THE SIGNIFICANCE OF INVESTIGATING THE RELATIONSHIP BETWEEN MATHEMATICAL THINKING AND COMPUTATIONAL THINKING USING LINGUISTIC ASPECTS

https://doi.org/10.47743/jopafl-2023-30-11

KRAMER Elena
Alexandru Ioan Cuza University of Iaşi, Faculty of Economics and Business Administration
Iaşi, Romania
elenak@braude.ac.il

GEORGESCU Mircea
Alexandru Ioan Cuza University of Iaşi, Faculty of Economics and Business Administration
Iaşi, Romania
mirceag@uaic.ro

Abstract: Computational thinking, defined as a way of thinking that can be applied to various fields that require problem-solving skills, has become prevalent in education. Students, i.e., future specialists, have to be prepared for complex thinking competence, necessary for solving business and societal problems, for which a combination of mathematical thinking and computational thinking is essential. The preliminary premise is that there is a correlation between ability in specific mathematical and computational fields. Therefore, this paper aims to highlight the significance of investigating the relations between those fields from linguistic point of view. In order to better understand the relationship between abilities in specific mathematical and computational fields, this paper presents an analysis of a new approach, namely, developing hypotheses for exploring the relationship between metalanguages of different fields of Mathematics and Computer Science. Additionally, the paper describes the first stage of a study on a doctoral level in an attempt to suggest possible statistical analyses suitable for testing hypotheses based on meta-analysis of the current literature.
Keywords: mathematical thinking, computational thinking, undergraduate students, education skills, metalanguage.
JFL Classification: C1, Y90

This Article was presented as a paper at the 15th edition of the Annual International Conference Globalization and Higher Education in Economics and Business Administration (GEBA 2023), which was held at the Alexandru Ioan Cuza University, Faculty of Economics and Business Administration in Iasi, Romania from the 19-21 October 2023.

Introduction and historical background

Computers and programming have revolutionized the world and have promoted technology literacy as a crucial skill to achieve academic and career success in the digital 21st century (Shute et al., 2017). Accordingly, Computational thinking (CT), a way of thinking that can be applied to various fields that need problem-solving skills, has become widely used in

education. Students, who are future specialists, should be prepared for engaging in complex thinking, required for the solution of business and societal problems. Thus, it is necessary to apply a combination of mathematical thinking (MT) and computational thinking (CT).
The field of computer science is relatively young, compared to the area of mathematics. However, what was the starting point? Analytic philosophy (Glock, 2013) began being popular on the verge of the 20th century. The term analytical indicates the ideals of clarity, accuracy and logical rigor of thinking that representatives of this area of philosophy strive to implement. Analytical philosophy aims to formalize knowledge in the fields of humanities and mathematics (formalization on some axiomatic basis). Many philosophers have conceived the concept of analytic philosophy (among them, Bertrand Russell, Alfred North Whitehead, Ludwig Wittgenstein, Gottlob Frege, etc.). Yes, as for programming, the most fundamental work is Principia Mathematica (often abbreviated PM). This is a three-volume work about the foundations of mathematics, written by the mathematician–philosophers Alfred North Whitehead and Bertrand Russell and published in 1910, 1912, and 1913 (Whitehead and Russell., 1997). It developed the Type Theory, the concept of types in programming today (Russell., 1908). Based on the formalization process of mathematics and the process of constructing formal calculus, such as mathematical logic, the Turing machine and lambda calculus concept were developed in 1936 (Petzold., 2008). Alan Turing (Petzold, 2008) responded to the question of what an algorithm is and what can be calculated at all. He built the abstract machine, so everything that could be calculated on this machine was defined as an algorithm. The concept of Turing machine and lambda calculus have attributed formal definitions to two types of programming languages that exist today: Functional Programming (programming languages like Lisp, Haskell) and Imperative Programming (programming languages like C, C++, Java) (Alegre and Moreno, 2015; Frame and Coffey, 2014). The first imperative programming language (Fortran) was originally developed in the 1950s, and the first functional programming language (Lisp) was developed in the 1960s.

