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

Thiago Schumacher Barcelos [1,2] and Ismar Frango Silveira [2]
[1] Instituto Federal de Educação, Ciência e Tecnologia de São Paulo – IFSP, Brazil. tsbarcelos@ifsp.edu.br
[2] Universidade Cruzeiro do Sul, Brazil. ismar.silveira@cruzeirodosul.edu.br

***Abstract* — The pervasiveness of computer devices in everyday situations poses a fundamental question about Computer Sciences as being part of those known as basic sciences. However, it would be more beneficial not to consider computation only as a technique, but instead as a way of reasoning and problem solving. Under this perspective, there are inherent relationships among the knowledge, skills and attitudes that emanate from this field and those ones commonly related to Math. This paper discusses the relationship between the so-named Computational Thinking and the foundations of Math Education, based on a literature review. Three groups of skills that can be jointly developed by both areas are identified and some challenges and implications for education in Computer Sciences are discussed.**

***Keywords* – Computational thinking, Mathematics, didactic strategies**

I. Introduction

The lack of adequate domain of mathematical knowledge by students is a possible factor for the lack of interest and high dropout rates in Computer Sciences undergraduate courses. A brief literature review [1–4] helps to reaffirm such hypothesis, as several researchers indicate there might be some sort of correlation between students' prior mathematical knowledge and their performance in introductory courses in Computer Sciences. The relevance of math topics for a better understanding and modeling of computational processes is also discussed in the literature. In Latin America, poor educational attainments in Mathematics are a commonplace for students in basic education. As an example, the results of last Brazilian SARESP Exam (*Sistema de Avaliação de Rendimento Escolar do Estado de São Paulo* - Evaluation System for Educational Attainment in São Paulo state) indicate that 58.4% of students leaving high school have a performance in math topics which is considered to be under the minimum level expected [5]. Chilean's SIMCE (*Sistema de Medición de Calidad de la Educación* – System for Measurement of Education Quality) test was analyzed by [6], which stated that the inequality of opportunities plays an important role in poor results.

Regarding to international exams such as PISA (Program for International Student Assessment), even though Latin American countries have improved over time, they still remain among the worst performers in Mathematics (the same is true for reading). Figure 1, extracted from [7], compares the other OECD (Organization for Economic Co-operation Development) participating countries' averages to Argentina, Brazil, Chile, Mexico and Peru averages. It must be noted that in 2009 the difference between Latin American and OECD scores for Mathematics was around 100 PISA points, which is equivalent to a lag of two years of education.

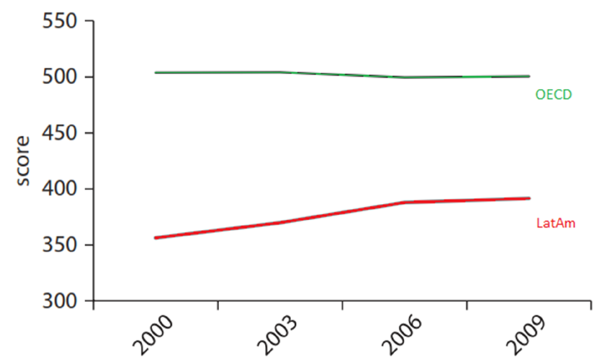

Figure 1. Evolution of Latin American versus OECD counties in PISA Mathematics test

Figure 2, also extracted from [7], shows in more detail the performance of some selected Latin American countries in PISA Mathematics exam.

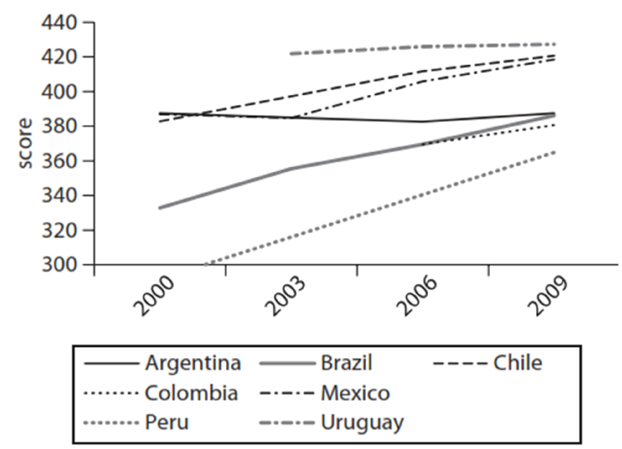

Figure 2. PISA evolution in Mathematics for Latin America, by country

The poor levels of educational assessments can be partially explained by the historically low economical levels of Latin American countries compared to other OECD countries. However, even with the economical growth perceived by this region in the last years, educational results did not yet improve at the same proportion of countries' economies.

Nonetheless, even in developed countries the difficulties with Mathematics learning are found. For instance, the RSA (Royal Society of Arts) in United Kingdom recently found out that some Universities are not publishing the mathematical prerequisites necessary for enrollment in their courses to avoid losing some potential students [8].

