
2. What representations do they prefer?
3. What kinds of mistake can occur there?

Methodology

To conduct this research, we analysed the submitted solutions of the 58 undergraduate students. All of them were in year 1 of the bachelor study programme Applied informatics and took the paper and pencil test focused on algorithmic graph theory as a part of assessment of the course Discrete mathematics 2 focused on combinatorics and graph theory. The test itself consisted of 6 problems. One of them is analysed in further details in this paper. The written informed consent was requested and collected after the students got their whole evaluation and passed/failed the course. The all of these 58 students attempted to solve the problem and provided their consent.

We expected that the participants are equipped with a decent knowledge of programming and therefore possess adequate algorithmic thinking skills, as they have passed the introductory course in programming in the previous semester, covering work with variables including arrays, sequences, conditionals, loops with numbered (for loops) and conditional repetition (while loops) of instructions. The test was administered at the end of the second semester of their study before the exam in regular programming course comprising the use of procedures, recursion, dynamic variables (FIFO/LIFO structures) and object-oriented programming. The different representations of graph, diagrams, incidence matrix, adjacency matrix and adjacency list were integral part of the discrete mathematics course.

The task of finding a graph that represents which area shares an edge with each other in the two dimensions is a frequently occurring problem when teaching and learning graph problems. Using the means of augmented reality (Buchner, 2018), we extended this problem by another dimension by not using planar areas, but some three-dimensional objects, see Figure 1a and 1b as one of the three different situations used in the exam.

We derived this task using the elements from the SOMA cube puzzle (Peter-Orth, 1985). The task was formulated as follows: “Represent the adjacency of the all parts of the kit (consisting of the 3 or 4 cubes) from which the ‘sofa’ is assembled. The tiny gap between the parts is only for better clarity.” The students were asked to scan a respective QR code and open a webpage inside a browser on their mobile device or with their web camera. No additional hard- or software aids were thus necessary in order to solve the task. By pointing the camera at the marker, the object as seen in Figure 1a and 1b was visible using Augmented Reality (AR). The students could turn the object around on their screen and shrink or enlarge it using common gestures with their fingers on the screen¹. A solution using a diagram as representation is given in Figure 1c.

