Journal of Universal Computer Science, vol. 24, no. 7 (2018), 815-845
submitted: 22/2/17, accepted: 17/7/18, appeared: 28/7/18 © J.UCS

Mathematics Learning through Computational Thinking Activities: A Systematic Literature Review

Thiago S. Barcelos
(Instituto Federal de Educação, Ciência e Tecnologia de São Paulo, São Paulo, Brazil
tsbarcelos@ifsp.edu.br)

Roberto Munoz
(Universidad de Valparaíso, Valparaíso, Chile
roberto.munoz@uv.cl)

Rodolfo Villarroel
(Pontifícia Universidad Católica de Valparaíso, Valparaíso, Chile
rodolfo.villarroel@pucv.cl)

Erick Merino
(Universidad de Valparaíso, Valparaíso, Chile
erick.merino@postgrado.uv.cl)

Ismar F. Silveira
(Universidade Presbiteriana Mackenzie, São Paulo, Brazil
ismar.silveira@mackenzie.br)

Abstract: Computational Thinking represents a terminology that embraces the complex set of reasoning processes that are held for problem stating and solving through a computational tool. The ability of systematizing problems and solve them by these means is currently being considered a skill to be developed by all students, together with Language, Mathematics and Sciences. Considering that Computer Science has many of its roots on Mathematics, it is reasonable to ponder if and how Mathematics learning can be influenced by offering activities related to Computational Thinking to students. In this sense, this article presents a Systematic Literature Review on reported evidences of Mathematics learning in activities aimed at developing Computational Thinking skills. Forty-two articles which presented didactic activities together with an experimental design to evaluate learning outcomes published from 2006 to 2017 were analyzed. The majority of identified activities used a software tool or hardware device for their development. In these papers, a wide variety of mathematical topics has been being developed, with some emphasis on Planar Geometry and Algebra. Conversion of models and solutions between different semiotic representations is a high level cognitive skill that is most frequently associated to educational outcomes. This review indicated that more recent articles present a higher level of rigor in methodological procedures to assess learning effects. However, joint analysis of evidences from more than one data source is still not frequently used as a validation procedure.

Keywords: Computational Thinking, Mathematics, Teaching/Learning Strategies, Curriculum
Categories: K.3.0, K.3.1, K.3.2

1 Introduction

Computer Science (CS) Education community has recently started to consider CS as a subject that should be part of school curriculum since initial series, being thus put at the same level as those sciences currently known as "basic sciences" – namely, Physics, Biology and Chemistry. However, the motivation for this has been often "self-triggered" by CS, as the main incentive to teach basic computational skills in initial series was to improve people's abilities to deal with computational devices. For instance, [Hood and Hood 2005] affirm that "a key to achieving widespread fluency on Information Technology is to make it part of K-12 curriculum". However, such fluency should not precisely be achieved by only mastering a collection of techniques, but should instead be considered as a way of organizing thinking for concrete problem-solving.

For this reason, a set of skills and abilities related to Computer Science field should be developed by students since earlier series of elementary school. Such set was named by Wing as Computational Thinking (CT) [Wing 2006]. This term has been used to describe the cognitive processes related to abstraction and decomposition to allow problem-solving using computational resources and algorithmic strategies, among other skills. Although the idea of using computers as tools to improve cognitive skills is not new, as it is referred in pioneer works by [Papert 1980] and [Jonassen 2000], the definition of Computational Thinking has contributed to draw attention to this question. This definition has been criticized by some authors for being too wide [Hu 2011, Hemmendinger 2010]; however, it is possible to immediately identify some similarities in CT skills with Math-related ones. Polya, in his classic work about problem solving skills, pointed out that abstraction, defined as a combination of analogy, generalization and specialization, and problem decomposition skills are crucial for a Math student to succeed in problem-solving tasks [Polya 2004].

On the other hand, Mathematics is a subject that is reportedly a hurdle to students in many countries. Hanushek, Peterson & Woessmann analyze statistics on Math educational achievements in United States to conclude that only 6% of students reach an advanced level at the end of the 8^{th} grade. It is noteworthy that even privileged groups, represented in this study by white students with parents who had at least college education, do not generate a higher proportion of students who achieve an advanced level in Math [Hanushek, Peterson and Woessmann 2010]. This may indicate that, as a whole, U.S. schools may be failing to reach adequate educational levels. In developing countries, the situation may be even more critical. For instance, Latin American countries are among the worst performers in Math. According to Aedo & Walker, the average score in PISA Mathematics exams of Argentina, Brazil, Chile, Mexico and Peru is about 100 points lower than the average obtained by students in other OECD (Organization for Economic Co-operation Development) participating countries. This is roughly equivalent to a two years lag in studies for students in these five Latin American countries [Aedo and Walker 2012].

In order to incorporate activities aimed at developing Computational Thinking skills in basic educational levels, one should consider what would be the impact of those activities in learning "traditional" school subjects. This is often the case as basic school is an environment where change is slow and "multiple competing priorities, ideologies, pedagogies, and ontologies all vie for attention" [The CSTA Standards Task Force 2011]. Some relationships between Mathematics and Computer Science in an educational setting have already been discussed (e.g. [Ke 2014, Krone, Sitaraman and Hallstrom 2011, Pioro 2006, Ralston 2005, Taylor, Harlow and Forret 2010]). Hence, it is reasonable to infer that Math teaching and learning would benefit in some way from the incorporation of Computational Thinking activities in basic school curricula.

Many CT activities related to Math skills and contents have been reported in scientific literature in last years, which motivates the need for a detailed analysis of educational outcomes obtained through these initiatives. Hence, in this article we present a Systematic Literature Review (SLR) aimed at identifying how relationships between Math and Computational Thinking have been demonstrated through didactic activities described in literature. More specifically, this research was guided by three research questions:

(Q1) How didactic activities related to Computational Thinking and Math are conducted and which is the target audience for such activities?

(Q2) Which CT and Math skills and topics are taught through didactic activities?

(Q3) Which research methods and procedures are used to identify learning outcomes?

The remainder of this article is organized in the following way. In section 2 we present some definitions of Computational Thinking that are adopted in literature and discuss how skills related to Mathematics are associated to Computational Thinking. In section 3 the method for performing SLR is described. Results are presented in section 4 and further discussed in section 5. In section 6, some conclusions are presented.

2 Theoretical Background

2.1 Computational Thinking

The article by Jeanette Wing organized the notion that skills related to Computer Science should be developed by students since basic educational levels [Wing 2006]. Initially Wing presented the concept of Computational Thinking using various concrete examples of application of Computer Science concepts and a set of generic principles. Later, reference curricula for Computer Science Education in basic education were created or updated to reflect skills that were progressively associated to Computational Thinking. Computer Science Standards proposed by ACM CSTA (Computer Science Teachers Association) was one of the first initiatives with this goal [The CSTA Standards Task Force 2011]. Moreover, as of 2014, seven European countries had already incorporated coding as a compulsory activity in specific levels of basic education – Bulgaria, Czech Republic, Denmark, UK, Portugal, Slovakia and Spain [European Schoolnet 2015], later followed by Austria, Bulgaria, Estonia, Hungary, Ireland, Lithuania, Malta, Poland, France and recently, Finland.

An operational definition proposed by CSTA states that Computational Thinking is, essentially, a problem solving process that allows formulating problems in such a way that a computer, along with other tools, can be used to obtain a solution [Computer Science Teachers Association 2011]. The revised version of Computer Science Standards proposed by this association is organized around five core concepts: Computing Systems, Networks and the Internet, Data and Analysis, Algorithms and Programming and Impacts of Computing [Computer Science Teachers Association 2017]

2.2 Mathematic skills and Computational Thinking

The development of Computing as an area of knowledge during the 20th and 21st centuries has been closely related to Mathematics, particularly if one considers that their first theoretical models, such as Turing Machine or λ-Calculus, for instance, were created to mathematically demonstrate the feasibility of automating numerical calculations. Denning [2005] argued that, throughout the development of Computing, its activities were based on methods and knowledge borrowed from three areas: Natural Sciences, Engineering and Mathematics. The experimentation-based method of Natural Sciences is used for the definition of heuristic algorithms, for instance, while software design and development are clear applications of Engineering methods and techniques. Symbolic representations and axiom-based deduction methods from Mathematics define the basis for demonstrating the correctness and performance of algorithms.

Although there is a crescent notion that some skills related to Computer Science should be developed since basic educational levels, previous works have shown these skills developed together with other “traditional” school subjects, such as Biology, Literature, Arts and – of course – Mathematics [León, Robles and Román-González 2016]. A reasonable justification for this research strategy is stated in 2011 version of CSTA CS Standards, whose authors state that “K-12 education is a highly complex, highly politicized environment where multiple competing priorities, ideologies, pedagogies, and ontologies all vie for attention” [Computer Science Teachers Association 2011]. According to the same authors, as Computational thinking is positioned as a practical approach for problem solving, it can be pervasively applied across subjects. Hence, demonstrating the educational benefits of applying Computational Thinking has been the aim of previous works, as it will be later illustrated in SLR results.