On the other hand, the Computer Science Education community has recently started to consider CS as a subject that should be part of the school curriculum since the 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 Computer Sciences, 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, [9] affirms that "a key to achieving widespread fluency on Information Technology is to make it part of the K-12 curriculum". However, there are some authors that even question whether Computer Sciences is a "real" science or not, as Ophir [10] states, given the industrial and commercial pressure under which such science is meant to bring to light new achievements disguised as new products.

Clearly, the pervasiveness of computer-based devices in everyday situations poses a fundamental question about Computer Sciences being part of those known as basic sciences. By doing this, it would be straightforward to consider it to be taught in initial series as a strategy to foster the students' interest for careers in Computing. However, such skills should not exactly be taught as a collection of techniques, but instead as a way of organizing thinking for concrete problem solving.

For this reason, a subset of skills and basic skills related to the area of Computer Sciences should be developed by students since the earlier series of elementary school. Such subset of skills and abilities are named by Wing [11] as *Computational Thinking* (CT). This term is currently used to describe the cognitive processes related to abstraction and problem decomposition to allow their resolution using computational resources and algorithmic strategies, among other skills.

When considering the incorporation of computational thinking to basic education, some questions arise: would this new competence support a more effective learning of critical areas like Mathematics? Would it be possible to achieve some kind of "competence transfer" among different knowledge domains? Would a curriculum that merge computational thinking and mathematics eventually attract more talented students to future careers in Computer Sciences and Technology?

In this sense, the present paper presents and discusses some research trends related to the teaching of Computer Sciences as a basic science, discussing its possible relationships with Mathematics Education. Thus, its main contribution is the identification of some research paths in which both areas can benefit from each other in the joint development of teaching strategies. The paper is organized as follows: in Section II we briefly situate how Mathematics and other knowledge areas have been used to support the development of Computer Sciences; then, we present some tentative definitions of Computational Thinking found in literature and show their relationship with expected competencies and skills to be developed by students. In Section III, we identify and discuss three competencies for Math education defined in Latin American guidelines which can be directed related to Computational Thinking skills. We present some educational activities found in literature which are associated to each one of the three selected competencies. In Section IV, we discuss possible research paths based on the proposed organization of CT skills, and in Section V we present our final considerations.

II. The Nature of Computer Sciences and Computational Thinking

The technological advances of computing over the XX and XXI centuries have a very close relationship to the feasibility of numerical calculations. The classic paper written in 1936 by Alan Turing, "On Computable Numbers, with an application to the Entscheidungsproblem" defines the structural basis of computing as it is known nowadays by presenting a mathematical proof about the feasibility of performing numerical calculations with a conceptual machine, later called the Universal Turing Machine. However, other knowledge areas were also incorporated to Computing in order to allow the construction of electronic computing machines and further development of programming languages and compilers. Hence, the field of Computing has built its Body of Knowledge on top of diverse knowledge areas. Denning [12] argues that, throughout its development and maturation, the activities of Computing bear themselves on the so-said Natural Sciences, and also on Engineering and Mathematics. The experimentation-based scientific method is used, for instance, to develop heuristic algorithms or to define models for the software development processes. On the other hand, software design and development are clearly Engineering activities, since they are applications of some well-known techniques. The symbolic representation and axiom-based deduction system from Mathematics are the basis for the study of algorithms complexity and numerical analysis.

In spite of having its foundations dwelt in other areas, Computing has singular reasoning mechanics for problem solving, whose applications go beyond the frontiers of Computing itself – one may consider, for instance, recent applications of Computing to research questions of Health or Social Sciences, allowing the analysis of an amount of data far superior than it would be possible before tackling such issues from a computational point of view.

Other knowledge areas can benefit from some specific Computing skills. Some areas are closely related to Computing, such as game design [13], while others are not, such as biology, chemistry and physics [14], [15]. However, in both cases it is convenient to identify such skills. As mentioned before, Wing [11] proposes to collectively name these specific skills with the term Computational Thinking, defined by the following characteristics:

- *Conceptualizing instead of programming*. Solving a problem by applying Computational Thinking can reduce big and seemingly intractable problems, breaking them down into smaller ones which are simpler to solve. This requires the ability to think abstractly and at multiple levels. One may notice that this requires more than simply applying programming techniques;

- *It is a fundamental skill, not an utilitarian one*. Computational Thinking skills are not utilitarian or mechanical; instead, they allow the resolution of many problems using computers, which are nowadays an ubiquitous feature. Therefore, these skills should be developed by all students in every field of knowledge;

- *It is the way human beings think, not computers*. Problem solving through computational thinking consists of a singular approach to representing a problem so that it can be solved by computers. Therefore, it is not a reduction of reasoning to merely simulate computer processing;

- *It complements and combines Mathematical and Engineering Thinking*. The definition in [11] considers the contribution of Mathematics and Engineering to Computing, as mentioned before, but also recognizes that Computer Sciences can expand the perspective and possibilities of those fields;

- *It generates ideas, not artifacts*. Computational Thinking is not necessarily a pathway to producing software or hardware artifacts. It indicates that the fundamental concepts of Computing are present to solve problems in diverse contexts of everyday life – even in the absence of computational devices;

- *For everyone, everywhere*. Finally, computational thinking can be considered to be useful for all persons in various applications.