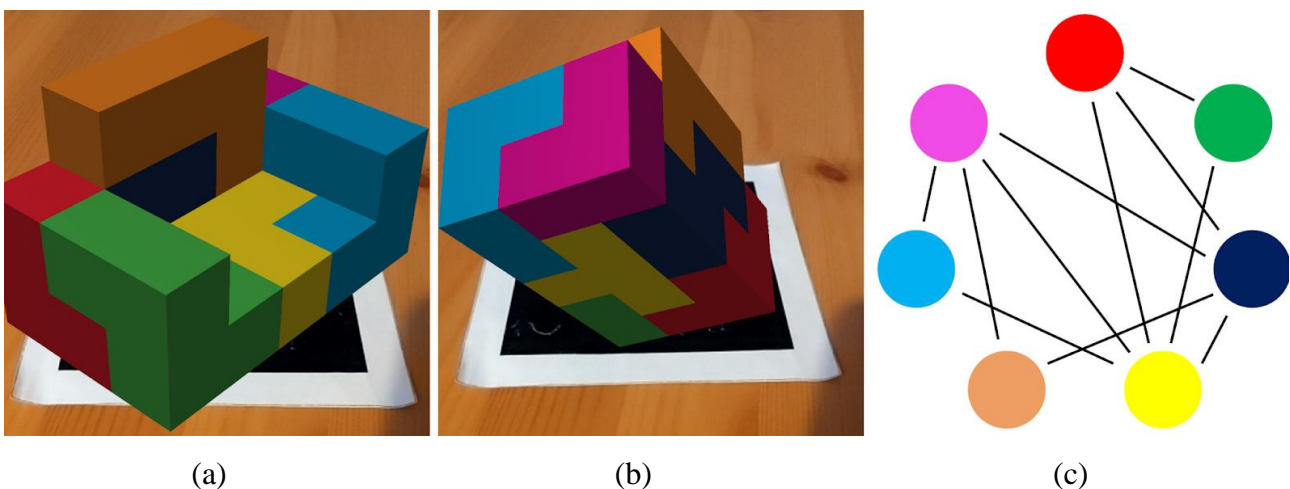


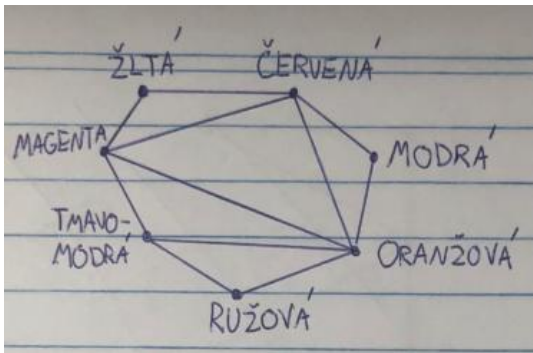
Figure 1: Two views of the sofa consisting of SOMA cube elements (a and b) and diagram of its graph representation (c)

¹ See <https://colette-project.eu/AR/somas.html> for the used AR-marker as well as the presented setting (Soma Sofa 1).

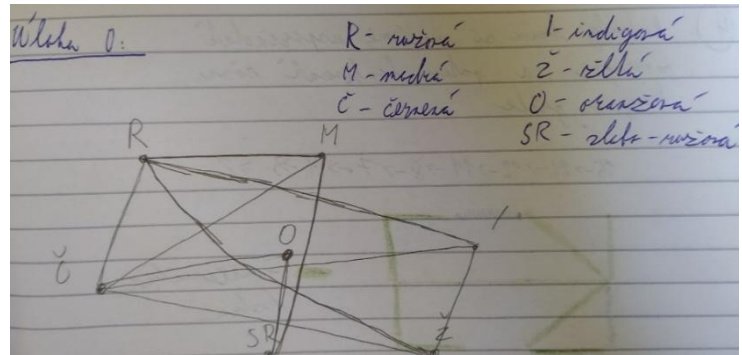
Findings

Out of the 58 solutions submitted by the students, 45, i.e., 77.6 % were correct. Although the students were not instructed which representation they should use, only the 3 students used the form of adjacency matrix, one of them used the adjacency matrix and the diagram, while other used the representation of the graph by a diagram. No students used incidence matrix or adjacency list. This may result from the dominant use of the diagram representation of graph during the lectures and problem-solving sessions throughout the whole semester.

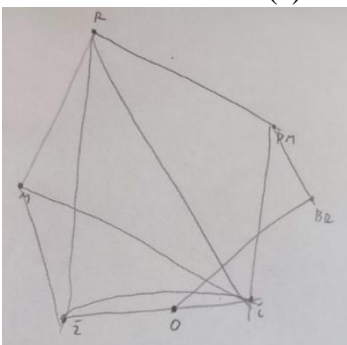
Different names of vertices were used among the correct solutions. Only one student labelled the vertices as v_1, v_2 , etc. and on top he provided a table which label means the solid of what colour. Some students used the full colour names (Figure 2a), some just the abbreviations (Figure 2b) and some also included a list of abbreviations (Figures 2c and 2d). Most solutions have the vertices placed in shape of regular heptagon (Figures 2a and 2b), only occasionally the vertices were placed in different configurations (Figure 2c). One student also used the colours and drew the bottom and side view (Figure 2d) on top of the graph representation.



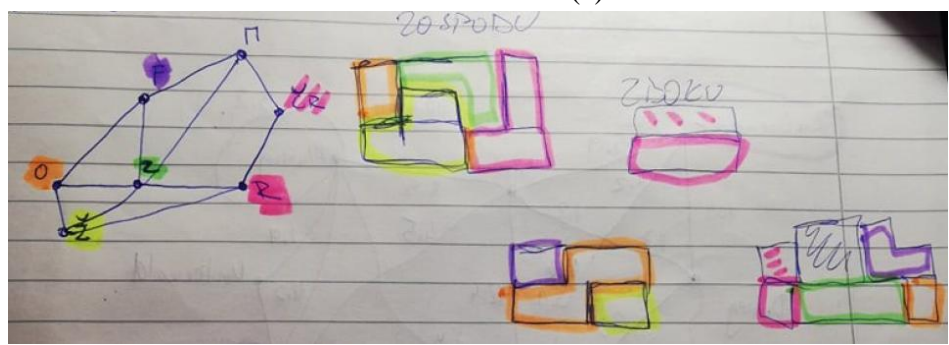
(a)