The identification of skills that may be shared between Computational Thinking and Mathematics may be a starting point to demonstrate the aforementioned benefits. According to [Perrenoud, Thurler, de Macedo, Machado and Allessandrini 2007], a competency (or high-order skill) is “a capacity of acting efficiently in a certain kind of situation, supported by knowledge, but without being limited to it”. The possibility of applying knowledge in different contexts is also present in a definition by European Commission, according to which a competency is “a multifunctional and transferable package of knowledge, skills and attitudes that all individuals need for their achievement and personal development, inclusion and employment“ [European Commission 2004].

In this sense, previous works have already identified high-order skills that may be “exchanged” between Computational Thinking and Mathematics. [Barcelos and Silveira 2013] performed an analysis of curriculum guidelines for Mathematics in basic education in three countries (United States, Brazil and Chile) and compared the skills enumerated in these guidelines to those present in various definitions and applications of Computational Thinking available in literature. These authors identified as a result three high-order skills:

- *Alternating between different semiotic representations*: this skill involves translating a situation expressed in one symbolic representation into another. This is applicable to different semiotic representations available in Math knowledge base, such as charts, tables and formulas, and also to verbal or algorithmic representation of a solution.
- *Establishing relationships and identifying patterns*: a skill that is related to situations in which a student should identify regularities and deduce or establish a formation rule. In the case of Mathematics, it is usually related to numerical regularities, but is also related to core abstraction skills for problem solving described by Polya [2004]: analogy, generalization and specialization.
- *Building descriptive and representative models*: by applying this skill, a student should be able to use mathematical or algorithmic language to build models that explain familiar situations, such as profit and loss, kinematics, opinion polls, and many others. The computational support to build and test models (by using spreadsheets, chart plotting software and programming language tools) may bring a potential to apply Computational Thinking to other school subjects.

Finally, it is important to point out that similar descriptions of these skills have subsequently appeared in related works. For instance, K-12 Computer Science Framework, which orientated the definition of aforementioned CS Standards, also associated the development and use of abstractions as a skill that is shared between Computer Science and Math [Association for Computing Machinery, code.org, Computer Science Teachers Association, Cyber Innovation Center and National Math and Science Initiative 2016].

3 Methods

In order to plan and conduct the SLR, we followed the guidelines proposed by [Kitchenham 2004] and further detailed by [Wohlin et al. 2012]. These guidelines define the following stages and activities:

Stage 1. Planning the review.
- 1.1. Identification of the need for a review
- 1.2. Specification of research questions
- 1.3. Development of a review protocol

Stage 2. Conducting the review.
- 2.1. Identification of research
- 2.2. Selection of primary studies
- 2.3. Study of quality assessment
- 2.4. Data extraction and monitoring
- 2.5. Data synthesis

Stage 3. Reporting the review.

3.1 Planning the Review

Initially, a preliminary search was performed in November 2017 using the search engines of ACM, IEEE, and Google Scholar that, to the best of our knowledge, would concentrate most research on CT and its relationships with Math. We did not find any literature review with the same objective as the one proposed in this paper; hence, this search was useful to indicate the need for a review (Activity 1.1). The research questions for the review (Activity 1.2) and their motivations were already described in Section 1.

The next step is related to the definition of a review protocol (Activity 1.3). According to [Wohlin et al. 2012], the protocol should indicate the study search strategy, which includes the online repositories to search for studies, the definition of search strings and inclusion and exclusion criteria. Four repositories were included in this review: ACM, IEEExplore, SpringerLink, and ScienceDirect. Although research on CT traits and activities is quite recent, it was expected that the number of studies about this theme would have increased in last years. Hence, we defined two inclusion criteria for studies that would cover the proposed research questions:

(IC1) the study must indicate some relationship between CT and some skill, specific skill or topic related to Mathematics;
(IC2) the study must describe a didactic activity and present the results of its evaluation.

After preliminary tests with search engines provides by repositories, it could be identified that many documents with the terms “computational thinking” and “math” (or “mathematics”) could be retrieved that did not necessarily cover any topics related to mathematics. Many times, only a generic statement could be found about the relationship between mathematics and CT. Hence, in order to narrow down the results, other keywords were added to the search string, together with the additional restriction that a word with the prefix “math” should be present in the title, abstract or keywords. This strategy was incorporated as an attempt to ensure that mathematics was a main topic of the study. The final version of search string was defined as follows:

```
(("computational thinking" AND (abstract:math* OR document_title:math*
OR keyword:math) AND ("activit*" OR "class*" OR "course") AND ("assess*"
OR "evaluat*" OR "test*")))
```

Where * operator indicates “any word with the given prefix”, colon operator indicates a search specifically in the given document field, and quotation marks indicate literal search as usual. The search string was adapted to particular features of every search engine as necessary. Also, search engines of every repository were configured so that retrieved results would only include papers actually stored in that repository, in order to avoid duplicate entries (such as a paper stored in one repository and indexed by other one, for example).

Likewise, three exclusion criteria were defined:

(EC1) The article describes a tutorial, demonstration, discussion panel proposal or interview;
(EC2) CT (or Computer Science Education) is the article's theme; however, no relationships to Math are explicitly presented or discussed;
(EC3) The article is out of context and addresses another research theme.

3.2 Conducting the Review

Queries were performed in January 15th, 2018, in four selected repositories when 232 documents were retrieved and then classified, based on their titles and abstracts. When it was still not clear if an article could be included in the review, it was marked for later verification of its full text. After this analysis, 71 articles matched at least one of inclusion criteria and 161 articles were excluded. Statistics for this analysis are presented in Table 1.

	Included articles		**Excluded articles**			**Total**
Database	IC1	IC2	EC1	EC2	EC3	
ACM	2	13	5	15	25	60
IEEExplore	2	6	3	16	24	51
ScienceDirect	1	7	0	3	7	18
SpringerLink	24	16	6	13	44	103
Total	71		161			232

Table 1: Statistics for retrieved articles

In order to assess inter-rater reliability with respect to selection of studies, a sub-sample of 20 from 232 papers was coded independently by two reviewers. The inter-rater reliability (r) was 0.83, showing good agreement between the two coders.

It was not necessary to filter queries by publication year as we assumed that publications that addressed Computational Thinking and related subjects would appear after 2006, year of the publication which defined this term [Wing 2006]. For a first analysis of scientific community's interest on this research topic, we initially separated the included articles in two groups. First group (EXP), with 42 articles, included those articles that described the application of a didactic activity designed to help students develop CT-related skills, together with any experimental evaluation of its effects. The second group (DI) was formed by 29 articles where authors discussed relationships between CT and Math based on their viewpoint, a documental analysis, or other types of theoretical work. We also included in this group articles in which didactic activities were only proposed with no description of its application or evaluation.

A graph depicting the number of articles in both groups by publication year (Figure 1) shows that there is a growing interest in scientific community for development and evaluation of didactic experiences in which concepts or skills related to CT and Math are mentioned. The first didactic experiment included in results was reported in 2009, and a growing tendency is verified since 2011 with a slight decrease in 2014 and 2016. However, the number of selected articles published in 2017 was roughly four times bigger than the previous year.

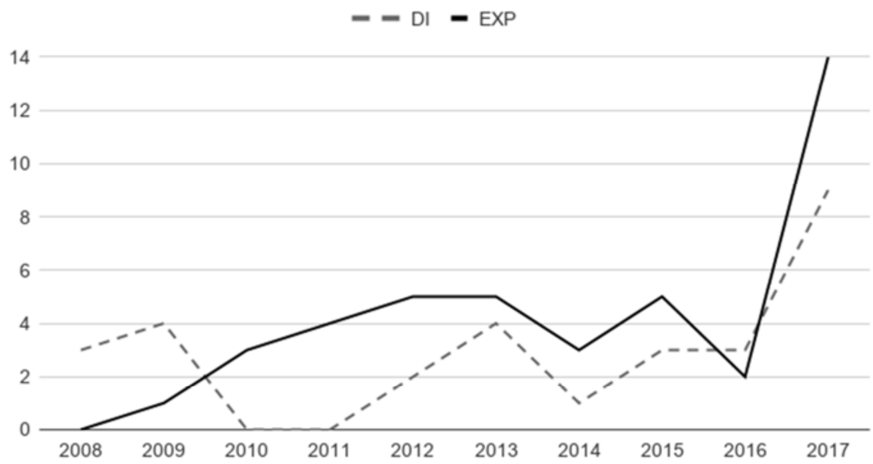

Figure 1: Selected articles presenting didactic experiments (EXP) and theoretical discussions (DI) about CT and Math by year of publication

The origin of the studies was determined by the affiliation of its authors. In this sense, it was identified that, in EXP group, 25 studies (59% of total) were published by authors affiliated to institutions in the United States of America. 14 studies (33.3%) were published by authors of other countries (Brazil, Cyprus, Spain, Canada, Qatar, Finland, France, Greece, Hungary, England, Italy and New Zealand). The remaining three articles (7.1%) were published by authors affiliated to institutions in different countries.

