Introducing Finite Automata in Basic Education: A Work-in-Progress Proposal

Júlia Veiga[1], **Simone André da Costa Cavalheiro**[1], **Luciana Foss**[1]

[1]Center for Technological Development – Federal University of Pelotas (UFPel)
Pelotas – RS – Brazil

{jvsilva,simone.costa,lfoss}@inf.ufpel.br

Abstract. *This paper presents a proposal for an educational activity to introduce finite automata in Basic Education through a narrative-based approach grounded in Project-Based Learning. Aiming to bridge the gap between Theoretical Computer Science and early school levels, the proposal consists of seven sequential missions where students take on the role of cyber defense agents. Each mission introduces core finite automata concepts – such as states, transitions, acceptance or rejection of a word, and language recognition – using progressive challenges that encourage problem solving and student autonomy.*

1. Introduction

Computer Science Education (CSEd), once primarily focused on Higher Education, has increasingly become part of schooling at earlier levels. In Basic Education[1], most efforts focus on programming using visual languages, such as Scratch [Resnick et al. 2009], robotics, or even unplugged activities [Kite et al. 2021, Ching and Hsu 2024, Battal et al. 2021]. However, topics related to Theoretical Computer Science remain unexplored – even when covered in official documents, as is the case with automata in Brazil's National Common Curricular Base (BNCC) [Brazil 2022]. Introducing subjects such as automata to students in Basic Education is challenging given its abstract nature and the high failure rates even at the undergraduate level. Despite its importance in the Computing area, the course of formal languages and automata is often perceived as dry, abstract, and poorly connected to professional practice [Ramos 2009]. This highlights the need for more engaging and contextualized approaches.

One possible way to support Basic Education students in learning this topic is through active learning methodologies [Berbel 2011]. While traditional methods prioritize the transmission of information and are teacher-centered, active approaches promote self-directed learning and the collaborative construction of knowledge, with the teacher acting as a facilitator. Students who perceive themselves as autonomous in their school interactions tend to show positive outcomes in motivation, engagement, learning, academic performance, and psychological well-being [Reeve 2009]. In this context, this paper presents a work-in-progress activity based on finite automata. The proposal seeks to bring fundamental computing concepts closer to the school context by incorporating elements of Problem-Based Learning (PBL) [Barrows 1996], an active methodology. In the activity, students take on the role of scientists engaged in a cyber defense mission.

[1]In this paper, "Basic Education" refers to primary and secondary education, i.e., levels 1, 2, and 3 of the International Standard Classification of Education [UNESCO Institute for Statistics 2012].

The activity is structured into seven missions, each one of them aligned with different concepts of automata, promoting a gradual progression in student understanding.

The remainder of this paper is organized as follows: Section 2 provides the theoretical background on finite automata; Section 3 reviews related work on introducing automata concepts in Basic Education; Section 4 presents the proposal; and Section 5 offers final remarks and directions for future research.

2. Finite Automata

Automata Theory investigates abstract computational models, commonly referred to as "machines" [Hopcroft et al. 2001, Mogensen 2024]. Such models are applied in areas including lexical analysis, text corpus analysis, and system verification. Formally, a Deterministic Finite Automaton (DFA) is defined by a tuple M = (Q, Σ, δ, $q0$, F), where:

- Q is a non-empty finite set of states;
- Σ is a finite non-empty set of symbols (the alphabet);
- δ : Q $\times$ $\Sigma \rightarrow Q$ is the transition function, which assigns a new state based on a given state and input symbol, although it may not be defined for every possible pair;
- $q0 \in Q$ is the initial state;
- $F \subseteq Q$ is the set of accepting (or final) states.

There is also an extension of the transition function: $\hat{\delta}(q, w)$, where $q \in Q$ and $w \in \Sigma^*$, which returns $q' \in Q$ as the state reached by the automaton when it processes the input w. From $\hat{\delta}$, it is possible to define $L(A)$, the language accepted by A, as:

$$L(A) = \{w \mid \hat{\delta}(q_0, w) \in F\}.$$

Finally, to define a Nondeterministic Finite Automaton (NFA), the transition function is given by $\delta : Q \times \Sigma \rightarrow \mathcal{P}(Q)$, where $\delta(q, a)$ returns a set of possible next states. That is, for a given state $q \in Q$ and an input symbol $a \in \Sigma$, the function δ produces a subset of Q.

3. Related Work

Given the scarcity of direct approaches to introduce automata in Basic Education, this section presents two related studies that illustrate how automata concepts have been introduced to children and adolescents.