A Short background of Computational thinking (CT) and Mathematical thinking (MT)

The nature of the relationship between MT and CT has been extensively researched. There are different definitions and characterizations for these two types of thinking. MT and CT are two complex processes. The first involves the application of mathematical skills for solving mathematical problems and it is enriched and enhanced in an atmosphere of questioning, challenge, and reflection (Mason et al., 2011). CT can include processes such as decomposition, abstraction, algorithmic design, debugging, iteration, and generalization of problems (Shute et al., 2017).
What are the main similarities between the two processes? Both are problem-solving methodologies, as they involve pattern recognition in the structure of problems. Moreover, both of them involve processes, such as decomposition (breaking down problems into smaller steps); algorithm design (deriving general principles from multiple examples); and modelling (translating real-world objects or phenomena into mathematical equations and/or computational relationships) (Liu and Wang, 2010). These two processes also have some common heuristic strategies and more general problem-solving behaviors, e.g., abstract thinking and metacognition, trial and error, ambiguity, flexibility, and the ability to consider and assess multiple ways of problem-solving (Shute et al., 2017).

Compared to MT, CT is a relatively new area of research. It was first presented by Papert in 1990 and, since then, its definition, teaching, and assessment have been discussed by various scholars (Grover and Pea, 2013). Being a relatively new field, the most widely researched issue is in which way the development of CT can assist in teaching math subjects. Furthermore, most empirical studies have targeted CT in school and, hence, little attention has been paid to CT in higher education. In order to differentiate between CT and MT, it is necessary to determine properties that characterize only one type of thinking or, alternately, characterize both types but are a necessary part of one of them.

MT Properties

Discovering – every "new" property that is discovered in the mathematics area, actually already exists in nature. This is the ability to recognize the existing model and to describe it precisely.

Invariant recognition – in mathematics, an invariant is a property of a mathematical object (or a class of mathematical objects) that remains unchanged after operations or transformations of a certain type are applied to the objects.

Proof of existence without design – sometimes, the proof is theoretical, and it is not possible to build a physical model or structure for it. This kind of thinking ability is a part of MT.

CT Properties

Engineering thinking (invention thinking) – a way of thinking that looks for invention. For example, developing new algorithms that do not exist yet. It requires creative and critical thinking.

Solutions without precision. For example, attempting to find a graph sufficiently large is definitely suitable for engineers. For mathematicians, however, it is not defined properly at all.

Algorithmic thinking – also a part of MT. Nevertheless, there is one of the most significant examples of CT skills (Doleck et al., 2017).

Ability to understand and use different programming languages belonging to conceptually different types of programming – Functional Programming (programming languages like Lisp, Haskell) and Imperative Programming (programming languages like C, C++, Java).

Since the field of computer sciences is relatively young, the distribution into fields is less stable than the distribution into field of mathematics. Finding out that in some field, there is a strong correlation to some field of mathematics, it follows that there is justification for this field to be a separate field of computer sciences.

Significance of studying this topic

The field of CT has been rapidly developing. However, most empirical studies have targeted CT in school and, as a result, relatively little attention has been paid to CT in higher education. According to Harris et al. (2015), the value of mathematics in engineering remains a central problem. The researchers argue that mathematics should be a fundamental concern in the design and practice of first-year engineering. The data was gathered from interviews with engineering students (and their lecturers) who experienced problems with mathematics during their first-year study in different courses. This study has exerted efforts in the enhancement of their understanding of the problems of 'becoming engineers'. These problems were due to the mismatch between their expectations and the realities of the course and have once again highlighted the fact that mathematics was central to these problems. A continuing effort is required for clarifying the value of mathematics to engineering in practice.