(c)



(b)



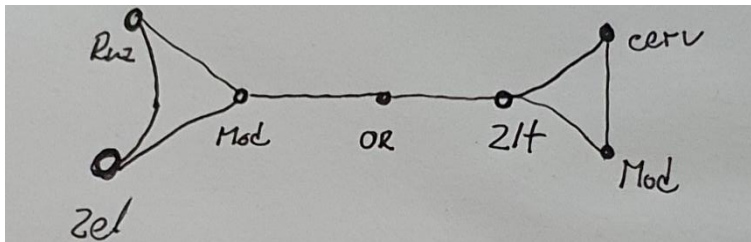
(d)

Figure 2: Students' solutions with different labelling of vertices

Ružová = pink; modrá = blue; červená = red; indigová = indigo; žltá = yellow; oranžová = orange; slaboružová = light pink; tmavo-modrá = dark blue. BR probably states for bledoružová = light pink and BM for bledomodrá = light blue. Zospodu = bottom view; z боку = side view.

The majority of incorrect solutions can be considered as flips when one or more edges were missing. The solution in Figure 3a is an example of solution where quite a lot of edges were missing. Furthermore, the student did not distinguish between the two shades of blue. On the other

hand, the solution in Figure 3b represents a solution with excess edges. In addition, the student considered the relation of being adjacent as reflexive.



(a)

Uloha 1

tabuľka súvislosti

	V1	V2	V3	V4	V5	V6	V7
V1	1	1	1	1	0	1	1
V2	1	1	1	0	1	0	1
V3	1	1	1	1	0	0	1
V4	1	0	1	1	1	1	1
V5	0	1	0	1	1	1	1
V6	1	0	0	1	1	1	1
V7	1	1	1	1	1	1	1

V1 - červená
 V2 - zelená
 V3 - modrá
 V4 - žltá
 V5 - tmavomodrá
 V6 - šedá
 V7 - fialová

(b)

Figure 3: Examples of incorrect students' solutions

On top on typical solutions in Figure 2, there were several solutions demonstrating unique approach to the representation. For instance, two students considered the small unit cubes instead of the tricubes and tetracubes as vertices (Figure 4). One of them (solution in Figure 4a) further concluded that the graph is not connected as he considered the tiny gaps between the parts of the given problem as separating. The second unique student's solution (Figure 4b) is a correct one for this choice of representation. The corrections made by student demonstrate that it was quite challenging to keep tracking of so many (27) vertices.

The matrix representation of graph was preferred by only three of 58 participating students, two of them were correct. The incorrect representation by adjacency matrix (Figure 3b) was the only case when student used two different representations. Even though the diagram fits the matrix, the represented graph does not fit the adjacency of part of the SOMA 'sofa'.