Wing also argues that introducing Computational Thinking in formal education also means fostering the interest for Computing, showing its versatility and relevance to solve world's problems. Of course, not all students will choose a degree in Computing, but this educational strategy could create and intensify interdisciplinary relationships among Computing and other areas. One can even speculate if, by adopting this strategy, schools and universities could contribute to educating better computer scientists in the future.

The notion of Computational Thinking introduced by Wing has been heavily discussed and detailed by other authors. Hu [17] argues that Wing [11] does not directly define Computational Thinking by pointing out, for instance, what would be its thinking elements, structures or traits. However, the value of a precise definition is also questioned. For the author, it is possible that some aspects of Computational Thinking have been used even before the existence of modern computers. In this sense, CT may be a mixture of thinking elements previously related to other thinking paradigms, such as logical, math, analytical or engineering-oriented thinking. Hence, *doing* computing should be a way of enhancing Computational Thinking skills.

This activity-oriented view of CT can also be observed in the discussion by Lee *et al.* [16] about opportunities to develop learning environments for CT. Based on previous classroom experiences, the authors identify three domains which can be used to engage students in activities involving CT skills: modeling and simulation (e.g. of biological phenomena), robotics, and game design and development. Three fundamental skills were identified in the problem-solving approach used by students: abstraction, automation and analysis. In Table I, we present a brief summary of how these skills were applied in each of the three domains.

TABLE I. EXAMPLES OF COMPUTATIONAL THINKING APPLIED TO THREE DOMAINS (EXTRACTED FROM [16])

	Abstraction	Automation	Analysis
Modelling & Simulation	Selecting features of real-world to incorporate in a model	Time stepping using a model as an experimental testbed	Were the correct abstractions made? Does the model reflect reality?
Robotics	Design robot to react to a set of conditions	Program check sensors to monitor conditions	Are there situations that were not taken into account?
Game Design & Development	Games are abstracted into a set of scenes containing characters	Game respond to user actions	Do the elements incorporated make the game fun to play?

Barr and Stephenson [18] report the results of a work group aimed at producing an operational definition for Computational Thinking applied to North American K-12 education (i.e., at basic educational levels). The results of the work group include a comprehensive list of core CT concepts (including the three skills also mentioned by [16]) and a discussion of how these concepts are related to Computer Science itself and to basic and middle school subjects. The included concepts are: data collection, analysis and representation, problem decomposition, abstraction, algorithms and procedures, automation, parallelization and simulation.

The ACM Model Curriculum for K-12 Computer Science [19] emphasizes the need to develop computer skills in basic education in order to assist the development of problem-solving skills, provide support and deal with other sciences, as well as acting as a motivation factor for students. This reference curriculum suggests the teaching of Computing concepts for K-12 students. Such approach includes concepts, skills and competencies also discussed by Bezerra *et al.* [20]. The latest revision of the CSTA (Computer Science Teachers Association) K-12 Computer Science Standards [21] confirms and details the recommendations made by the previous two versions of the ACM Model Curriculum.

Isbell *et al.* [22] consolidated the results of a debate about the nature of Computing in order to guide the definition of curricula for undergraduate courses. An alternative view on Computational Thinking arises from this debate: for the authors, the concept of Computing is centered on the production of representative *models* of a particular domain. These models should be expressed in a particular language to be manipulated, modified and interpreted by a machine. In this case, the key concepts that are said to underlie CT are: models, abstraction, interpretation (of data or patterns); scale and limits for processing, model-based system simulation and possible automation of such models. The authors even choose to use the alternative expression *computationalist thinking* to emphasize that the meaning of their definition is related to the way of thinking of "one that does computing". One may notice that the idea of building models based on a formal language is significantly closer to Mathematics; this topic will be further discussed later in this article.

It it possible to identify some convergence in the definitions of CT presented by Lee *et al.* [16], Barr and Stephenson [18] and Isbell *et al.* [22]. The three works indicate that applying CT may be closely related to the abstraction of a process to allow its automation. This abstraction may be conceived by data collection and analysis, then represented as a model to allow its implementation on a computer in an algorithmic form. On the other hand, the work by Basawapatna *et al.* [23] alternatively defines Computational Thinking based on the identification of patterns in digital games built with software agents. These games are programmed by students during learning activities using a visual programming language based on building blocks. Remarkably, this is a far more specific proposal. The authors do not intend to necessarily define CT as a paradigm for reasoning, but instead identify a potential for knowledge transfer among students, as some patterns that appear in computer games can also be identified in physical and biological phenomena. Among the patterns identified, there are: generation, collision, transport, diffusion and hill climbing. The issue of knowledge transfer is essential to discuss the relationship between Computational Thinking and Mathematics, and this is the focus of the following section.

III. COMPUTATIONAL THINKING AND MATHEMATICS EDUCATION IN THE VIEW OF CURRICULUM GUIDELINES

The influence of technology on society and the consequent need to contextualize it in basic levels of the educational system are presented in various Latin American national curriculum guidelines. For instance, Chile's educational reform, started in 1997, stated that a great deal of manual work should be replaced by technological resources; furthermore, [24] clearly defines that Computing should not be understood only as an instrument for solving technical problems, but also as a model of reasoning. In this model, Computing has its own and true identity, regarding both the questions that it tries to answer and the methods applied to solve problems. Still according to these authors, the relationship between Mathematics and Computing is straightforward.