Flegg et al. (2012) conducted an exploratory study of students' experience in their first year of engineering mathematics studies. Their findings illustrated the relevance of mathematics to different engineering majors, to future studies, and the importance of problem-solving tasks in conveying the relevance of mathematics more effectively than other forms of assessment. Flegg et al. argued that acknowledging the role of mathematics in engineering was perceived by students as crucially relevant. Hence it was necessary to ensure that students take steps to overcome any mathematical difficulties they encounter, promoting progression through the engineering degree.

CT is a rather new area of research and, thus, the relevant question is how to learn and how to teach computational thinking. According to Hsu et al. (2018), CT is considered as an important competence that is required for adaptation to the future. However, educators, especially school teachers and researchers, have not clearly identified the way of teaching it. This study performed a meta-review of the studies published in academic journals between 2006 and 2017, aiming to analyze application courses, adopt learning strategies, participants, teaching tools, programming languages, and course categories of CT education. The review results indicated that the promotion of CT in education made considerable progress in the last decade. In addition to the increasing number of CT studies, conducted in different countries, the subjects, research issues, and teaching tools became also more diverse in recent years. Furthermore, the review found that CT was mainly applied to the activities of program design and computer science, while some studies were related to other subjects. Most of the studies focused on programming skills, training, and mathematical computing, while others adopted a cross-domain teaching mode to enable students to manage and analyze materials of various domains by computing. Moreover, as the cognitive ability of students of different ages varies, the CT ability to cultivate methods and content criteria, should vary accordingly.

University is a configuration part of much bigger establishments and processes, corresponding to the social and individual demands of youth and students. The benefits of this study reside in the possibility of building a model that facilitates effective teaching based on knowledge of the CT-MT relationships even at an early stage (school). Thus, it allows educating better future specialists.

Hypotheses and a brief introduction to the methodology

The preliminary hypothesis is that there is a strong correlation between ability in specific mathematical and computational fields. This study uses the deductive approach (Saunders et al., 2009), according to which the hypothesis was developed, and a research strategy for examining the hypothesis was designed, as is described below. This research process corresponds to five sequential stages of deductive research processing presented by Robson (2002):

1. deducing a hypothesis;
2. expressing the hypothesis in operational terms;
3. investigating this operational hypothesis;
4. examining the specific outcome of the inquiry;

5. if necessary, modifying the theory considering the findings.

The key question is how to test thinking. This study suggests paying attention to the linguistic aspect of the question. Mathematics and programming were not chosen by chance; both disciplines are purely language constructs. There is no possibility of directly analyzing the differences in the types of thinking; rather, there is an option of more or less acknowledging the statement that thinking always happens in some language as a fact (De Saussure, 1916; Heidegger, 1927). Chomsky (1968), "the father of modern linguistics," maintains that there is a strong relationship between language and thinking.

All languages can be divided into a "formal language" and a "metalanguage" (Tarski, 1944). A metalanguage is a language used for describing a formal language. The theory of truth, conceived by Tarski in 1935 (Gruber, 2016) - advocates that formal language should be encompassed in the metalanguage. For example, the proposition "2+2=4" belongs to formal mathematical language, while "2+2=4 is valid" is categorized as belonging to metalanguage of mathematics.

Since mathematics and programming have explicit languages, and people use metalanguage while thinking, we can investigate the relationship between metalanguages to assess the thinking process. Every field of Mathematics and Computer Science has its professional slang. It consists of terms, specific syntax construction of sentences, specific order of words in any sentence, etc. This professional slang of any separate field of Mathematics and Computer Science is defined as local metalanguage.

In order to obtain a more explicit hypothesis regarding the relationship between abilities in specific mathematical and computational fields, this study presents a new approach, i.e., comparing separately the local metalanguages of different field of Mathematics and Computer Science. The data mining stage of this study applies modern technology - a suitable neural network that compares texts written in different local metalanguages.