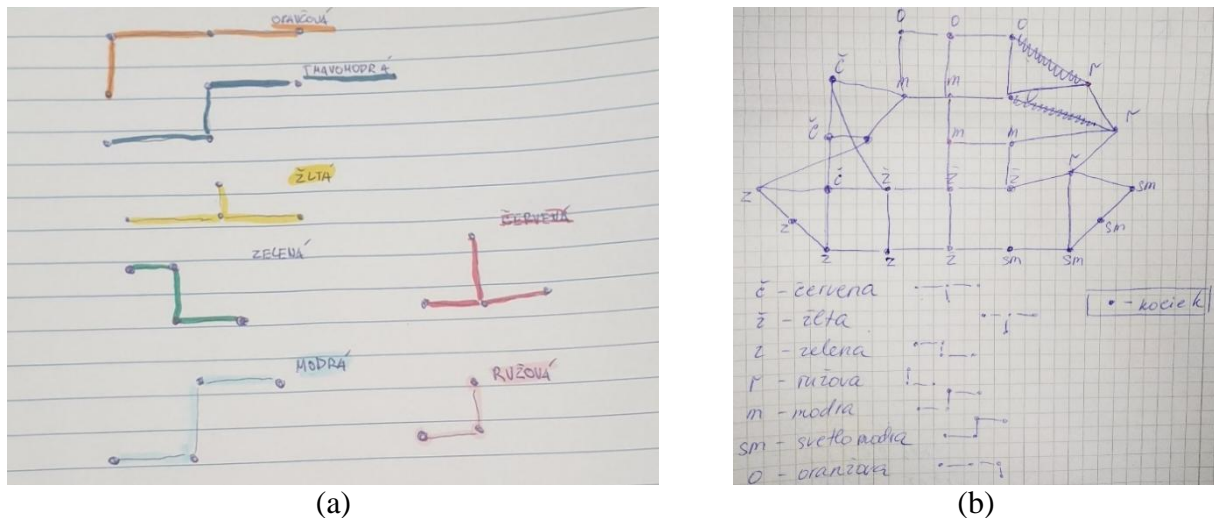


Figure 4: Solution of students' representing the cubes as the vertices

Discussion

The rate of correct solutions was much lower than expected in case of the students who passed the introductory course of programming and the course focused on the graph theory specifically. The observed success-rate was even lower than the success-rate of a similar group of students solving modelling problems in algorithmic graph theory (Medová et al., 2019). One of the possible explanations vests in that during the course and in ample cases of textbooks the problem situation is already given by diagram representation of graph and the abstraction job as such is already done by the author of the task (Fojtík, 2021). Also, pre-prepared 'suitable illustrative graphs ... using colours' are often considered an effective mean 'to emphasize the characteristics of the concepts' (Milková, 2009). However, the informed choice of suitable representation is one of the crucial aspects when tackling unstructured open problems (Swan & Burkhardt, 2014) particularly in algorithmics, and influenced by the level of abstraction of the solver. Therefore, the process of abstraction itself, regardless the type of representation, is essential and crucial for solving problems

in the situations requiring to omit the unnecessary and redundant information (Wetzel et al., 2020), most frequently in cases of real-life applications. The students should estimate the necessity of the information based on the definition of the particular problem.

Our findings suggest that the students used mostly the diagrammatic representation of graphs. It complies with the observation by Hazzan and Hadar (2005) than students tend to over emphasize the visual aspect of graph theory. The students may use the diagrammatic representation because the problem was not set to any complex problem situation, because, as stated by González et al. (2021) even the students with low level of reasoning in graph theory should be aware of limitations of the different known representations for different purposes. On the other hand, as we stated in our previous work (Milicic et al., 2021) any different representations of graphs can lead to different variables for representing the graph in the computer and it may be easier to analyse or use for subsequent tasks and solving processes. Adjacency and incidence matrices can directly lead to use of arrays and adjacency list is just a small step to the FIFO list. In contrast to diagram, the representations of graph by matrix and list permit to get the information about the adjacency of two vertices in constant time but the instantiations vary in space complexity. Even though the diagrammatic representation cannot be used for computer processing, it seems to be suitable for students while learning the principles of algorithms (Melušová & Vidermanová, 2011).

Conclusions

The tasks to represent the relations between the parts of geometric shapes are often used in graph theory education. The technology of augmented reality enables us to use some three-dimensional geometric shapes instead of usual planar problems without any additional hard- or software aids. The task to omit the redundant information and come up with an abstract representation caused some difficulties even in case of students equipped with their decent programming capability and experience. It seems that more attention should be paid to different representations of graph during the course, particularly to creating the representation by students instead of pre-prepared graphs.

However, the extent to which is the level of abstraction connected to the ability to solve (modelling) problems in algorithmic graph theory is still to be investigated and estimated.

Acknowledgment

The project is (partially) funded by the ERASMUS+ grant program of the European Union under grant no. 2020-1-DE03-KA201-077363 and by Slovak Research and Development Agency, grant no. APVV-20-0599. Neither the European Commission nor the project's national funding agency PAD are responsible for the content or liable for any losses or damage resulting of the use of these resources.

References

Bocconi, S., Chiocciariello, A., Dettori, G., Ferrari, A., & Engelhardt, K. (2016). *Developing computational thinking in compulsory education. Implications for policy and practice*. European Commission - Scientific and Technical Research Reports. <https://doi.org/10.2791/792158>