According to the Brazilian National Curriculum for Mathematics, the impact of technology to the life of individuals will require skills that do not involve simply dealing with machines [25]. Coincidentally, this is the same point of view used to differentiate the skills related to Computer Thinking from those associated to *Computer Literacy* (defined as the ability to operate computers properly to accomplish necessary tasks). The same line of reasoning goes on, affirming that the impact of technology, whose most important instruments nowadays are computers, will require that teaching of Mathematics shifts towards a curricular perspective that prioritizes the development of skills and procedures so that students could recognize and orient themselves in a world of ever-moving knowledge.

Based on the literature review presented in the previous section, it is possible to envision that some inherent characteristics of CT could be useful to help students to contextualize some contents of Mathematics, whether allied to computational resources or not. The reciprocal relationship may also be true: as mentioned earlier, Mathematics and some aspects of mathematical thinking act as a support for Computing activities.

To illustrate this point, three skills for Mathematics were identified. Two criteria were used for their selection. The first one was that these skills are present, whether as a general objective or as part of a specific content, in the Brazilian *Supplementary Curriculum Guidelines for Mathematics* [26] and in the Chilean *Fundamental Objectives and Minimum Obligatory Contents* [27]; for sake of comparison, both documents are focused on educational standards for the final years of basic education. The second one was the proximity of the skills to didactic activities related to CT already described in the literature. For each of the three skills, the following discussion presents possible paths for the joint development of Computational Thinking skills.

A. Mathematical representations and their semiotic relationship to algorithms

The reading and interpretation of symbols, codes and names inherent to mathematical language are part of the Math competencies expected from students. In particular, it is expected that students are able to translate a situation expressed in one "language", i.e., in one symbolic representation, into another. For instance, students should develop skills to transform situations described in discursive language to charts, tables, formulas and other representations, and vice versa [25]. Within the domain of the different representations allowed by mathematical language, is also expected that students are able to "translate" one representation into another, for example, convert data from a table to a chart and deduce its algebraic representation [27].

However, one of the main issues in Mathematics education dwells exactly over this key point. Worthington and Carruters [28] affirm that young students first meet Math as an unfamiliar "foreign" language, rather than a transparent symbolic system, thus causing considerable difficulties for their understanding. In this point, the learning of algorithmic representation can present some impacts over the learning of Mathematic semiotics – for instance, in a broader sense, one may consider that the elementary school procedures for adding, subtracting, multiplying and dividing are nothing but algorithms that give an operational meaning to the underlying concepts of the operations.

The ability of describing solutions for a problem using an algorithmic language is one of the core Computational Thinking skills defined in [21]. Such representation has some similarities to the algebraic language, particularly regarding to the representation of variables. Nonetheless, algorithms are inherently dynamic models, in opposition to the algebraic representation, which is typically used to express the static relationship between unknown magnitudes [22]. According to Mor and Noss [29], an algorithm under the imperative-structured paradigm is defined by a set of procedures to be followed step by step. This organization is quite similar to the structure of discursive language. Thus, to represent a problem in algorithmic form can be considered as an intermediate step between the verbal narrative and the algebraic language. By

Look at the problem below:

Y = X + 4

If X = 7, what is Y?

Fill in the blanks to draw a 10 sided shape:

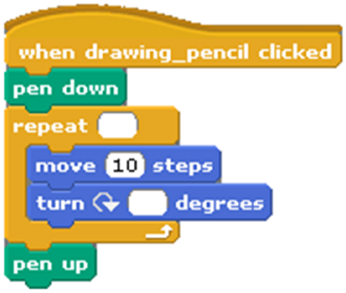

Figure 3. An example of correspondence between mathematical symbols and algorithmic representation (adapted from [30])

using this intermediate step, students can potentially experience a smoother transition from the imprecision and ambiguity of the spoken language to the formality of mathematical language.

An initial evidence of the enrichment of both the algorithmic language and Mathematics for students is presented by Lewis and Shah [30]. The authors identify a positive correlation between the scores of fourth graders on standardized tests about general mathematical knowledge and the scores of the same students on tests involving basic concepts about the construction of algorithms, implemented using the Scratch platform [31] during an after-hours workshop. The authors point out, albeit not exhaustively, some possible overlap between mathematical content and that content developed in the workshop classes. Figure 3, adapted from [30], shows an example of a possible relationship between mathematical symbols and algorithmic representation. The first question is taken from a standardized Math test and is related to the linear relationships between quantities expressed. The student is expected to understand that the value of y can be obtained through the information given. In the second case, there is clearly more content involved – the problem is about drawing a regular polygon, which requires the additional knowledge about the sum of a polygon's interior angles being 360 degrees, so at each iteration the student must repeat commands that turns the drawing direction in $360/n$ degrees, depending on the number n of sides of the polygon to be drawn, which is also the number of loop steps to be performed. Thus, the same concept of linear relationship between quantities is applied, but the narrative of the algorithm progressively allows students to identify and test their own hypotheses (as prescribed in [27]), actively constructing their own mathematical formalism.