Research methodology

The methodology of this study consists of three main stages, as follows:

- Data mining stage with technology for text comparison, probably using a suitable neural network. This stage compares many text files from different fields of Mathematics and Computer Science. The aim is to obtain a more explicit hypothesis regarding the relationship between abilities in specific mathematical and computational fields.
- Qualitative research conducted with the students' participation, according to the hypothesis formulated at the first stage.
- Quantitative research conducted with the students' participation, according to the first stage hypothesis and analysis of the results from the second stage.

This paper focuses on the analysis of the first stage.

The First Stage - Data Mining

Data mining (Cheng, 2017; Roiger, 2017; Hand, 2007; Romero,2013) is a collective name that refers to a set of methods for detecting previously unknown, non-trivial, practically useful and accessible knowledge of data, necessary for making decisions in various fields of human activity. The more complete definition of data mining is finding knowledge in databases. The basis of data mining methods is all kinds of classification, modeling, and forecasting methods based on decision trees, neural networks, genetic algorithms, etc.

Usually, the process is as follows:
- There is a sufficiently large database.
- It is assumed that there is some "hidden knowledge" in the database.
- Developing methods for discovering knowledge hidden in large volumes of initial "raw" data is necessary. In the current conditions of global competition, the found patterns (knowledge) can be a source of additional competitive advantage.

What does "hidden knowledge" imply? It must be knowledge of:
- previously unknown - that is, such knowledge that should be new (and not confirming any previously obtained information);
- non-trivial - that is, those that cannot be seen (with direct visual analysis of data or when calculating simple statistical characteristics);
- practically useful - that is, such knowledge that is of value to the researcher or consumer;
- accessible for interpretation - such knowledge that is easy for visual presentation to the user and for explanation in terms of the subject-matter.

Data mining methods can be applied both for working with big amounts of data and for processing relatively small amounts of data (obtained, for example, from the results of individual experiments or when analyzing data about the company's activities). As a criterion for sufficient amount of data, both the field of research and the applied analysis algorithm are considered.

Models for Text Comparison

For the purpose of comparison, a text must be transformed into some mathematical object, for example, a vector. It means that Natural Language Processing must be applied to any text.

What is Natural Language Processing (NLP)?

Natural Language Processing (NLP) (Chowdhary, 2020; Kang et el., 2020) is a machine learning technology that allows computers to interpret, manipulate, and understand human language. Organizations today have large volumes of audio and written text data from various communication channels, such as e-mails, text messages, social media feeds, video, audio, and more. They use NLP software to automatically process these data, analyze the intent or sentiment in the message, and respond to human communication in real time.

Natural language processing is important to the efficient analysis of written and audio data. In this way, differences in dialects, slang, and grammatical irregularities, typical of everyday conversations, can be overcome. It may be used for text classification and extraction.

How does NLP work?

Natural Language Processing combines computational linguistics, machine learning, and deep learning models for human language processing.

Computational linguistics constitutes the science of understanding and building models of human language by using computers and software tools. Researchers use computational linguistics techniques, e.g., syntactic and semantic analysis, to create platforms that are designed to help machines understand spoken human language or texts. Tools such as language translators, text-to-speech synthesizers, and speech recognition software are based on computational linguistics.

Correlation and a Measure of Correlation between Two Sets of Data

Correlation (from Latin correlatio "ratio"), or correlation dependence - a statistical relationship between two or more random variables, while changes in the values of one or more of these quantities are accompanied by a systematic value change of one or other quantities. In mathematical statistics, the Pearson correlation coefficient (Cohen et el.,2009), also known as the pair correlation coefficient or the Pearson moment product correlation coefficient, measures the magnitude of a linear relationship (correlation) between two variables. It takes values from -1 to +1. A coefficient value of +1 means the presence of a complete positive linear relationship, and a value of -1 means the presence of a complete negative linear relationship.

Clusters

Clustering (or cluster analysis) is the task of dividing a set of objects into groups called clusters. Each group comprises "similar" objects, and the objects of different groups should be as different as possible. The cluster analysis is described by the following steps:

- Selection of a sample of objects for clustering.
- Definition of a set of variables by which the objects in the sample are assessed. If necessary, normalize the values of the variables.
- Calculation of similarity measure values between objects.
- Application of the cluster analysis method to create groups of similar objects (clusters).
- Presentation of analysis results.