The same analysis in DI group showed that 8 studies (27.6% of total) came from authors of USA institutions, 12 studies (41.4%) were published by authors in institutions of Australia, China, Brazil, Sweden, Norway, France, Italy and Turkey, while 9 studies (31%) came from international collaborations.

4 Results

4.1 Target Audience

In each EXP group article, the target audience of the proposed or reported activity was identified. Due to the fact that searched databases displayed articles mainly in English, the American terminology was adopted for education levels: "elementary school", for the first five years of basic education; "middle school" for the three following years of basic education; and "high school", for the last four years. The term "kindergarten" was freely used for educational activities previous to official basic education - in general, authors of articles who carry out experiences outside the context of the United States also use this terminology to report their experiences. For articles in which this strategy was not used, the education level was deducted from reading in order to unify the counting. Collected data is presented in Table 2.

Educational Level	**# articles**	**% articles (n = 42)**
Single audience:		
Elementary School	6	14.3%
Middle School	10	23.8%
High School	1	2.4%
Undergraduate students	11	26.2%
Teachers	2	4.8%
Graduate students	2	4.8%
Mixed audiences:		
Kindergarten / Elementary School	2	4.8%
Elementary School / Middle School	2	4.8%
Elementary / High School	1	2.4%
Middle School / High School	2	4.8%
Middle School / Teachers	1	2.4%
Middle School / High School / Undergraduates	1	2.4%
High School / Undergraduate students	1	2.4%
Total	42	100.0%

Table 2: Target audience of didactic activities

It is possible to identify that a significant number of studies (24 articles, or 57.1% of total) presented experiences addressed to one or more levels of basic education, including the articles that describe experiences targeted to more than one audience. On the other hand, 11 articles were identified to have exclusively undergraduate courses as target audience or had their experimental validation in that environment. Although original definition of Computational Thinking [Wing 2006] considers its development as a proposal mainly focused on basic educational levels, many researchers also propose computational thinking development oriented activities in conjunction with math for undergraduate students. These activities are focused mainly on introductory courses, seeking to mitigate high rates of desertion and failure in subjects of Computer Science. However, it is also necessary to note that community's interest in the conduction of experiments in elementary education has been growing during the latest years. Presented as evidence, of 26 articles demonstrating this kind of experiences, 13 of them (50.0% of total) were published between 2015 and 2017.

4.2 Software Tools and Materials

Didactical experiences described in selected studies have predominantly used software tools during their development. From the mapping of tools and materials used in articles, 35 occurrences of use of software tools were identified and barely 7 occurrences of activities that do not depend on computer use. In Table 3, software tools which were used are presented, as well as their frequency of use. It should be noted that when a study mentioned more than one software tool, the one that was most used was considered for this analysis.

Software tools	**Frequency of use**
Scratch	7
Agent-based programming	4
Specific-purpose software tool	4
Learning Object	4
Robotics + Software	3
Java	2
Python	2
MATLAB, Octave	2
MAPLE, Racket	2
Electronic circuit boards	1
Logo 3D	1
ScratchJr	1
Hopscotch	1
CAD Software	1
Total	35

Table 3: Software tools used in didactic activities

From 35 studies that mentioned software tools, 22 used a programming language or environment for didactic activities. The most used programming environment was Scratch, with 7 occurrences, and its variant ScratchJr, used in just one study. This finding is consistent with the results of a previous SLR, focused on the use of Scratch, which identified that math skills are reportedly developed in activities using this language [Moreno-León and Robles 2016]. It is also noteworthy that in 4 studies an agent-based programming tool was used; these studies were published from 2013 to 2017 and demonstrate a potential to engage students in modeling and simulation activities, as it was previously proposed by [Lee et al. 2011]. Six studies used conventional programming languages, such as Java, Python, MATLAB or Octave, while two studies used MAPLE and Racket, which are respectively environments suited for algebraic and functional programming.

In 8 studies a Learning Object or specific software tool was used, probably in an attempt to explore some aspect of Mathematics learning that is not necessarily covered by a programming activity. Among the described software tools, there is a mobile game for measurement estimation [Arroyo et al. 2017], an interactive gallery of graphical artifacts [Wilkerson-Jerde 2014] and a simulation tool [Alab, Magan and Garci 2013]. Although Computational Thinking may be frequently mistaken for programming, the first definition by [Wing 2006] already indicated that CT skills had the purpose of generating ideas and not (necessarily) artifacts. For example, in [Hauze, French, Castañeda-Emenaker, French and Singer 2017] an activity where students use a Computer-Aided Design (CAD) software to model pieces to build a guitar is described, and in [Biró, Csernoch, Máth and Abari 2015] problems involving data analysis using spreadsheets are applied to evaluate algorithmic skills of secondary students in Hungary in correlation to their Math skills.

4.3 Research Strategies for Learning Effects Evaluation

Next step was to identify the level of methodological rigor in the evaluation of learning effects in selected studies. Different criteria are found in literature to evaluate the quality of a research work. For studies that adopt the quantitative paradigm, [Trochim and Donelly 2006] indicate that experiments should adopt a random distribution of individuals in control and experimental groups. When this cannot be done due to environmental restrictions (which is often the case in educational research), a quasi-experiment may be adopted as the research design, where the comparison is performed between non-equivalent groups, as there is not a random distribution of individuals to groups. For a study that adopts the qualitative paradigm, [Creswell and Miller 2000] propose that the quality should be analyzed through three "lenses": of the researcher, of participants of the study, or of people outside the study. According to these authors, the features most commonly associated to a rigorous qualitative study include the "prolonged involvement in the field", understood as the involvement of a researcher in research environment for a period of 4 months up to one year, a "dense and rich description" of observed phenomena, including environment and conversations of the research subjects, and triangulation of methods, which is the use of different data gathering instruments to obtain corroborating evidences.

First step was to identify research paradigms and methods adopted by the studies classified in the EXP category, based on the classification above outlined. The classification is presented in Table 4.

Research paradigm / strategy	# articles	% articles (n = 42)
Quasi-experiment	21	50.0%
Experience report	8	19.1%
Qualitative case study	5	11.9%
Qualitative	3	7.1%
Experiment	3	7.1%
Qualitative mixed-methods	2	4.8%
Total	42	100.0%

Table 4: Research paradigm and methods for studies in the EXP category

It is noteworthy that in half of the didactic experiences that presented the results of some experimental evaluation the quasi-experimental design was adopted. This is an expected result, considering that educational research is commonly constrained by logistical and organizational issues. “Experience report” stands for studies aimed at describing the execution of a didactic activity but presenting only anecdotal evidence of learning effects, or an insufficient sample size. Five studies adopted the qualitative case study method, while three studies adopted other qualitative methods, such as discourse analysis, and two studies claimed to have used a mixed methods perspective due to the sample size that was obtained and analyzed in the studies. In Table 5 the data gathering instruments used in the studies and their frequency is presented. It should be noted that a single study can use more than one data gathering tool.

Instrument	# of occurrences	% articles (n = 42)
Learning assessment –Pre and post-tests	17	40.9%
Classroom observation	15	35.7%
Survey	11	26.2%
Interview	10	23.8%
Grades	8	19.1%
Analysis of developed artifacts	6	14.3%
Learning assessment - Post-test	5	11.9%
Student approval rates	4	9.5%

Table 5: Data gathering techniques used in studies in the EXP category

The assessment of learning itself is typically carried out through problem solving related to mathematical school content or using results of already available official assessments. The joint use of tests before and after the offering of the didactical experience was identified in 17 studies, while the use of a single test after the offering of the experience occurs in 5 studies. It is possible to identify an intensive use of questionnaires by researchers. Data gathering tools usually associated to the qualitative paradigm, like observation of class activities (15 occurrences) and interviews (10 occurrences) are also frequent. However, as seen in Table 6, roughly half of the studies (47.6%) use a sole data collection instrument, which could be a limitation for the final validation of the results, considering the educational context where the research is inserted. This point will be discussed in section 4.