This relationship has already been focused by Clements and Battista [32], who argue that the construction of algorithms for drawing polygons using the Logo language helped students to progress from their intuitive notions to more elaborate mathematical ideas about angles and rotation, allowing them to move to higher levels of geometric reasoning. In-Ok and Hee-Chan [33] held a similar study, driven toward gifted students,

Figure 4. Progressive levels of appropriation of Computational Thinking skills by students and a possible association to the types of Math reasoning defined by Brousseau (adapted from [16])

which concluded that even when students already use mathematical abstraction at a higher level, the process of constructing algorithms can help them to develop superior mathematical skills, especially those ones related to problem solving.

The shown evidences indicate that algorithmic solutions can help students to conjecture, test, and then verify results in a dynamic, but still mathematical way. Eventually, students can return to using the "static" mathematical notation and express the results found with a greater deal of confidence. As far as the specific domain of geometry is concerned, Lee [34] states that the process of formulating hypothesis, testing and verifying results is fundamental to develop the sense of geometrical reasoning which can be divided in three different and progressive types (pragmatic, semantic, intellectual) and even motivate students to create informal proofs. These three types of reasoning, due to Brousseau [35], are applicable to other mathematical contents. We compare this classification to a very similar three-stage approach, proposed by Lee *et al.* [16] to engage students in Computational Thinking activities. This comparison is graphically depicted in Figure 4. The first stage, "use", occurs when the student consume a computational artifact constructed by someone else. This is similar to the pragmatic level of reasoning. The second stage, called "modify", starts when students are confident enough to make modifications to the artifact. This comprehension can be related to the semantic level. Finally, the third stage, "create", happens when students are able to develop entirely new projects based on their ideas, which is equivalent to the intellectual level. Even in a rough level, this equivalence may provide an intellectual framework for the adequate development of didactic activities involving the manipulation and understanding of math and computational languages and the other skills discussed in this paper.

B. *Establishing relationships and identifying pattern regularities*

Brazilian Curricular Directives [26] state that a competence expected from students is the ability to "identify regularities in similar situations to establish rules, algorithms and properties". One recommended content in the Chilean Obligatory Contents [27] is the "solving of numerical challenges (…) including calculations oriented to the resolution of numerical regularities". From these recommendations, it is straightforward to infer that students should be encouraged to take exploratory attitudes toward the Mathematical facts in the world around them, actively establish some natural relationships. This assertion about Math reasoning has many similarities with Computational Thinking: pattern matching is one of the first skills that Wing [11] associates with CT. As we mentioned earlier, the CT definition by Basawapatna et al. [23] is completely based on pattern matching skills.

Whilst simple visual pattern matching seems to be one of the first fundamental cognitive skills children develop in early ages, the identification of regularities in different patterns and the subsequent abstraction of such regularities demands higher-order cognitive processes and structures. This kind of abstraction allows, for instance, mental operations which come from simple extrapolation of results deductable from a regular sequence to more complex inferences. Researches on Ethnomathematics have been bringing to light some interesting results, normally arisen from computer-unaware experiences and often involving pattern matching from nature. Some examples are Turner Contemporary [36], which brings a extensive range of activities for teachers and students to develop Mathematical skills from the analysis of patterns commonly found in nature; and Fisher [37], which presents a comprehensive study held with New Zealander Kuruti natives to understand their cognitive mechanisms for pattern matching related to counting, number and coordinates representation and basic operations, illustrated by Figure 5.

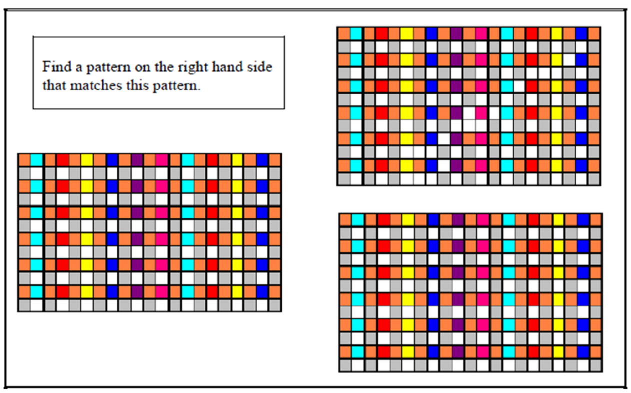

Figure 5. Pattern matching exercise using traditional bamboo-walls patterns woven from bamboo stalks (extracted from [37])

The formation of numerical and logical sequences is a mathematical content often explored in student experiments involving the identification of patterns employing computing resources. Mor and Noss [29] report three episodes of educational activities involving the identification of the formation rules of sequences by students through the development of algorithms using an educational software. The students' narrative indicate the cognitive processes involved in the identification of sequence regularity. The authors conclude that students may situate the abstraction of sequence rules more easily by utilizing the algorithmic language to generate the sequence before the deduction of its mathematical representation.