After obtaining and analyzing the results, changing the selected metric and clustering method is possible until an optimal result is achieved.

Methodological process of the first stage of this study

At the first stage, the following fields of mathematics are investigated: linear algebra, abstract algebra, combinatorics and probability, mathematical analysis, set theory and logic. Moreover, the following fields of Computer Science are explored: functional programming, imperative programming, object-oriented programming, data structures and algorithms, automata-based programming, and compilation.

About 1000 texts in each field of Mathematics and Computer Science are given. Thus, the process of the first stage is the following:

- Using any NLP model for text comparison, like n-gram model (syntactic analysis) or Word2vec (semantic analysis). As a result, each text is converted to a vector (or any other mathematical object that enables comparison).
- Choosing suitable metrics for calculating the distance between the vectors. It should be a correlation metric, like Pearson or Spearman coefficient.

After obtaining the matrix of distances between the texts, any clustering method is applied to find the correlation between different fields of Mathematics and Computer Science.

Variables and measures

MT ability: The functionality of math ability is based on math-related tests, separately for the following math fields: linear algebra, abstract algebra, combinatorics and probability, mathematical analysis, set theory, and logic. Students will be asked to solve a set of exercises, some from each field, and a grade will be given separately for different fields. For each field, students will obtain a grade between 1 (lowest grade) and 10 (highest grade) (a 6 is a pass) (Korpershoek at el., 2015).

CT ability: The operationalization of computational ability is performed and measured in the same way. The tests are related to the following fields: functional programming, imperative programming (Alegre et al., 2015), object-oriented programming, fata structures and algorithms, automata-based programming, and compilation.

Suggested Statistical analysis

Descriptive statistics for the entire study is presented in the form of means and standard deviations for continuous variables, including CT and MT outcomes.

The Spearman correlation matrix is analyzed to identify bivariate correlations between CT and MT variables. The researchers of this study use principal component analysis (PCA) for the purpose of obtaining common factors between the elements of CT and MT both jointly and separately. Advance analyses are performed to find meaningful structure between CT and MT, specifically Partial Least Squares (regression (PLS) (Höskuldson, 1988) and hierarchical clustering. PLS assist in identifying which combinations of MT elements predict CT, while clustering indicates common combinations of MT and CT which are beyond the scope of bivariate analysis.

Discussion and conclusions

There are several points for discussion about the methodological process of the first stage of this research:

Since the two models, N-gram model and the Word2vec model, perform different kinds of analysis (syntactic and semantic analysis), it is recommended trying both of them in order to compare the results.

It seems that the Euclidean metric is not suitable here. Let us suppose there are two text files. Each word of the first file appears in the second file ten times. The result of calculating the Euclidean metric is that there is a sufficiently significant distance between the two files. However, they have the same words and, thus, they have the same meaning. Since the correlation is measured, a correlation metric such as Pearson or Spearman coefficient (Van Dongen and Enright, 2012) can be applied here.

Since the Euclidean metric is not suitable here, so K- means clustering method (Lloyd, 1957; Steinhaus, 1957). It is a key question whether the PAM algorithm can find the correct cluster partition. For example, if there are three points A, B, and C., A is close to B, and B is close to C. However, A is not sufficiently close to C, according to the calculated distance method. As a result, A and C will be divided into different clusters. In the case of different field of Mathematics and Computer Science, though, they should be in the same cluster. Thus, density-based spatial clustering of applications with noise seems to be suitable here.

