Theoretical Computer Science in Basic Education: A Systematic Review*†

Braz A. S. Junior[1], **Simone A. C. Cavalheiro**[1], **Luciana Foss**[1]

[1]Progama de Pós Graduação em Computação – Universidade Federal de Pelotas (UFPel)
CEP 96.010-610 – Pelotas – RS – Brazil

{badsjunior,simone.costa,lfoss}@inf.ufpel.edu.br

Abstract. *This paper presents a systematic literature review on theoretical computer science in basic education. Computing education has been reignited with the new computational thinking wave, calling out for the importance of problem-solving skills. Yet, it has been focusing its efforts in programming with visual languages, games and robotics. Meanwhile, theoretical computer science has rarely been touched in education. Therefore this work investigates 17 journal and conference paper that present solutions including this topic on basic education in the last 20 years. To conclude, a discussion about what topics of theoretical computer science were used takes place and a need for further research is identified.*

Resumo. *Este artigo apresenta uma revisão sisemática da literatura sobre informática teórica na educação básica. A educação em computação foi reacendida com a nova onda do pensamento computacional, clamando pela importância de habilidades para a resolução de problemas. No entanto, seus esforços têm se concentrado em programação com linguagens visuais, jogos e robótica. Enquanto isso, informática teórica tem sido raramente abordada na educação. Por isso, este trabalho investiga 17 artigos de periódico e de conferências que apresentam soluções incluindo este tópico na educação básica nos últimos 20 anos. Para concluir, é realizada uma discussão sobre os tópicos da informática teórica utilizados e indentificada uma demanda por mais pesquisas na área.*

1. Introduction

Computing Education (CEd) has gotten an increasing attention from the scientific community in the last years, making good use of promising and appealing allies, such as block-programming languages [Weintrop 2019], gamification [Gari et al. 2018], robotics [Yesharim and Ben-Ari 2018] and computational thinking [Angeli and Giannakos 2020]. Although, the roots of computation seems to have been left behind, **Systematic Literature Reviews (SLR)** [Garneli et al. 2015, Crick 2017] covering CEd as a whole have spotted that few content related to **Theoretical Computer Science (TCS)** has been approached.

*Concluded postgraduate paper.

†This work was supported by Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Finance Code 001, MCTIC/CNPq (Rede Sacci), SMED/Pelotas, PREC and PRPPG/UFPel.

A broad overview [Crick 2017] shows CEd has been done specially through the trend of using programming and the new central theme to the field: Computational Thinking. What "refers to a collection of computational ideas and habits of mind that people in computing disciplines acquire through their work in designing programs, software, simulations, and computations performed by machinery." [Crick 2017]. It is mentioned there that some pedagogic approaches are centred in topics of TCS: such as understanding automata [Isayama et al. 2016]; finite state and Turing machines [Korte et al. 2007]; and abstract/"notional" machines [Sorva 2007]. While others stand at least raise some discussion around them: as in algorithm design[Levitin 2016]; and concurrency/synchronization [Kolikant 2004].

Narrowing the scope, a SLR [Garneli et al. 2015] explores the educational contexts used for CEd in k-12. It makes clear how popular programming tools are in this field, specially visual programming languages. Studies making use of game design, tangible kits and robotics are also shown to be popular. The approaches closest to TCS in this SLR are about computer modeling and computation [Benacka and Reichel 2013, Sengupta and Farris 2012, Sengupta et al. 2013]; being 3 out of the 47 they analyzed.

These studies investigate CEd as a whole and focused on k-12, but they have no compromise with locating TCS within CEd. They actually expose how rare is TCS in CEd, far overshadowed by visual programming languages approaches. From this perspective, they leave us with the following Research Questions (RQ):

- **RQ1** - What are the studies explicitly treating TCS in k-12? Where are they from? Where are they published?
- **RQ2** - What are the topics or concepts of TCS being approached in k-12? What are the teaching strategies used?

Therefore, the purpose of this work is to find how is TCS going in k-12 education and discuss what could be its role there. That is why we present a SLR on TCS-centered approaches in k-12. The rest of this paper is organized as follows. Section 2 describes the method used in the research process. Section 3 presents the results and a discussion around them. Section 4 concludes the paper calling for further research.