However, there are reports that this transition might not be trivial in all cases: Setti [1] uses the Fibonacci sequence in an activity driven to CS freshmen. The students are required to identify the sequence rule and then build an algorithm for calculating terms of the sequence. The author concludes that the transition from the mathematical representation of the sequence to its algorithmic representation is not straightforward for most of the students. Besides, it is shown that the root causes for this difficulty rely mainly on the differences between both semiotic representations. It can be conjectured whether such students would present the same difficulties if they had been first exposed to Computational Thinking foundations in early years, instead of the so-called "traditional" approach.

C. *Descriptive and representative models*

The third skill is related to the definition and interpretation of mathematical models to analyze and explain situations. Teaching of Mathematics should involve situations that are familiar to the student, such as: profit and loss calculation based on charts; the application of statistics and probability to the estimation of voting intentions in an electoral campaign, among others [26]. In this context, the building of models by the students themselves is of particular interest; according to [27], students should be progressively stimulated to construct models representing everyday situations since the second year of middle school. This didactic strategy is called *Mathematical Modeling*.

When modeling is used on a didactic perspective, the definition of what a valid model should differ from the definition used when modeling is used as a professional activity. For Bassanezi [38], a model is useful in an educational setting if it expresses ideas in a clear and unambiguous way and is able to generate results whose numerical solutions can be generated using computational resources. This conception of Mathematical Modeling in the classroom is especially useful because its results can be extended by the application of the Computational Thinking skills previously discussed in this article. The definition of CT proposed by the CSTA [21] incorporates phenomena modeling and simulations as one of its core skills.

The computational support for the Mathematical Modeling activities is usually restricted to the use of spreadsheets, chart plotting software and web browsing [39] – in other word, the skills required from the students (and eventually developed by them) are restricted to the computation literacy level. On the other hand, the usage of more powerful and more expressive computational tools may stimulate students to develop and apply Computational Thinking skills of a higher order. For example, Lee *et al.* [16] describe an activity where students produce a contagion model for a specific disease considering the layout of their school and its number of students. The model was implement used StarLogo TNG, a block-based computing environment based on software agents. Mathematical Modeling "augmented" by appropriate software tools even have a potential for producing actions of relevant social impact: the EcoAgents project (http://tltl.stanford.edu/projects/ecoagents) is an example of such initiatives. The project aims to enable students in basic educational levels to use sensors and software tools to produce models of climatic changes.

If we assume both CT activities we just described are in some way related to the classic concept of Mathematical Modeling presented in the literature, they would belong to the socio-critical perspective as described by Barbosa [40], as they deal with issues closely related to the students' reality and whose resolution by modeling may allow a wider and more critic comprehension of the world around them. CT activities, applied by using appropriate software tools, may adhere to Bassanezi's definition and expand the possibilities of modeling in the classroom. The theoretical ground for the augmented possibilities of knowledge discovery and production when using new technological devices is discussed in depth by Villarreal and Borba [41].

IV. SOME RESEARCH PATHS

Based on the research trends presented on this article, some challenges to the Latin American research community in Computing Teaching can be identified, as far as the incorporation of Computational Thinking to elementary education is concerned. Based on the presented literature review, it is possible to conclude that an equivalence association between Math and computing contents for elementary education is still incipient. More research is needed in order to obtain a comprehensive mapping between the Mathematics and Computing body of knowledge. With the previous discussion about three skills defined by the Brazilian and Chilean Curriculum Guidelines, we intend to contribute to the organization and debate about this mapping.

Once the mapping is established, it becomes necessary to develop and evaluate didactic activities that involve Math contents that can be explored and developed together with CT skills. Some authors [17], [18] indicate that the main pathway to stimulating Computational Thinking skills is engaging students in *doing* computing, not as a tool, but as a problem-solving way of reasoning. This can be accomplished by the creation and evaluation of such activities.

It is important to point out that CT activities should also be developed for and applied in the first semesters of higher education courses, as there is a whole generation of students who did not have the opportunity to explicitly develop this type of reasoning in basic education. Some interesting examples are provided by Settle [13], who describes an undergraduate course on game design offered for students with various backgrounds. Qin [14] describes a bioinformatics course based on CT concepts and designed to introduce the area to biology students.

Mathematical Modeling shall bring a bigger challenge in this context. Perhaps a "revised", computer-augmented Mathematical Modeling didactic strategy will be necessary in order to accommodate the possibilities brought by CT skills and adequate software tools. Such tools must be expressive enough to model phenomena (for instance, allowing the specification of software agents, as in [16], [42]) but still keeping the syntactic structure simple enough so that beginners can rapidly produce initial results, and progressively generate more complex results as they explore the tool.

Last, but not least, the incorporation of Computational Thinking concepts to the initial and on-duty teacher training should be an important step for its dissemination. Basic education is an environment often subject to many constraints and regulations, where appropriate teacher support is fundamental to any desired change. This should be a long term initiative, and preliminary attempts to introduce Computational Thinking concepts in education courses appear to be promising [15], [43].

V. FINAL CONSIDERATIONS

In a world where computational devices are more and more pervasive and vital to most activities, the interest for graduation courses in Computing is diminishing. This paradox may be explained by poor educational results in Mathematics in the initial series. On the other hand, Computational Thinking organizes a set of skills related to the adequate comprehension and use of Computing concepts that are becoming fundamental to everyone.