# of data collection instruments	# articles	% articles (n=42)
1	20	47.6%
2	13	31.0%
3	6	14.3%
4	3	7.1%
Total	42	100.0%

Table 6: Number of data collection instruments

As seen in the previous section, most of the studies describe didactical experiences in which students handle software tools in order to build their own artifacts, such as small programs, digital games and spreadsheets. However, an analysis of the digital artifacts produced by students was carried out only in 6 studies. For example, Taylor, Harlow and Forret [2010] include programs produced with Scratch in the analysis of a varied set of collected data. The purpose is to produce a qualitative analysis of the mathematical concepts explored by 10 year-old students during games creation activities. [Sengupta, Krishnan, Wright and Ghassoul 2015] use a similar perspective to identify relationships between several elements of physical artifacts and computational models created by students and their STEM learning outcomes. Ke [2014] used a tool for semi-automatic analysis of Scratch code produced by students in order to identify how their mathematical thinking was incorporated into the produced games.

4.4 Math Skills and Contents

Subsequently we analyzed which math-related skills and contents are being developed in conjunction with CT in the selected studies. From the grouping of studies that develop similar topics, it was possible to identify eight groups of Math contents, that are shown in Table 7. For each group, examples of developed topics are also presented.

Math content	# articles	% articles (n = 42)	Contents' examples
Euclidean geometry	20	47.6%	Area, scale, angles, properties of plane geometric figures
Algebra	12	28.6%	Use of variables, algebraic equations, inequalities, complex numbers
Arithmetic	10	23.8%	Basic arithmetic operations, integer division, ratio and proportions, decimal and hexadecimal numbering systems
Math modeling	8	19.0%	Modeling of pre and post conditions of functions State machine modeling
Physics	7	16.7%	Speed, acceleration, forces, torques, speed x time graph interpretation
Calculus	4	9.5%	Graph analysis, definition of functions, gradient functions, partial derivatives
Statistics	4	9.5%	Probability, decision trees, fundamental counting principle
Linear Algebra	4	9.5%	Cartesian coordinates system, matrices, vectors, analytic geometry

Table 7: Developed Math contents

In order to establish the grouping of articles we opted to separate studies that mentioned the development of Algebra from those studies that mentioned Linear Algebra due to the particularities of the latter, which is more strongly focused towards the conversion between different semiotic representations - this will be relevant in the following discussion. We also opted to include Math modeling in the grouping, although it is not properly a topic but a didactic strategy, as we intended to associate studies that mentioned the strategy with higher order skills that will be relevant in our analysis.

Once again, it should be noted that the same study can address skills or contents in more than one group (i.e., Algebra and Physics). The full list of Math contents developed by the didactical activities described in each article is presented in Appendix A. From that mapping, it is possible to note that a wide variety of mathematical skills and contents are being addressed, with some predominance for activities that develop topics related to Planar geometry, Algebra and Arithmetic.

Although an analysis of developed Math topics can provide some insights about the joint development of CT and Math, it is also relevant to associate the educational outcomes with high-order skills that may provide a clearer association between the two subjects. Hence, a mapping between Math contents and high-order skills was performed, based on the three skills identified by [Barcelos and Silveira 2013]. Considering the predominance of studies with north-American authors, the Common Core State Standards for Math [Common Core State Standards Initiative 2012] was used as a reference to map contents to high-order skills. Again, two authors of this article were responsible for this mapping. The authors worked in an independent fashion to map the 42 articles in the EXP group to one or more of the three high order skills. The reported inter-rater reliability (r) for this mapping was 0.89, and discrepant mappings were discussed in second stage in order to reach a common result. The result of the mapping is displayed in Figure 2.

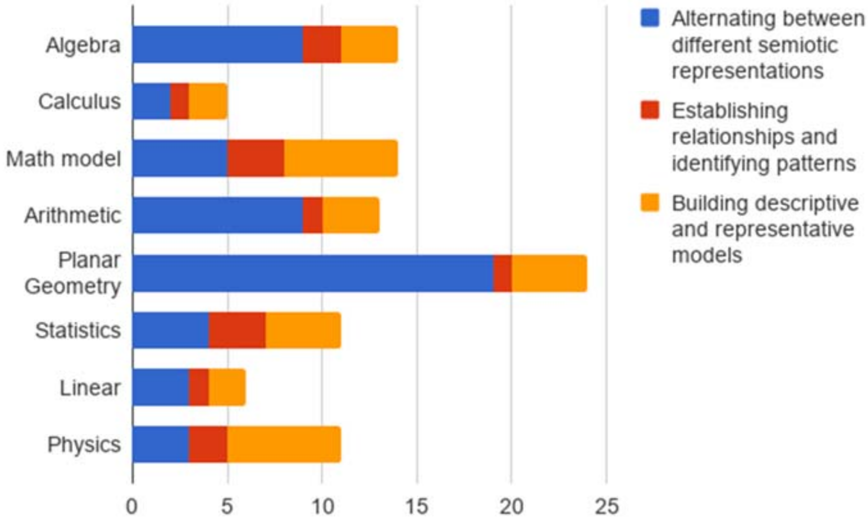

Figure 2: Association of developed Math topics to high order skills proposed by [Barcelos and Silveira 2013]

The mapping revealed that the alternation between different semiotic representation was the high order skill most frequently developed in the selected studies. Apart from being present in national curricular guidelines for Math, this skill is frequently applied in activities that involve some sort of programming activity. This may be interpreted as a materialization of the proposal by [Mor and Noss 2008], for whom programming can be seen as an intermediate step between the ambiguities of natural language and the precision of Math language. This was the skill mostly associated with Math topics, in special Planar geometry and Arithmetic. The use of visual tools in these studies, such as Scratch, Logo and variations of CAD software, may foster the mentioned skill development.

The second high order skill – establishing relationships and identifying patterns – is mostly associated to activities involving Algebra, Statistics and Math modeling. They consist of Math languages that are appropriated to present the generalization of formation rules and patterns, which is also an activity that is closely related to traits of abstraction skills [Polya 2004]. The third skill, building descriptive and representative models was associated to Math modeling activities and developed Physics topics. In the first case it can be rather an expected result as a descriptive model is a mandatory outcome of Math modeling. In the second case, it can be seen that authors that chose Physics as the context for developing Computational Thinking found a variety of situation in which computational simulation together with a mathematical addressing of phenomena could be applied.

The topics of Algebra and Calculus are used to contextualize engineering projects [Chiu et al. 2013], sensors for robotics projects [Dyne and Braun 2014] and data processing with electronic spreadsheets [Tort, Blondel and Bruillard 2008]. Studies that use visual programming tools such as Scratch, Logo and Python, due to the particular characteristics of these tools, invariably explore development of Cartesian plane and Analytic Geometry topics. Lewis and Shah [Lewis and Shah 2012] and Taylor, Harlow and Forret [Taylor et al. 2010] discuss the appropriation of geometric relations and Cartesian coordinates by elementary school students when producing their programs with Scratch. 3D programming environments can even demand the mobilization of spatial geometry concepts [Taylor et al. 2010].

Studies related to Physics, in particular to concepts of kinematics, appear in the revision due to the mathematical addressing needed for their analysis. The work of Sengupta et al. [Sengupta, Kinnebrew, Basu, Biswas and Clark 2013] and Sengupta and Farris present an approach using programming based on agents in order to introduce concepts of speed and acceleration in elementary school students. Hoji, Vianna and Felix [Hoji, Vianna and Felix 2012] present a didactical strategy which introduces the concepts of forces and torques to students of a professional mechanics course, using Octave, a mathematical programming language.

The lack of studies that use the Mathematical Modeling strategy should be considered, since the construction and interpretation of models might be considered as a skill that is common to Math and Computational Thinking as mentioned before in this section. Only eight studies use that approach. For example, in [Buteau and Muller 2017] students had to model dynamic systems based on complex systems such as the logistic function or the Mandelbrot set. The models generated had to be graphical, numerical and mathematical. When students finished these previous steps, they had to implement it with Maple language. The evaluation of this approach was based on grades and informal interviews. On the other hand, authors expose the necessity to incorporate other types of evaluation instrument, such as artifact analysis. Further, [Oliveira 2012] works with four different university courses whose subject was Turing machines. The author evaluated students' skills related to abstraction, through the ability to abstract what one or more machines compute, knowing the input and output of some of its calculations. It was found the existence of a statistically significant correlation, between mental ability to compute and student performance in the in courses where the intervention was performed.