- Buchner, J. (2018). *Real - nur besser. Augmented Reality für individualisiertes Lehren und personalisiertes Lernen*. Computer+Unterricht.
- Dagdilelis, V., & Satratzemi, M. (1998). DIDAGRAPH: software for teaching graph theory algorithms. *ACM SIGCSE Bulletin*, 30, 64–68. <https://doi.org/10.1145/290320.283024>
- Ferrari, P. L. (2003). Abstraction in mathematics. *Philosophical Transactions: Biological Sciences*, 358(1435), 1225–1230. <https://doi.org/10.1098/rstb.2003.1316>
- Geschke, A., Kortenkamp, U., Lutz-Westphal, B., & Materlik, D. (2005). Visage — Visualization of algorithms in discrete mathematics. *ZDM Mathematics Education*, 37(5), 395–401. <https://doi.org/10.1007/s11858-005-0027-z>
- González, A., Gallego-Sánchez, I., Gavilán-Izquierdo, J. M., & Puertas, M. L. (2021). Characterizing levels of reasoning in graph theory. *Eurasia Journal of Mathematics, Science and Technology Education*, 17(8), em1990. <https://doi.org/10.29333/ejmste/11020>
- Griffin, S. (2004). Building number sense with Number Worlds: a mathematics program for young children. *Early Childhood Research Quarterly*, 19(1), 173–180. <https://doi.org/10.1016/j.ecresq.2004.01.012>
- Ferrari, P. L. (2003). Abstraction in mathematics. *Philosophical Transactions: Biological Sciences*, 358(1435), 1225–1230. <https://dx.doi.org/10.1098%2Frstb.2003.1316>
- Fojtík, M. (2021). Using GeoGebra in graph theory. In Kortesi, P. et al. (Eds.) *Proceedings of the GeoGebra Summit 2021 - CEEPUS Summer University in Miskolc, 28 June - 9 July 2021*. 56–68.
- Hazzan, O., & Hadar, I. (2005). Reducing abstraction when learning graph theory. *Journal of Computers in Mathematics and Science Teaching*, 24(3), 255–272. <https://www.learntechlib.org/primary/p/5942/>
- Hodnik Čadež, T. (2018). Exploring processes in constructing mathematical concepts and reasoning through linking representations. *CEPS journal*, 8(2), 5–8. <https://doi.org/10.26529/cepsj.545>
- Milicic, G., van Borkulo, S.P., Medová, J., Wetzel, S. & Ludwig, M. (2021). Design and development of a learning environment for computational thinking: The Erasmus+ Project, *EDULEARN21 Proceedings*, 7376–7383. <https://doi.org/10.21125/edulearn.2021.1495>
- Milková, E. (2009). Constructing Knowledge in Graph Theory and Combinatorial Optimization. *WSEAS Transactions on Mathematics*, 8(8), 424–434. <https://doi.org/10.5555/1639374.1639380>
- Medová, J., Páleníková, K., Rybanský, L., & Naštická, Z. (2019). Undergraduate Students' Solutions of Modeling Problems in Algorithmic Graph Theory. *Mathematics*, 7(7), 572, 1–16. <https://doi.org/10.3390/math7070572>
- Peter-Orth, C. (1985). All solutions of the Soma cube puzzle. *Discrete Mathematics*, 57(1), 105–121. [https://doi.org/10.1016/0012-365X\(85\)90160-8](https://doi.org/10.1016/0012-365X(85)90160-8)

- Prayitno, A. T., Nusantara, T., Hidayanto, E., & Rahardjo, S. (2022). Identification of graph thinking in solving mathematical problems naturally. *Participatory Educational Research*, 9(2), 118–135. <https://doi.org/10.17275/per.22.32.9.2>
- Swan, M. & Burkhardt, H. (2012). Designing assessment of performance in mathematics. *Educational Designer: Journal of the International Society for Design and Development in Education*, 2(7), 1–41. <https://www.educationaldesigner.org/ed/volume2/issue7/article24/>
- Vidermanová, K. & Melušová, J. (2011). Teaching Graph Theory with Cinderella and Visage: an undergraduate case. *New Trends in Mathematics Education: DGS in education*, Constantine the Philosopher University in Nitra, 79–85.
- Wetzel, S., Milicic, G. & Ludwig, M. (2020). Gifted Students' Use of Computational Thinking Skills Approaching a Graph Problem: A case study, *EDULEARN20 Proceedings*, 6936–6944, 2020. <http://dx.doi.org/10.21125/edulearn.2020.1797>
- Wing, J. (2017). Computational thinking's influence on research and education for all. *Italian Journal of Educational Technology*, 25(2), 7–14. <https://doi.org/10.17471/2499-4324/922>