A way to incorporate CT to the basic education is to analyze its relationship with other knowledge areas already present in the basic education level, such as Mathematics. In this article, we make an attempt to organize activities that promote CT (and may promote Math concepts at the same time) associating them with three skills, extracted from Brazilian and Chilean curriculum guidelines: Math and algorithmical representations, pattern identification and Mathematical Modeling. Obviously, the separation of skills is useful only for the sake of discussion – didactic activities may span across more than only one of the three areas. Based on a literature review, these appear to be the most promising skills to be stimulated in the joint development of CT and Math.

By analyzing research works related to the three skills, we identify and propose to the community research paths with a greater potential to clearly demonstrate the benefits of Computational Thinking teaching. This way, Computing fundamentals will be accessible in an adequate way to a wider audience.

REFERENCES

[1] M. de O. G. Setti, "O Processo de Discretização do Raciocínio Matemático na Tradução para o Raciocínio Computacional: Um Estudo de Caso no Ensino/Aprendizagem de Algoritmos," PhD Thesis, UFPR, Curitiba, 2009.

[2] T. Beaubouef, "Why computer science students need math," *SIGCSE Bulletin*, vol. 34, no. 4, pp. 57–59, Dec. 2002.

[3] B. C. Wilson and S. Shrock, "Contributing to success in an introductory computer science course: a study of twelve factors," in *Proceedings of the thirty-second SIGCSE technical symposium on Computer Science Education*, New York, 2001, pp. 184–188.

[4] P. F. Campbell and G. P. McCabe, "Predicting the success of freshmen in a computer science major," *Communications of the ACM*, vol. 27, no. 11, pp. 1108–1113, Nov. 1984.

[5] R. Targino, "Quase 6 em 10 alunos do ensino médio de SP saíram da escola com desempenho ruim em matemática em 2011," *UOL Educação*, 07-Mar-2012.

[6] O. Larrañaga and A. Telias, "Inequality of Opportunities in the Educational Attainment of Chilean Students," in *Série Documentos de Trabajo*, Santiago, Chile: Universidad de Chile, 2009.

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

[8] G. Paeton, "University maths 'too difficult' for British students," *The Telegraph*, 10-Feb-2012.

[9] C. S. Hood and D. J. Hood, "Toward integrating computing concepts into the K-12 curriculum," *SIGCSE Bulletin*, vol. 37, no. 3, p. 375–375, Jun.