As mentioned above, the preliminary hypothesis is that there are specific fields of mathematics in which success is correlated with specific areas in the programming field. When investigating which areas exactly correlate, one should know how the correlation relates to the learning method in these courses. Examination of the type of these relationships can provide an understanding of successes or failures among students in specific courses. Hence, it can provide tools that can improve teaching processes that help students in acquiring the necessary missing skills and forms of thinking in CT. These tools are helpful for several categories of people. Students are able to plan higher education studies, even at the school stage, if the relationship between MT and CT is known in advance. Lecturers can teach according to the plan that should lead to fewer students' failure in specific courses.

References

1. Alegre, F. and Moreno, J. (2015). Haskell in Middle and High School Mathematics. TFPIE, 1.
2. Berelson, B. (1952). Content Analysis in Communication Research. Glencoe, Ill: Free Press.
3. Biletzki, A. and Matar, A. (2020). Ludwig Wittgenstein. In: E. N. Zalta, ed. The Stanford Encyclopedia of Philosophy (Spring Edition).
4. Cheng, J. (2017). Data-mining research in education. arXiv preprint arXiv:1703.10117.
5. Chomsky, N. (2006). Language and mind. Cambridge: Cambridge University Press.
6. Chowdhary, K. (2020). Natural language processing. Fundamentals of artificial intelligence. Berlin: Springer, pp. 603-649. https://doi.org/10.1007/978-81-322-3972-7
7. Cohen, I., Huang, Y., Chen, J. and Benesty, J. (2009). Pearson correlation coefficient. Noise reduction in speech processing. Heidelberg: Springer, pp. 1-4. https://doi.org/10.1007/978-3-642-00296-0_5
8. Collis, J. and Hussey, R. (2003). Business research. 2nd edition. Britain, Germany: Palgrave MacMillan Publishers.
9. De Saussure, F. (2011). Course in general linguistics. New York: Columbia University Press.
10. Doleck, T., Bazelais, P., Lemay, D., Saxena, A. and Basnet, R. (2017). Algorithmic thinking, cooperativity, creativity, critical thinking, and problem solving: exploring the relationship between computational thinking skills and academic performance. Journal of Computers in Education, 4, pp. 355–369. https://doi.org/10.1007/s40692-017-0090-9
11. Van Dongen, S. and Enright, A.J. (2012). Metric distances derived from cosine similarity and Pearson and Spearman correlations. arXiv preprint arXiv:1208.3145.
12. Flegg, J., Mallet, D. and Lupton, M. (2012). Students' perceptions of the relevance of mathematics in engineering. International Journal of Mathematical Education in Science and Technology ,43(6), pp. 717-732. https://doi.org/10.1080/0020739X.2011.644333
13. Frame, S. and Coffey, J.W. (2014). A comparison of functional and imperative programming techniques for mathematical software development, Journal of Systemics, Cybernetics and Informatics, 12(2), pp. 1–10.
14. Glock, H. (2013). What is analytic philosophy. Journal for the History of Analytical Philosophy, 2(2).
15. Grover, S.R. and Pea, R. (2013). Computational thinking in K–12: a review of the state of the field. Educational Researcher, 42(1), pp. 38-43. https://doi.org/10.3102/0013189X12463051
16. Gruber, M. (2016). Alfred Tarski and the" Concept of Truth in Formalized Languages": A Running Commentary with Consideration of the Polish Original and the German Translation (Vol. 39). Heidelberg: Springer.
17. Hand, D.J. (2007). Principles of data mining. Drug safety, 30, pp. 621-622. https://doi.org/10.2165/00002018-200730070-00010
18. Harris, D., Hernandez-Martinez, P., Pepin, B., Williams, J., Black, L., with the TransMaths Team (2015). Mathematics and its value for engineering students: what are the implications for teaching? International Journal of Mathematical Education ,46(3), pp. 321-336. https://doi.org/10.1080/0020739X.2014.979893
19. Harwood, T.G. and Garry, T. (2003). An overview of content analysis. The marketing review, 3(4), pp. 479-498. https://doi.org/10.1362/146934703771910080
20. Heidegger, M. (2010. Being and time. Albany, NY: Suny Press.