2. Methods

We used the following four search engines due to their popularity and repository size within the fields of CS and/or Education: ACM Digital Library[1]; Springer Link[2]; IEEE Xplore[3]; and ERIC[4]. The search string was defined considering the synonyms of TCS and basic education as shown in Table 1. Producing the following search string:

```
("theoretical computer science" OR "theoretical
computing" OR "theoretical informatics") AND
((school OR primary OR secondary OR k-12 OR
basic OR kid*) AND education)
```

[1] https://dl.acm.org/
[2] https://link.springer.com/
[3] https://ieeexplore.ieee.org/Xplore/home.jsp
[4] https://eric.ed.gov/?

Table 1. Word table of synonyms for the search string

		basic	
	Computer Science	k-12	
Theoretical	Computing	primary	education
	Informatics	secondary	
		school	
		kids	

Using the search string in the aforementioned search engines we screened the title of the 1004 results (606 in ACM DL; 375 in Springer Link; 17 in IEEE Xplored; and 6 in ERIC) for studies meeting the inclusion criteria for this study were: written in English; being explicitly related to CEd; published in scientific journals or conferences; published within the last 20 years. Leaving 101 reports (74 in ACM DL; 16 in Springer Link; 8 in IEEE Xplored; and 3 in ERIC) to have their abstracts screened while consulting the following exclusion criteria: (1) does not explicit any content related to TCS; (2) does not target/include k-12 education; (3) unable to access; and (4) being a duplicate. At the final of the screenings, a total of 17 papers were considered, as summarized in Figure 1.

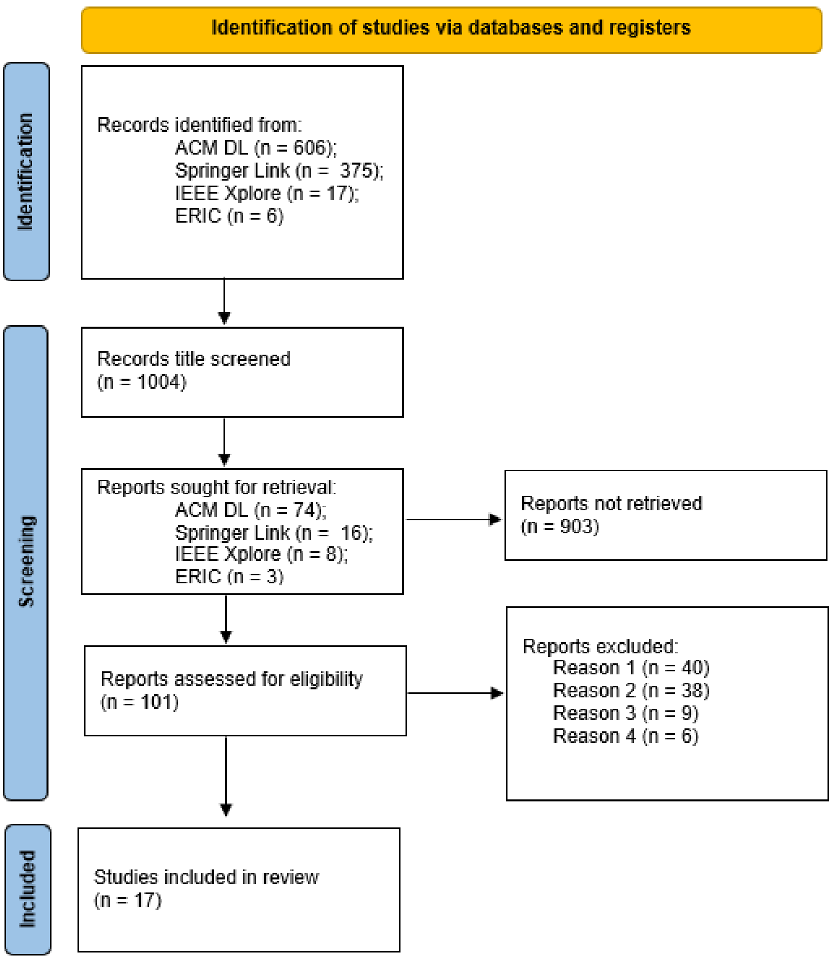

Figure 1. Work flowchart for this SLR.