4.5 Conceptual Discussions

According to Wohlin et al. [Wholin et al. 2000], the research questions that guide a systematic revision of the literature must be focused on experimental evidences brought by studies associated with them. However, the particularities of research in Education, context in which this revision is inserted, justify at least a more detailed look of the conceptual studies that were identified from the query results described in section 2.2. A previous analysis of the 63 articles classified in that group allowed to define two categories for grouping:

- Perspective (21 articles): Works whose argumentation is sustained mainly by the experience or personal opinion of the author;

- Curricular Mapping (42 articles): Works that seek to analyze comparatively the competencies, skills and contents that are present in curricula guidelines of Math and Computation fields;

The articles of the Perspective category start with the work of [Wing 2006]. In them, it is discussed about the importance of Computational Thinking and it is mentioned, barely in a generic way, its possible relations to Math. The articles in the Curricular Mapping category seek to identify convergences between Computational Thinking and other contents of school curricula, necessarily mentioning Math in that analysis.
For example, in [Nolan and Lang 2009] analyze how to integrate Computational Thinking into the curriculum of university statistics. In [Isbell et al. 2010], the authors redefines the discipline of computing as focused around the notion of modeling, principally those models that are automatable and automatically manipulable. As results, the authors proposed a curriculum that focused on modeling, scales and limits, simulation, abstraction, and automation. On the other hand, in [Djurdjevic-Pahl, Pahl, Fronza and El Ioini 2017], the authors proposed a pathway into Computing Education (CE) through Computational Thinking (CT), starting from traditional mathematics curricula for primary schools. In their proposal, in the first instance, they do not involve computer programming or the management of ICT, but instead, they seek to develop essential skills in computational thinking, such as abstraction and logical thinking. To finish, [Barcelos and Silveira 2012] propose three common competences between CT and Math from the analysis of Brazil and Chile's reference curricula, as well as the CSTA curriculum.

5 Discussion

From the analysis of the studies included in this revision, it is possible to infer that there is a growing interest in the scientific community in exploring the relationship between Computational Thinking and Math. A significant increase during the last years in the amount of studies in which didactical activities are presented and assessed empirically contributed to that conclusion. Although the concept of Computational Thinking originally refers to skills that should be developed by elementary school students, it was verified that the didactical activities described are not restricted only to that educational level: nearly 31% of the reported didactical activities were offered to university students. This indicates a tendency to use activities based on skills and concepts framed around the "Computational Thinking" term as an approach to mitigate evasion and dropout rates in careers associated to Computer Sciences. However, this can be considered as a "detour" in the original targets of Computational Thinking since university education is typically the closest field of application for educational research in Universities. Some didactical experiences presented in studies target the training of teachers to diffuse development of CT, however, such initiatives are very preliminary (only 3 studies from a universe of 42) and are in a need for bigger systematization.

The developed didactical activities are related to a wide range of mathematical contents. Most activities used computational tools, which are also very diverse. This is an indicator of great flexibility and potential of computational concepts and software tools as a support for teaching and contextualizing Math. This presents new evidences in comparison to those previously presented by [Grover and Pea 2013], who mentioned that Computational Thinking was not being used to teach other disciplines.

According to [Husén 1997], learning phenomenon is inherently complex, and therefore, its research demands a complementary use of the quantitative and qualitative paradigms. A consequence, from an operational point of view, is the need of gathering and analyzing data from different sources, in order to allow a more precise comprehension of the observed phenomena. In this sense, the methodological limitations identified by [Grover and Pea 2013] are also verified in the results of this systematic revision. In the group of studies with experimental evaluation, roughly half of the studies does not use multiple information data sources in order to analyze the results of teaching-learning process from different perspectives. Learning assessments composed by pre and post tests and analysis of classroom observations stand out as the most used data gathering tools. Brennan and Resnick [Brennan and Resnick 2012] suggest that the skills acquired by students are evidenced in artifacts built by them, a theory that was also evidenced previously by other authors [Basawapatna, Koh, Repenning, Webb and Marshall 2011, Denner, Werner and Ortiz 2012, Kafai, Franke, Ching and Shih 1998]. Despite most of didactic experiences carried out by students use tools for creation of computation gadgets, only six studies use analysis of the artifacts, digital or not, produced during the activities.

6 Conclusions

Computational Thinking represents a set of skills related to Computer Sciences that should be developed by elementary school students. This development has the potential of bringing some benefits to students; however, it becomes necessary to understand the relations between Computational Thinking and traditional disciplines of school curricula and what the possible benefits of joint didactic strategies are. In this article, we present a systematic revision of the literature with studies published between 2006 and 2017 that address relations between Computational Thinking and Math, whose results allow identifying the advances and limitations of the research in that area.

Most of the described didactic experiences have basic education students as target audience. However, a relevant amount of experiences was developed with university students. On the other hand, there are not enough reports of experiences developed for the initial and continuous training of teachers. A wide variety of mathematical topics is being developed, with a predominance concerning Planar Geometry and Algebra. The high-level skills shared between the two paradigms of thought that is mostly associated to the reported educational outcomes is the conversion between different semiotic representations; however, very few studies use the construction and assessment of mathematical and computation models. Physics is the topic mostly associated with such activity. Pre and post tests and classroom observation are the most used data gathering instruments, but half of the studies uses only a single data source for the assessment.

On one hand, it is possible to identify a clear advance in availability and variety of didactical activities that involve Computational Thinking and Math, as well as a higher methodological rigor in the assessment of results. However, there are still target audiences, mathematical skills and information sources that have been barely explored by scientific community. Through the identification of these limitations, further and deeper studies could be envisioned to fill such gaps.

References

[Aedo and Walker 2012] Aedo, C., Walker, I.: 'Skills for the 21st Century in Latin America and the Caribbean'; Washington, DC: The World Bank (2012).

[Alab, Magan and Garci 2013] Alab, O. O., Magan, A. J., Garci, R. E.: 'Exploring student representational approaches in solving rechargable battery design problems'; In 2013 IEEE Frontiers in Education Conference (FIE) (2013), pp. 1685–1687. https://doi.org/10.1109/FIE.2013.6685124

[Arroyo, Micciollo, Casano, Ottmar, Hulse and Rodrigo 2017] Arroyo, I., Micciollo, M., Casano, J., Ottmar, E., Hulse, T., Rodrigo, M. M.: 'Wearable Learning: Multiplayer Embodied Games for Math'; In Proceedings of the Annual Symposium on Computer-Human Interaction in Play. New York, NY, USA: ACM (2017), pp. 205–216. https://doi.org/10.1145/3116595.3116637

[Association for Computing Machinery, code.org, Computer Science Teachers Association, Cyber Innovation Center and National Math and Science Initiative 2016] Association for Computing Machinery, code.org, Computer Science Teachers Association, Cyber Innovation Center, National Math and Science Initiative: 'K12 Computer Science Framework'; (2016). Retrieved from http://www.k12cs.org

[Barcelos and Silveira 2012] Barcelos, T. S., Silveira, I. F.: 'Teaching Computational Thinking in initial series: An analysis of the confluence among mathematics and Computer Sciences in elementary education and its implications for higher education'; Presented at the 2012 XXXVIII Conferencia Latinoamericana En Informatica (CLEI) (2012). https://doi.org/10.1109/CLEI.2012.6427135

[Barcelos and Silveira 2013] Barcelos, T., Silveira, I. F.: 'Computational Thinking and Mathematics: possible relationships revealed by an analysis of national curriculum guidelines'; In V. C. X. Wang (Ed.), Encyclopedia of Education and Technology in a Changing Society (1a). Hershey: IGI Global (2013).

[Basawapatna, Koh, Repenning, Webb and Marshall 2011] Basawapatna, A., Koh, K. H., Repenning, A., Webb, D. C., Marshall, K. S.: 'Recognizing computational thinking patterns'; In Proceedings of the 42nd ACM technical symposium on Computer science education. New York: ACM (2011), pp. 245–250. https://doi.org/10.1145/1953163.1953241

[Biró, Csernoch, Máth and Abari 2015] Biró, P., Csernoch, M., Máth, J., Abari, K.: 'Measuring the Level of Algorithmic Skills at the End of Secondary Education in Hungary'; Procedia - Social and Behavioral Sciences, Vol. 176 (2015), pp. 876–883. https://doi.org/10.1016/j.sbspro.2015.01.553

[Brennan and Resnick 2012] Brennan, K., Resnick, M.: 'New frameworks for studying and assessing the development of computational thinking'; In Proceedings of the 2012 annual meeting of the American Educational Research Association. Vancouver: American Educational Research Association (2012).

[Buteau and Muller 2017] Buteau, C., Muller, E.: ‘Assessment in Undergraduate Programming-Based Mathematics Courses’; Digital Experiences in Mathematics Education, Vol. 3, No. 2 (2017), pp. 97–114. https://doi.org/10.1007/s40751-016-0026-4

[Chiu, Malcolm, Hecht, DeJaegher, Pan, Bradley and Burghardt 2013] Chiu, J. L., Malcolm, P. T., Hecht, D., DeJaegher, C. J., Pan, E. A., Bradley, M., Burghardt, M. D.: ‘WISEngineering: Supporting precollege engineering design and mathematical understanding’; Computers & Education, Vol. 67, No. 0 (2013), pp. 142–155. https://doi.org/10.1016/j.compedu.2013.03.009

[Common Core State Standards Initiative 2012] Common Core State Standards Initiative: ‘Commom Core State State Standards - Math’; (2012). Retrieved 30 June 2017, from http://www.corestandards.org/Math

[Computer Science Teachers Association 2011] Computer Science Teachers Association: ‘Operational Definition of Computational Thinking for K-12 Education’; (2011). Retrieved from https://csta.acm.org/Curriculum/sub/CurrFiles/CompThinkingFlyer.pdf

[Computer Science Teachers Association 2017] Computer Science Teachers Association: ‘CSTA K-12 Computer Science Standards, revised 2017’; New York: ACM Computer Science Teachers Association (2017). Retrieved from http://www.csteachers.org/page/standards

[Creswell and Miller 2000] Creswell, J. W., Miller, D. L.: ‘Determining validity in qualitative inquiry’; Theory Into Practice, Vol. 39, No. 3 (2000), pp. 124–130.