21. Höskuldsson A. (1998). PLS regression methods. Journal of Chemometrics 2(3), pp. 211-228. https://doi.org/10.1002/cem.1180020306
22. Hox, J.J. and Boeije, H.R. (2005). Data collection, primary vs. secondary. K. Kempf-Leonard (Ed.), Encyclopedia of Social Measurement, (pp. 593.599), Elsevier Science.
23. Hsu, T.C., Chang, S.C. and Hung, Y.T. (2018). How to learn and how to teach computational thinking: Suggestions based on a review of the literature. Computers and Education, (126), pp. 296–310. https://doi.org/10.1016/j.compedu.2018.07.004
24. Kang, Y., Cai, Z., Tan, C.W., Huang, Q. and Liu, H. (2020). Natural language processing (NLP) in management research: A literature review. Journal of Management Analytics, 7(2), pp. 139-172. https://doi.org/10.1080/23270012.2020.1756939
25. Korpershoek, H., Kuyper, H. and van der Werf, G. (2015). The relation between students' math and reading ability and their mathematics, physics, and chemistry examination grades in secondary education. Int J of Sci and Math Educ 13, pp. 1013–1037. https://doi.org/10.1007/s10763-014-9534-0
26. Krippendorff, K. (1980). Content Analysis: An Introduction to Its Methodology. Newbury Park, CA: Sage.
27. Liu, J. and Wang, L. (2010). Computational thinking in discrete mathematics. Paper presented at the Second International Workshop on Education Technology and Computer Science.
28. Lloyd, S. (1957). Least square quantization in PCM. Bell Telephone Laboratories Paper. Published in journal much later: Lloyd, Stuart P. (1982).
29. Luo, A. (2022). Content Analysis | A Step-by-Step Guide with Examples. Scribbr. https://www.scribbr.co.uk/author/amyluo/bbr
30. Mason, J., Burton, L. and Stacey, K. (2011). Thinking Mathematically. 2nd edition. Pearson Higher Ed.
31. Papert, S. (1990). A critique of technocentrism in thinking about the school of the future. Epistemology and learning memo #2. September 1990, MIT.
32. Petzold, C. (2008). The annotated Turing: a guided tour through Alan Turing's historic paper on computability and the Turing machine. Hoboken, NJ: Wiley Publishing.
33. Robson, C. (2002). Real World Research. 2nd edition. Oxford: Blackwell.
34. Roiger R. (2017). Data Mining: a Tutorial-Based Primer. Boca Raton: CRC Press.
35. Romero, C. and Ventura, S. (2013). Data mining in education. Wiley Interdisciplinary Reviews: Data mining and knowledge discovery, 3(1), pp. 12-27. https://doi.org/10.1002/widm.1075
36. Russell, B. (1908). Mathematical logic as based on the theory of types. American journal of mathematics, 30(3), pp. 222-262. https://doi.org/10.2307/2369948
37. Saunders, M., Lewis, P. and Thornhill, A. (2009). Research Methods for Business Students. 5th edition. London: Pearson.
38. Shute, V.J., Sun, C. and Asbell-Clarke, J. (2017). Demystifying computational thinking. Educational Research Review, 22, pp. 142-158. https://doi.org/10.1016/j.edurev.2017.09.003
39. Steinhaus, H. (1957). Sur la division des corps matériels en parties. Bull. Acad. Polon. Sci. 4(12), pp. 801–804. [French]
40. Tarski, A. (1944). The semantic conception of truth: and the foundations of semantics. Philosophy and phenomenological research, 4(3), pp. 341-376.
41. Weber, R.P. (1990). Basic Content Analysis. 2nd edition. Newbury Park, CA: Sage.
42. Whitehead, A. N. and Russell, B. (1997). Principia mathematica to* 56, 2nd edition. Cambridge: Cambridge University Press.
43. Zalta, E. (2022). Gottlob Frege. The Stanford Encyclopedia of Philosophy, Fall 2022, Metaphysics Research Lab, Stanford University.