3. Results and Discussion

Our SLR found 17 papers treating TCS in basic education, listed in the table 5, featuring the following. Traditional introductory courses on formal grammars/their corresponding automatons [Knobelsdorf et al. 2014] and graph theory [Carruthers et al. 2010], as well as virtual ones about complexity/computability [del Vado Vírseda 2019b, del Vado Vírseda 2020]. A moodle-supported online course on finite automata, push down automata and turing machines [Chuda 2007]. An automatic problem-generator tool to create exercises for students learning automatons [D'Antoni et al. 2020]. A game design activity using formal languages and regular expressions to build finite state and turing machines modeling games [Korte et al. 2007]. A puzzle game featuring fundamental concepts of automata theory [Isayama et al. 2016]. A card game using regular expressions to control finite state machines [Valente 2004]. A multi-agent system of ladybug robots being modeled by formal grammars [Kelemen and Kubík 2002]. A software (KARA) that allows students to control a ladybug using a FSM [Kiesmüller 2009]. A microworld environment of a zombie outbreak putting students to model the infection spread and containment measures [Maxville and Sandford 2020]. Flipped classroom experiments approaching mostly automata theory [Wolf et al. 2018, Morisse 2015]. And two papers discussing and suggesting curriculum, instead of actual experiments [Schlueter and Brinda 2008, Pantsar 2021].

In Tables 2-4, the column "N" reports the amount of papers found in the SLR that fits the row, this is represented between parenthesis in this paragraph. The studies were published in a variety of conferences and two journals, TOCE and Erkenntnis, ranging their themes from computer aided verification and robot motion to computing education and learning technologies, as listed in table 2. The authors, as seen in the Table 5, come from Germany (3), Spain (3), Slovakia (2), USA (2), Denmark (1), Finland (1), Australia (1), Scotland (1), Canada (1) and Japan (1). The disposition of TCS topics is shown in Table 3 as follows. The most approached topic was general Automata Theory (6), followed by specifically Finite State Machines/Automata (5) and Turing Machines (4). Which evens with Complexity Theory (4) and Computability (4). Then other Models of Computation and Formal Languages draw (3), followed by Regular Expressions (2). At last, Graph Theory was approached by a single work (1). Meanwhile, the teaching strategies used are shown in Table 4 as follows. The top teaching strategy is the use of Virtual/Digital Tools (6), followed by traditional lectures/tests (3) and unplugged activities, such as card and puzzle games (3). Then, there are Flipped Clasroom (2), Microworlds (1) and Robotics (1) initiatives.

Most papers give a special attention to the problem solving aspect of their proposed activities. Noteworthy, even an automated analysis of problem solving strategies is presented using the data gathered by an educational software (KARA) [Kiesmüller 2009]. Another common factor in the papers is the discussion of lack of related work (experiments on TCS in basic education), saying they are outdated (1980s and 1990s) [del Vado Vírseda 2019a] or having to relate to other CEd works like LOGO programming [Valente 2004]. It is clear that TCS is taught in basic education primarily through models of computation, generically or specifically one kind (FSM/TM). Additionally, formal Languages are used mostly to introduce automata. In second place comes computability and complexity theory, always approached together.

Table 2. Conferences and Journals where the selected studies were published.

Acronym	Full Name	N
E-Learn	World Conference on E-Learning in Corporate, Government, Healthcare, and Higher Education	1
ICALT	International Conference on Advanced Learning Technologies	1
RoMoCo	International Workshop on Robot Motion and Control	1
ICCAV	International Conference on Computer Aided Verification	2
CompSysTech	International Conference on Computer Systems and Technologies	2
SIGCSE	ACM Technical Symposium on Computer Science Education	2
Koli Calling	Koli Calling International Conference on Computing Education Research	1
ITiCSE	SIGCSE Conference on Innovation and technology in computer science education	1
WiPSCE	Workshop in Primary and Secondary Computing Education	1
CompEd	ACM Conference on Global Computing Education	1
WCCCE	Western Canadian Conference on Computing Education	1
TOCE	ACM Transactions on Computing Education	2
Erkenntnis	Erkenntnis	1

Table 3. Topics of Theoretical Computer Science approached in the selected studies.