[Denner, Werner and Ortiz 2012] Denner, J., Werner, L., Ortiz, E.: ‘Computer games created by middle school girls: Can they be used to measure understanding of computer science concepts?’; Computers & Education, Vol. 58, No. 1 (2012), pp. 240–249. https://doi.org/10.1016/j.compedu.2011.08.006

[Denning 2005] Denning, P. J.: ‘Is computer science science?’; Communications of the ACM, Vol. 48, No. 4 (2005), pp. 27–31. https://doi.org/10.1145/1053291.1053309

[Djurdjevic-Pahl, Pahl, Fronza and El Ioini 2017] Djurdjevic-Pahl, A., Pahl, C., Fronza, I., El Ioini, N.: ‘A Pathway into Computational Thinking in Primary Schools’; In T.-T. Wu, R. Gennari, Y.-M. Huang, H. Xie & Y. Cao (Eds.), Emerging Technologies for Education (Vol. 10108). Cham: Springer International Publishing (2017), pp. 165–175. https://doi.org/10.1007/978-3-319-52836-6_19

[Dyne and Braun 2014] Dyne, M. V., Braun, J.: ‘Effectiveness of a computational thinking (CS0) course on student analytical skills’; In Proceedings of the 45th ACM technical symposium on Computer science education. Atlanta, Georgia, USA: ACM (2014), pp. 133–138.

[European Comission. Directorate-General for Education and Culture. 2004] European Comission. Directorate-General for Education and Culture.: ‘Competencias clave para un aprendizaje ao lo largo de la vida: un marco de referencia europeo’; (2004). Retrieved from http://www.educastur.princast.es/info/calidad/indicadores/doc/comision_europea.pdf

[European Schoolnet 2015] European Schoolnet: ‘Computing our future: computer programming and coding. Priorities, school curricula and initiatives across Europe.’; (2015). Retrieved from https://www.researchgate.net/publication/284139559_Computing_our_future_Computer_programming_and_coding_-_Priorities_school_curricula_and_initiatives_across_Europe

[Grover and Pea 2013] Grover, S., Pea, R.: ‘Computational Thinking in K–12: A Review of the State of the Field’; Educational Researcher, Vol. 42, No. 1 (2013), pp. 38–43. https://doi.org/10.3102/0013189X12463051

[Hanushek, Peterson and Woessmann 2010] Hanushek, E. A., Peterson, P. E., Woessmann, L.: 'U.S. math performance in global perspective: how well does each state do at producing high-achieving students?'; Cambridge, MA: Program on Education Policy and Governance & Education, Harvard University Kennedy School (2010). Retrieved from http://hanushek.stanford.edu/sites/default/files/publications/Hanushek%2BPeterson%2BWoessmann%202010%20PEPG%20report.pdf

[Hauze, French, Castañeda-Emenaker, French and Singer 2017] Hauze, S., French, D., Castañeda-Emenaker, I., French, M., Singer, T.: 'Quantifying K-12 and college student learning outcomes of STEM guitar building'; In 2017 IEEE Integrated STEM Education Conference (ISEC) (2017), pp. 121–126. https://doi.org/10.1109/ISECon.2017.7910226

[Hemmendinger 2010] Hemmendinger, D.: 'A plea for modesty'; ACM Inroads, Vol. 1, No. 2 (2010), pp. 4–7. https://doi.org/10.1145/1805724.1805725

[Hoji, Vianna and Felix 2012] Hoji, E. S., Vianna, W. B., Felix, T. D. A.: 'A computer-aided math teaching approach for students in a technical institute: The experience with the Octave in the electro-mechanical technical course'; In Interactive Collaborative Learning (ICL), 2012 15th International Conference on. IEEE (2012), pp. 1–5. https://doi.org/10.1109/ICL.2012.6402119

[Hood and Hood 2005] Hood, C. S., Hood, D. J.: 'Toward integrating computing concepts into the K-12 curriculum'; SIGCSE Bulletin, Vol. 37, No. 3 (2005), pp. 375–375. https://doi.org/10.1145/1151954.1067576

[Hu 2011] Hu, C.: 'Computational thinking: what it might mean and what we might do about it'; In Proceedings of the 16th annual joint conference on Innovation and technology in computer science education. New York, NY, USA: ACM (2011), pp. 223–227. https://doi.org/10.1145/1999747.1999811

[Husén 1997] Husén, T.: 'Research paradigms in education'; In Educational research, methodology and measurement: an international handbook. Oxford: Pergamond (1997).

[Isbell, Stein, Cutler, Forbes, Fraser, Impagliazzo, et al. 2010] Isbell, C. L., Stein, L. A., Cutler, R., Forbes, J., Fraser, L., Impagliazzo, J., et al.: '(Re)defining computing curricula by (re)defining computing'; SIGCSE Bulletin, Vol. 41, No. 4 (2010), pp. 195–207. https://doi.org/10.1145/1709424.1709462

[Jonassen 2000] Jonassen, D. H.: 'Computers as Mindtools for Schools: Engaging Critical Thinking'; (2nd edition). Peason (2000).

[Kafai, Franke, Ching and Shih 1998] Kafai, Y. B., Franke, M. L., Ching, C. C., Shih, J. C.: 'Game Design as an Interactive Learning Environment for Fostering Students' and Teachers' Mathematical Inquiry'; International Journal of Computers for Mathematical Learning, Vol. 3, No. 2 (1998), pp. 149–184. https://doi.org/10.1023/A:1009777905226

[Ke 2014] Ke, F.: 'An implementation of design-based learning through creating educational computer games: A case study on mathematics learning during design and computing'; Computers & Education, Vol. 73 (2014), pp. 26–39. https://doi.org/10.1016/j.compedu.2013.12.010

[Kitchenham 2004] Kitchenham, B.: 'Procedures for performing systematic reviews' (Technical Report No. TR/SE-0401); UK: Keele University (2004).

[Krone, Sitaraman and Hallstrom 2011] Krone, J., Sitaraman, M., Hallstrom, J. O.: 'Mathematics throughout the CS curriculum'; Journal of Computing Sciences in Colleges, Vol. 27, No. 1 (2011), pp. 65–73.

[Lee, Martin, Denner, Coulter, Allan, Erickson, et al. 2011] Lee, I., Martin, F., Denner, J., Coulter, B., Allan, W., Erickson, J., et al.: 'Computational thinking for youth in practice'; ACM Inroads, Vol. 2, No. 1 (2011), pp. 32–37. https://doi.org/10.1145/1929887.1929902

[León, Robles and Román-González 2016] León, J. M., Robles, G., Román-González, M.: 'Code to Learn: Where Does It Belong in the K-12 Curriculum?'; Journal of Information Technology Education: Research, Vol. 15 (2016), pp. 283–303. https://doi.org/10.28945/3521

[Lewis and Shah 2012] Lewis, C. M., Shah, N.: 'Building upon and enriching grade four mathematics standards with programming curriculum'; In Proceedings of the 43rd ACM technical symposium on Computer Science Education. New York: ACM (2012), pp. 57–62. https://doi.org/10.1145/2157136.2157156

[Mor and Noss 2008] Mor, Y., Noss, R.: 'Programming as mathematical narrative'; International Journal of Continuing Engineering Education and Life-Long Learning, Vol. 18, No. 2 (2008), pp. 214–233.

[Moreno-León and Robles 2016] Moreno-León, J., Robles, G.: 'Code to learn with Scratch? A systematic literature review'; In 2016 IEEE Global Engineering Education Conference (EDUCON) (2016), pp. 150–156. https://doi.org/10.1109/EDUCON.2016.7474546

[Nolan and Lang 2009] Nolan, D., Lang, D. T.: 'Approaches to Broadening the Statistics Curricula'; In M. C. Shelley, L. D. Yore & B. Hand (Eds.), Quality Research in Literacy and Science Education. Dordrecht: Springer Netherlands (2009), pp. 357–381. https://doi.org/10.1007/978-1-4020-8427-0_18

[Oliveira 2012] Oliveira, O. L.: 'Statistical Evidence of the Correlation Between Mental Ability to Compute and Student Performance in Undergraduate Courses'; In Proceedings of the 17th ACM Annual Conference on Innovation and Technology in Computer Science Education. New York, NY, USA: ACM (2012), pp. 111–115. https://doi.org/10.1145/2325296.2325326

[Papert 1980] Papert, S.: 'Mindstorms: children, computers and powerful ideas'; New York: Basic Books (1980).