2005.
[10] S. Ophir, “Computer sciences and commercial forces: can computer science be considered science?,” *SIGCAS Computers and Society*, vol. 36, no. 2, Jun. 2006.
[11] J. M. Wing, “Computational thinking,” *Communications of the ACM*, vol. 49, no. 3, pp. 33–35, Mar. 2006.
[12] P. J. Denning, “Is computer science science?,” *Communications of the ACM*, vol. 48, no. 4, pp. 27–31, Apr. 2005.
[13] A. Settle, “Computational thinking in a game design course,” in *Proceedings of the 2011 conference on Information technology education*, New York, NY, USA, 2011, pp. 61–66.
[14] H. Qin, “Teaching computational thinking through bioinformatics to biology students,” *SIGCSE Bulletin*, vol. 41, no. 1, pp. 188–191, Mar. 2009.
[15] S. I. Ahamed, D. Brylow, R. Ge, P. Madiraju, S. J. Merrill, C. A. Struble, and J. P. Early, “Computational thinking for the sciences: a three day workshop for high school science teachers,” in *Proceedings of the 41st ACM technical symposium on Computer science education*, New York, NY, USA, 2010, pp. 42–46.
[16] I. Lee, F. Martin, J. Denner, B. Coulter, W. Allan, J. Erickson, J. Malyn-Smith, and L. Werner, “Computational thinking for youth in practice,” *ACM Inroads*, vol. 2, no. 1, pp. 32–37, Feb. 2011.
[17] C. Hu, “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, 2011, pp. 223–227.
[18] V. Barr and C. Stephenson, “Bringing computational thinking to K-12: what is Involved and what is the role of the computer science education community?,” *ACM Inroads*, vol. 2, no. 1, pp. 48–54, Feb. 2011.
[19] A. Tucker, “A Model Curriculum for K–12 Computer Science - Final Report of the ACM K–12 Task Force Curriculum Committee.,” 2006. [Online]. Available: http://csta.acm.org. [Accessed: 16-Apr-2012].
[20] L. N. Bezerra and I. F. Silveira, “Licenciatura em Computação no Estado de São Paulo: uma Análise Contextualizada e um Estudo de Caso,” in *Anais do CSBC 2011*, Natal, 2011.
[21] The CSTA Standards Task Force, “CSTA K-12 Computer Science Standards,” ACM Computer Science Teachers Association, New York, 2011.
[22] C. L. Isbell, L. A. Stein, R. Cutler, J. Forbes, L. Fraser, J. Impagliazzo, V. Proulx, S. Russ, R. Thomas, and Y. Xu, “(Re)defining computing curricula by (re)defining computing,” *SIGCSE Bulletin*, vol. 41, no. 4, pp. 195–207, Jan. 2010.
[23] A. Basawapatna, K. H. Koh, A. Repenning, D. C. Webb, and K. S. Marshall, “Recognizing computational thinking patterns,” in *Proceedings of the 42nd ACM technical symposium on Computer science education*, New York, 2011, pp. 245–250.
[24] Enlaces, “Informática Educativa en el Currículum de Enseñanza Media – Matemática,” 2012. [Online]. Available: http://www.eduteka.org/pdfdir/ChileCurriculoMatematicasTics.pdf. [Accessed: 17-May-2012].
[25] Brasil. Ministério da Educação, *Parâmetros Curriculares Nacionais: Ensino Médio*. Brasília: MEC/SEB, 1999.
[26] Brasil. Ministério da Educação, *PCN+ Ensino Médio: Orientações Curriculares Complementares aos Parâmetros Curriculares Nacionais*. Brasília: MEC/SEB, 2002.
[27] Chile. Ministerio de Educación, “Objetivos fundamentales y contenidos mínimos obligatórios de la Educación Media,” 1998. [Online]. Available: http://www.aep.mineduc.cl/images/pdf/2010/CurriculumMedia.pdf. [Accessed: 27-May-2012].
[28] M. Worthington and E. Carruthers, “Becoming bi-numerate: a study of teachers’ practices concerning children’s early ‘written’ mathematics,” in *Proceedings of European Early Childhood Education Research Association (EECERA) Conference*, Glasgow, 2003.
[29] Y. Mor and R. Noss, “Programming as mathematical narrative,” *International Journal of Continuing Engineering Education and Life-long Learning*, vol. 18, no. 2, pp. 214–233, 2008.
[30] C. M. Lewis and N. Shah, “Building Upon and Enriching Grade Four Mathematics Standards with Programming Curriculum,” in *Proceedings of the 43th SIGCSE technical symposium on Computer science education*, New York, 2012.
[31] M. Resnick, J. Maloney, A. Monroy-Hernández, N. Rusk, E. Eastmond, K. Brennan, A. Millner, E. Rosenbaum, J. Silver, B. Silverman, and Y. Kafai, “Scratch: programming for all,” *Communications of the ACM*, vol. 52, no. 11, pp. 60–67, Nov. 2009.
[32] D. H. Clements and M. T. Battista, “The effects of Logo on Children’s Conceptualization of Angle and Polygons,” *Journal for Research in Mathematics Education*, vol. 21, no. 5, pp. 365–371, 1990.
[33] J. In-Ok and L. Hee-Chan, “Case Studies in Thinking Processes of Mathematically Gifted - Elementary Students through Logo Programming,” in *Proceedings of the Sixteenth Asian Technology Conference in Mathematics*, Bolu, Turkey, 2011.
[34] K. H. Lee, “Mathematically gifted students’ geometrical reasoning and informal proof,” in *Proceedings of the 29th Conference of the International Group for the Psychology of Mathematics Education*, Melbourne, Australia, pp. 241–256.
[35] G. Brousseau, *Theory of Didactical Situation in Mathematics Education*. Dordrecht: Kluwer Academic Publishers, 1997.
[36] Turner Comtemporary, “Math thorugh patterns,” 2009. [Online]. Available: http://www.turnercontemporary.org/media/documents/turner-contemporary-maths-through-pattern.pdf. [Accessed: 31-May-2012].
[37] J. Fisher, “Enriching Students’ Learning Through Ethnomathematics in Kuruti Elementary Schools in Papua New Guinea,” in *Proceedings of the 3rd Intl. Conference on Ethnomathematics*, Auckland, New Zealand, 2006.
[38] R. Bassanezi, *Ensino-aprendizagem com modelagem matemática*. São Paulo: Contexto, 2002.
[39] L. do N. Diniz, “O Papel das Tecnologias da Informação e Comunicação nos Projetos de Modelagem Matemática,” Master's Thesis, UNESP, Rio Claro, 2007.
[40] J. C. Barbosa, “Mathematical Modelling in classroom: a socio-critical and discursive perspective,” *Zentralblatt für Didaktik der Mathematik*, vol. 38, no. 3, pp. 293–301, 2006.
[41] M. E. Villarreal and M. C. Borba, “Collectives of humans-with-media in mathematics education: notebooks, blackboards, calculators, computers and …notebooks throughout 100 years of ICMI,” *ZDM Mathematics Education*, vol. 42, no. 1, pp. 49–62, 2010.
[42] C. S. de Souza, A. C. B. Garcia, C. Slaviero, H. Pinto, and A. Repenning, “Semiotic traces of computational thinking acquisition,” in *Proceedings of the Third international conference on End-user development*, Heidelberg, 2011, pp. 155–170.
[43] A. Yadav, N. Zhou, C. Mayfield, S. Hambrusch, and J. T. Korb, “Introducing computational thinking in education courses,” in *Proceedings of the 42nd ACM technical symposium on Computer science education*, New York, NY, USA, 2011, pp. 465–470.