Acronym	Full Name	N
AT	Automata Theory	6
FSM	Finite State Machines	5
TM	Turing Machines	4
CT	Complexity Theory	4
Comp	Computability	4
FL	Formal Languages	3
MoC	Models of Computation	3
RegEx	Regular Expressions	2
GT	Graph Theory	1

Table 4. Teaching Strategies used in the selected studies.

Acronym	Full Name	N
Virt	Virutal/Digital Tool	6
Trad	Traditional Lectures and Tests	3
unP	Unplugged Card/Puzzle Game, Paper and Pencil	3
Flip	Flipped Classroom	2
mW	Microworlds	1
RT	Robotics/Tinkering	1

Table 5. Studies on Theoretical Computer Science in Basic Education

Reference	Publisher	Country	Topics	Strategy
[Kelemen and Kubík 2002]	RoMoCo	Slovakia	FL, MoC	RT
[Valente 2004]	ICALT	Denmark	MoC, FSM, RegEx	unP
[Chuda 2007]	CompSysTech	Slovakia	AT, FSM, TM	Virt
[Korte et al. 2007]	ITiCSE	Scotland	FL, Regex, FSM, TM	Virt
[Schlueter and Brinda 2008]	CompSysTech	Germany	AT	–
[Kiesmüller 2009]	TOCE	Germany	FSM	Virt
[Carruthers et al. 2010]	WCCCE	Canada	GT	Trad
[Knobelsdorf et al. 2014]	SIGCSE	USA	FL, AT, FSM, TM	Trad
[Morisse 2015]	ELC	Slovakia	FL, AT, Comp	Flip
[Isayama et al. 2016]	TOCE	Japan	AT	unP
[Wolf et al. 2018]	WiPSCE	Germany	AT	Flip
[del Vado Vírseda 2019a]	CompEd	Spain	Comp, CT	Trad
[del Vado Vírseda 2019b]	SIGCSE	Spain	Comp, MoC, CT	Trad
[del Vado Vírseda 2020]	Koli Calling	Spain	Comp, CT	Virt
[Maxville and Sandford 2020]	ICCAV	Australia	Mod	mW, Virt, unP
[D'Antoni et al. 2020]	ICCAV	USA	AT, FL	Virt
[Pantsar 2021]	Erkenntnis	Finland	Comp, CT, TM	–

4. Conclusion

This paper offers a SLR that was able to capture only 17 studies involving TCS in basic education using large and popular scientific search engines of the CS and Education fields. This shows a gap in the basic CEd, that stands large volumes of studies reporting the use of programming, gaming and robotics to teach CS. Given the current trend towards computational thinking and problem-solving, TCS should be considered more often in CEd due to its analytic and problem-centered nature.

References

Angeli, C. and Giannakos, M. (2020). Computational thinking education: Issues and challenges. *Computers in Human Behavior*, 105:106185.

Benacka, J. and Reichel, J. (2013). Computer modeling with delphi. In *International Conference on Informatics in Schools: Situation, Evolution, and Perspectives*, pages 138–146. Springer.

Carruthers, S., Milford, T., Pelton, T., and Stege, U. (2010). Moving k-7 education into the information age. In *Proceedings of the 15th Western Canadian Conference on Computing Education*, pages 1–5.

Chuda, D. (2007). Visualization in education of theoretical computer science. In *Proceedings of the 2007 international conference on Computer systems and technologies*, pages 1–6.

Crick, T. (2017). Computing education: An overview of research in the field. *London: Royal Society*.

del Vado Vírseda, R. (2019a). Computability and algorithmic complexity questions in secondary education. In *Proceedings of the ACM Conference on Global Computing Education*, pages 51–57.

del Vado Vírseda, R. (2019b). Introducing theoretical computer concepts in secondary education. In *Proceedings of the 50th ACM Technical Symposium on Computer Science Education*, pages 1272–1272.

del Vado Vírseda, R. (2020). From the mathematical impossibility results of the high school curriculum to theoretical computer science. In *Koli Calling'20: Proceedings of the 20th Koli Calling International Conference on Computing Education Research*, pages 1–5.