[Perrenoud, Thurler, de Macedo, Machado and Allessandrini 2007] Perrenoud, P., Thurler, M. G., de Macedo, L., Machado, N. J., Allessandrini, C. D.: 'As competências para ensinar no século XXI: a formação dos professores e o desafio da avaliação'; Porto Alegre: Artmed (2007).

[Pioro 2006] Pioro, B. T.: 'Introductory computer programming: gender, major, discrete mathematics, and calculus'; J. Comput. Sci. Coll., Vol. 21, No. 5 (2006), pp. 123–129.

[Polya 2004] Polya, G.: 'How to solve it: a new aspect of mathematical method'; Princeton: Princeton University Press (2004).

[Ralston 2005] Ralston, A.: 'Do we need ANY mathematics in computer science curricula?'; SIGCSE Bull., Vol. 37, No. 2 (2005), pp. 6–9. https://doi.org/10.1145/1083431.1083433

[Sengupta, Kinnebrew, Basu, Biswas and Clark 2013] Sengupta, P., Kinnebrew, J., Basu, S., Biswas, G., Clark, D.: 'Integrating computational thinking with K-12 science education using agent-based computation: A theoretical framework'; Education and Information Technologies, Vol. 18, No. 2 (2013), pp. 351–380. https://doi.org/10.1007/s10639-012-9240-x

[Sengupta, Krishnan, Wright and Ghassoul 2015] Sengupta, P., Krishnan, G., Wright, M., Ghassoul, C.: 'Mathematical Machines and Integrated Stem: An Intersubjective Constructionist Approach'; In S. Zvacek, M. T. Restivo, J. Uhomoibhi & M. Helfert (Eds.), Computer Supported Education. Cham: Springer International Publishing (2015), pp. 272–288.

[Taylor, Harlow and Forret 2010] Taylor, M., Harlow, A., Forret, M.: 'Using a Computer Programming Environment and an Interactive Whiteboard to Investigate Some Mathematical Thinking'; Procedia - Social and Behavioral Sciences - International Conference on Mathematics Education Research 2010 (ICMER 2010), Vol. 8, No. 0 (2010), pp. 561–570. https://doi.org/10.1016/j.sbspro.2010.12.078

[The CSTA Standards Task Force 2011] The CSTA Standards Task Force: 'CSTA K-12 Computer Science Standards'; New York: ACM Computer Science Teachers Association (2011). Retrieved from http://csta.acm.org/Curriculum/sub/CurrFiles/CSTA_K-12_CSS.pdf

[Tort, Blondel and Bruillard 2008] Tort, F., Blondel, F.-M., Bruillard, É.: 'Spreadsheet Knowledge and Skills of French Secondary School Students'; In R. Mittermeir & M. Sysło (Eds.), Informatics Education - Supporting Computational Thinking (Vol. 5090). Springer Berlin Heidelberg (2008), pp. 305–316. Retrieved from http://dx.doi.org/10.1007/978-3-540-69924-8_28

[Trochim and Donelly 2006] Trochim, W. M. K., Donelly, J. P.: 'Research methods knowledge base'; (2006). Retrieved from http://www.socialresearchmethods.net/kb

[Wholin, P, Höst, Ohlsson, Regnell and Wesslön 2000] Wholin, C., P, R., Höst, M., Ohlsson, M., Regnell, B., Wesslön, A.: 'Experimentation in Software Engineering: an Introduction'; USA: Kluwer Academic Publishers (2000).

[Wilkerson-Jerde 2014] Wilkerson-Jerde, M. H.: 'Construction, categorization, and consensus: student generated computational artifacts as a context for disciplinary reflection'; Educational Technology Research and Development, Vol. 62, No. 1 (2014), pp. 99–121. https://doi.org/10.1007/s11423-013-9327-0

[Wing 2006] Wing, J. M.: 'Computational thinking'; Communications of the ACM, Vol. 49, No. 3 (2006), pp. 33–35. https://doi.org/10.1145/1118178.1118215

[Wohlin, Runeson, Höst, Ohlsson, Regnell and Wesslén 2012] Wohlin, C., Runeson, P., Höst, M., Ohlsson, M., Regnell, B., Wesslén, A.: 'Systematic Literature Reviews'; In Experimentation in Software Engineering. Springer Berlin Heidelberg (2012), pp. 45–54. Retrieved from http://dx.doi.org/10.1007/978-3-642-29044-2_4

Appendix

Authors	Year	Educational Level	Sample	Experimental Design	Competencies	Mathematical Contents
[Jacobs 2009]	2009	Middle	100	experience report	ABDSR, BDARM	Arithmetic, Statistics, Algebra
[Freudenthal, Roy, Ogrey, Magoc and Siegel 2010]	2010	Undergraduate	N/S	experience report	ABDSR, ERAIP, BDARM	Physics, Calculus
[Taylor et al. 2010]	2010	Elementary	60	qualitative case study	ABDSR, BDARM	Euclidean Geometry, Linear
[Ahamed et al. 2010]	2010	Middle, Teacher	12	quasi-experiment	ABDSR, ERAIP, BDARM	Statistics, Physics
[Boyce, Campbell, Pickford, Culler and Barnes 2011]	2011	Middle, High School	41	experiment	ABDSR, ERAIP	Linear
[Cervesato 2011]	2011	Undergraduate	27	experience report	ABDSR	Algebra
[Rizvi, Humphries, Major, Lauzun and Jones 2011]	2011	Undergraduate	35	quasi-experiment	ABDSR	Euclidean Geometry, Linear
[Bryant, Weiss, Orr and Yerion 2011]	2011	Undergraduate	145	quasi-experiment	ABDSR, ERAIP	Euclidean Geometry
[Lewis and Shah 2012]	2012	Middle	47	quasi-experiment	ABDSR	Euclidean Geometry, Algebra

Competencies: ABDS = Alternating between different semiotic representations; ERAIP = Establishing relationships and identifying patterns; BDARM = Building descriptive and representative models. N/E: Not specified with detailed

Table 8: Selected studies that present didactical experiences (EXP) (Part – 1)

[Hoji et al. 2012]	2012	Undergraduate	25	quasi-experiment	BDARM	Linear, Physics, Algebra
[Gibson 2012]	2012	Elementary, High School	N/S	experience report	ABDSR, ERAIP	Algebra
[Oliveira 2012]	2012	Undergraduate	79	quasi-experiment	ERAIP, BDARM	Math model
[Hsi and Eisenberg 2012]	2012	Elementary, Middle	21	quasi-experiment	ABDSR	Euclidean Geometry
[Ke and Im 2014]	2013	Middle	64	qualitative case study	ABDSR	Arithmetic
[Chiu et al. 2013]	2013	Middle	70	quasi-experiment	ABDSR	Euclidean Geometry, Arithmetic, Algebra
[Wilkerson-Jerde 2014]	2013	Middle	20	quasi-experiment	ABDSR	Euclidean Geometry
[Alab et al. 2013]	2013	Graduate	6	quasi-experiment	BDARM	Physics, Calculus
[Sengupta et al. 2013]	2013	Middle	24	quasi-experiment	ABDSR, BDARM	Physics
[Magana 2014]	2014	Undergraduate	224	quasi-experiment	ABDSR	Euclidean Geometry
[Ke 2014b]	2014	Middle	64	qualitative case study	ABDSR, BDARM	Math model, Arithmetic
[Van Dyne and Braun 2014]	2014	Undergraduate	51	quasi-experiment	ABDSR	Algebra
[Kyriakides, Meletiou-Mavrotheris and Prodromou 2016]	2015	Elementary	15	qualitative case study	ABDSR	Euclidean Geometry

Table 9: Selected studies that present didactical experiences (EXP) (Part – 2)

[Calao, Moreno-León, Correa and Robles 2015]	2015	Middle	42	quasi-experiment	BDARM	Math model
[Bean, Weese, Feldhausen and Bell 2015]	2015	Undergraduate	33	quasi-experiment	ABDSR	Euclidean Geometry
[Biró et al. 2015]	2015	Undergraduate	950	quasi-experiment	ABDSR, ERAIP	Arithmetic
[Sengupta et al. 2015]	2015	Elementary	30	qualitative	ABDSR, BDARM	Euclidean Geometry
[Buteau and Muller 2017b]	2016	Undergraduate	45	experience report	ERAIP, BDARM	Math model
[Basu et al. 2016]	2016	Middle	15	quasi-experiment	BDARM	Physics
[Gaio and Di Paola 2017]	2017	Elementary, Middle	370	experience report	ABDSR, ERAIP	Algebra
[Sullivan, Strawhacker and Bers 2017]	2017	Kinder, Elementary	10	experience report	ABDSR	Euclidean Geometry
[Simpson, Burris and Maltese 2017]	2017	Elementary	11	qualitative	ABDSR	Math model, Arithmetic
[Jones-Harris and Chamblee 2017]	2017	High School, Undergraduate	6	qualitative mixed-methods	ABDSR	Euclidean Geometry, Algebra, Calculus
[Cesar et al. 2017]	2017	Master	30	experience report	ABDSR, ERAIP, BDARM	Math model, Statistics
[Merritt, Chiu, Peters-Burton and Bell 2017]	2017	Teacher	19	qualitative case study	ABDSR	Arithmetic
[Costa, Campos and Dario Serey Guerrero 2017]	2017	Middle	46	quasi-experiment	ABDSR, ERAIP, BDARM	Statistics

Table 10: Selected studies that present didactical experiences (EXP) (Part – 3)

[Arroyo et al. 2017]	2017	Elementary	53	experiment	ABDSR	Euclidean Geometry
[Tsouccas and Meletiou-Mavrotheris 2017]	2017	Elementary	15	qualitative	ABDSR	Euclidean Geometry
[Niemelä, Partanen, Harsu, Leppänen and Ihantola 2017]	2017	Teacher	540	qualitative mixed-methods	ABDSR, BDARM	Math model, Planar Geometry, Arithmetic, Algebra
[Ziaeefard, Miller, Rastgaar and Mahmoudian 2017]	2017	Middle, High School	127	quasi-experiment	ABDSR, BDARM	Euclidean Geometry
[Psycharis and Kallia 2017]	2017	High School	66	quasi-experiment	ABDSR	Math model
[Hauze et al. 2017]	2017	Middle, High School, Undergraduate	769	quasi-experiment	ABDSR	Euclidean Geometry
[Sung, Ahn and Black 2017]	2017	Kinder, Elementary	66	experiment	ABDSR	Math model, Euclidean Geometry

Table 8: Selected studies that present didactical experiences (EXP) (Part – 4)

Authors	**DOI**	**Title**
[Jacobs 2009]	10.1007/978-3-642-01973-9_3	Building Excitement, Experience and Expertise in Computational Science among Middle and High School Students
[Freudenthal et al. 2010]	10.1145/1734263.1734276	MPCT: media propelled computational thinking
[Taylor et al. 2010]	10.1016/j.sbspro.2010.12.078	Using a Computer Programming Environment and an Interactive Whiteboard to Investigate Some Mathematical Thinking
[Ahamed et al. 2010]	10.1145/1734263.1734277	Computational thinking for the sciences: a three day workshop for high school science teachers
[Boyce et al. 2011]	10.1145/1999747.1999816	Experimental evaluation of BeadLoom game: how adding game elements to an educational tool improves motivation and learning
[Cervesato 2011]	10.1145/1999747.1999778	Discovering logic through comics
[Rizvi et al. 2011]	10.1109/CSEET.2011.5876101	A new CS0 course for at-risk majors
[Bryant et al. 2011]	ISSN: 1937-4771	Using the context of algorithmic art to change attitudes in introductory programming
[Lewis and Shah 2012]	10.1145/2157136.2157156	Building upon and enriching grade four mathematics standards with programming curriculum
[Hoji et al. 2012]	10.1109/ICL.2012.6402119	A computer-aided math teaching approach for students in a technical institute: The experience with the Octave in the electro-mechanical technical course
[Gibson 2012]	10.1145/2325296.2325308	Teaching graph algorithms to children of all ages

Table 9: Title and DOI selected studies that present didactical experiences (EXP) (Part – 1)

[Oliveira 2012]	10.1145/2325296.2325326	Statistical evidence of the correlation between mental ability to compute and student performance in undergraduate courses
[Hsi and Eisenberg 2012]	10.1145/2307096.2307137	Math on a sphere: using public displays to support children's creativity and computational thinking on 3D surfaces
[Ke and Im 2014]	10.1007/s10798-013-9248-6	A case study on collective cognition and operation in team-based computer game design by middle-school children
[Chiu et al. 2013]	10.1016/j.compedu.2013.03.009	WISEngineering: Supporting precollege engineering design and mathematical understanding
[M. Wilkerson-Jerde 2014]	10.1007/s11423-013-9327-0	Construction, categorization, and consensus: student generated computational artifacts as a context for disciplinary reflection
[Alab et al. 2013]	10.1109/FIE.2013.6685124	Exploring student representational approaches in solving rechargable battery design problems
[Sengupta et al. 2013]	10.1007/s10639-012-9240-x	Integrating computational thinking with K-12 science education using agent-based computation: A theoretical framework
[Magana 2014]	10.1016/j.compedu.2013.11.012	Learning strategies and multimedia techniques for scaffolding size and scale cognition
[Ke 2014b]	10.1016/j.compedu.2013.12.010	An implementation of design-based learning through creating educational computer games: A case study on mathematics learning during design and computing
[Van Dyne and Braun 2014]	10.1145/2538862.2538956	Effectiveness of a computational thinking (CS0) course on student analytical skills
[Kyriakides et al. 2016]	10.1007/s13394-015-0163-x	Mobile technologies in the service of students' learning of mathematics: the example of game application A.L.E.X. in the context of a primary school in Cyprus
[Calao, Moreno-León, Correa and Robles 2015]	10.1007/978-3-319-24258-3_2	Developing Mathematical Thinking with Scratch

Table 10: Title and DOI selected studies that present didactical experiences (EXP) (Part – 2)

[Bean et al. 2015]	10.1109/FIE.2015.7344237	Starting from scratch: Developing a pre-service teacher training program in computational thinking
[Biró et al. 2015]	10.1016/j.sbspro.2015.01.553	Measuring the Level of Algorithmic Skills at the End of Secondary Education in Hungary
[Sengupta et al. 2015]	10.1007/978-3-319-25768-6_18	Mathematical Machines and Integrated Stem: An Intersubjective Constructionist Approach
[Buteau and Muller 2017b]	10.1007/s40751-016-0026-4	Assessment in Undergraduate Programming-Based Mathematics Courses
[Basu et al. 2016]	10.1186/s41039-016-0036-2	Identifying middle school students' challenges in computational thinking-based science learning
[Gaio and Di Paola 2017]	10.1007/978-3-319-70308-4_3	Discrete Mathematics in Lower School Grades? Situation and Possibilities in Italy
[Sullivan et al. 2017]	10.1007/978-3-319-57786-9_10	Dancing, Drawing, and Dramatic Robots: Integrating Robotics and the Arts to Teach Foundational STEAM Concepts to Young Children
[Simpson et al. 2017]	10.1007/s11165-017-9678-3	Youth's Engagement as Scientists and Engineers in an Afterschool Making and Tinkering Program
[Jones-Harris and Chamblee 2017]	10.1007/978-3-319-52691-1_3	Understanding African-American Students' Problem-Solving Ability in the Precalculus and Advanced Placement Computer Science Classroom
[Cesar et al. 2017]	10.1016/j.jpdc.2016.12.027	Introducing computational thinking, parallel programming and performance engineering in interdisciplinary studies
[Merritt et al. 2017]	10.1007/s11165-016-9604-0	Teachers' Integration of Scientific and Engineering Practices in Primary Classrooms
[Costa et al. 2017]	10.1109/FIE.2017.8190655	Computational thinking in mathematics education: A joint approach to encourage problem-solving ability

Table 11: Title and DOI selected studies that present didactical experiences (EXP) (Part – 3)

[Arroyo et al. 2017]	10.1145/3116595.3116637	Wearable Learning: Multiplayer Embodied Games for Math
[Tsouccas and Meletiou-Mavrotheris 2017]	10.1145/3136907.3136951	Enhancing the Technological, Pedagogical and Content Knowledge (TPACK) of in-service primary teachers in the use of tablet technologies
[Niemelä, Partanen, Harsu, Leppänen and Ihantola 2017]	10.1145/3141880.3141885	Computational thinking as an emergent learning trajectory of mathematics
[Ziaeefard et al. 2017]	10.1016/j.robot.2017.07.013	Co-robotics hands-on activities: A gateway to engineering design and STEM learning
[Psycharis and Kallia 2017]	10.1007/s11251-017-9421-5	The effects of computer programming on high school students' reasoning skills and mathematical self-efficacy and problem solving
[Hauze et al. 2017]	10.1109/ISECon.2017.7910226	Quantifying K-12 and college student learning outcomes of STEM guitar building
[Sung et al. 2017]	10.1007/s10758-017-9328-x	Introducing Computational Thinking to Young Learners: Practicing Computational Perspectives Through Embodiment in Mathematics Education

Table 12: Title and DOI selected studies that present didactical experiences (EXP) (Part – 4)