D'Antoni, L., Helfrich, M., Kretinsky, J., Ramneantu, E., and Weininger, M. (2020). Automata tutor v3. In *International Conference on Computer Aided Verification*, pages 3–14. Springer.

Gari, M. R. N., Walia, G. S., and Radermacher, A. D. (2018). Gamification in computer science education: A systematic literature review. In *2018 ASEE Annual Conference & Exposition*.

Garneli, V., Giannakos, M. N., and Chorianopoulos, K. (2015). Computing education in k-12 schools: A review of the literature. In *2015 IEEE Global Engineering Education Conference (EDUCON)*, pages 543–551.

Isayama, D., Ishiyama, M., Relator, R., and Yamazaki, K. (2016). Computer science education for primary and lower secondary school students: Teaching the concept of automata. *ACM Transactions on Computing Education (TOCE)*, 17(1):1–28.

Kelemen, J. and Kubík, A. (2002). Robots and agents in basic computer science curricula. In *Proceedings of the Third International Workshop on Robot Motion and Control, 2002. RoMoCo'02.*, pages 309–317. IEEE.

Kiesmüller, U. (2009). Diagnosing learners' problem-solving strategies using learning environments with algorithmic problems in secondary education. *ACM Transactions on Computing Education (TOCE)*, 9(3):1–26.

Knobelsdorf, M., Kreitz, C., and B"ohne, S. (2014). Teaching theoretical computer science using a cognitive apprenticeship approach. In *Proceedings of the 45th ACM technical symposium on Computer science education*, pages 67–72.

Kolikant, Y. B.-D. (2004). Learning concurrency: evolution of students' understanding of synchronization. *International Journal of Human-Computer Studies*, 60(2):243–268.

Korte, L., Anderson, S., Pain, H., and Good, J. (2007). Learning by game-building: a novel approach to theoretical computer science education. In *Proceedings of the 12th annual SIGCSE conference on Innovation and technology in computer science education*, pages 53–57.

Levitin, A. (2016). Algorithm design strategies in cs curricula 2013: hits and misses. *Journal of Computing Sciences in Colleges*, 31(3):78–84.

Maxville, V. and Sandford, B. (2020). Computational science vs. zombies. In *International Conference on Computational Science*, pages 622–633. Springer.

Morisse, K. (2015). Implementation of the inverted classroom model for theoretical computer science. In *E-Learn: World Conference on E-Learning in Corporate, Government, Healthcare, and Higher Education*, pages 426–435. Association for the Advancement of Computing in Education (AACE).

Pantsar, M. (2021). Cognitive and computational complexity: Considerations from mathematical problem solving. *Erkenntnis*, 86(4):961–997.

Schlueter, K. and Brinda, T. (2008). Characteristics and dimensions of a competence model of theoretical computer science in secondary education. *ACM SIGCSE Bulletin*, 40(3):367–367.

Sengupta, P. and Farris, A. V. (2012). Learning kinematics in elementary grades using agent-based computational modeling: a visual programming-based approach. In *Proceedings of the 11th international conference on interaction design and children*, pages 78–87.

Sengupta, P., Kinnebrew, J. S., Basu, S., Biswas, G., and Clark, D. (2013). Integrating computational thinking with k-12 science education using agent-based computation: A theoretical framework. *Education and Information Technologies*, 18(2):351–380.

Sorva, J. (2007). Notional machines and introductory programming education. *Trans. Comput. Educ*, 13(2):1–31.

Valente, A. (2004). Exploring theoretical computer science using paper toys (for kids). In *IEEE International Conference on Advanced Learning Technologies, 2004. Proceedings.*, pages 301–305. IEEE.

Weintrop, D. (2019). Block-based programming in computer science education. *Communications of the ACM*, 62(8):22–25.

Wolf, A., Wilhelm-Weidner, A., and Nestmann, U. (2018). A case study of flipped classroom for automata theory in secondary education. In *Proceedings of the 13th Workshop in Primary and Secondary Computing Education*, pages 1–6.

Yesharim, M. F. and Ben-Ari, M. (2018). Teaching computer science concepts through robotics to elementary school children. *International Journal of Computer Science Education in Schools*, 2(3).