The work of Isayama et al. (2016) investigates the feasibility of introducing fundamental automata theory concepts, specifically DFA, to students of Basic Education (ages 9-12) using gamification. To this end, the authors developed a DFA-based puzzle game in which students solved challenges involving the recognition or labeling of automata based on input sequences. Analysis of game logs of 90 children revealed that approximately 60% achieved an accuracy rate of 70% or higher in the questions, suggesting that students at this age are capable of grasping basic automata concepts. The study also discusses students' deeper understanding of the concepts, the challenges of early introduction, and the effectiveness of the gamification approach.

Hartmann et al. (2001) present the Kara educational programming environment, which uses finite state machines (i.e., finite automata) as its computational model to introduce fundamental programming concepts in a visual way. Designed to support general education, Kara enables beginners to specify behaviors through state machines, controlling a virtual ladybug in a simple and interactive environment. Kiesmüller (2009) demonstrated the use of this tool to automatically diagnose problem-solving strategies adopted by secondary school students. By recording and analyzing student interactions with the system, the author identified different behavioral patterns and argued that providing adaptive feedback can foster learner autonomy and improve the quality of the learning process.

Unlike previous approaches, which rely on software tools that require technological infrastructure, our proposed activity was designed to be more flexible. It can be implemented in both plugged and unplugged formats, as it presents a structured sequence of learning tasks that can be adapt to diverse school contexts, including low-resource settings. Furthermore, none of the reviewed studies adopt a structured PBL methodology to introduce automata in Basic Education. This absence reinforces the innovative character of the proposal and highlights its potential to fill a relevant pedagogical gap.

4. Our Proposal

To support students' understanding of automata concepts in Basic Education, we developed an activity composed of seven sequential tasks. The reason for the sequence in which concepts are introduced was presented by Silva et al. (2024) . Each task introduces a fundamental concept of finite automata in a progressive and engaging manner, following the concept of "low floor, high ceiling" [Resnick 2017]. The activity takes place in a scenario in which students take on the role of young cyber defense scientists tasked with investigating and restoring a compromised system.

Task 1, which has as its objective to identify the final state, presents the following narrative: *"We intercepted a command tape sent by an intruder. We must determine where the defense system stopped after processing it."* In the proposed task, students receive a complete input tape and an automaton. Before running the tape, they are challenged to predict the final state. Then, they simulate the execution step by step to verify their hypothesis. The concepts covered include **states** and **transitions**, with the main objective of introducing the notions of **state transitions** and **final states**, while developing observational skills by linking input sequences to system behavior.

Task 2, which has as its objective to explore the impact of different initial states, presents the following narrative: *"We suspect the intruder injected the code through multiple access points. Let us test the tape starting from different initial states to analyze its effects."* In this task, students run the same input tape from different initial states and observe how the final state changes. The concepts covered include **initial states** and **transitions**. The objective is to highlight the importance of the initial state and to reinforce how transitions depend on both the current state and the input symbol.

Task 3, which has as its objective to verify whether a word will be accepted or rejected, presents the following narrative: *"A new digital signature was intercepted. Will the firewall accept or reject this sequence?"* In this task, students simulate reading a given tape and determine whether the automaton ends in an accepting state. The concepts covered include **acceptance or rejection of input words**. The objective is to introduce the concept of acceptance by finite automata and relate it to real-world scenarios such as access control.

Task 4, which has as its objective to introduce the concept of undefined transitions, presents the following narrative: *"Some sequences are causing system crashes. We must identify commands that trigger undefined transitions."* In this task, using finite automata with partial transition function, students test input sequences and identify failure points caused by undefined transitions. The concepts covered include **undefined transitions**. The objective is to explore the limitations of finite automata and raise awareness of edge cases in the behavior of the system.

Task 5, which has as its objective to construct an input tape, presents the following narrative: *"We know where verification starts and where it must end. Let us create a command sequence that transitions the system safely."* In this task, given a finite automata and a partially filled tape, students must complete the sequence so that the automaton reaches a final state. The concepts covered include word construction, transition planning, and acceptance. The objective is to encourage analytical thinking and problem solving by **building valid input strings (words)**.

Task 6, which has as its objective to introduce the concept of language recognition, presents the following narrative: *"We need to understand all the patterns recognized by the system. Let us map its secure vocabulary."* In this task, students generate all possible strings (words) and test them against the automaton, compiling a set of accepted words. The concept covered is the **recognized language**. The objective is to support an intuitive understanding of the language recognized by finite automata.

Task 7, which has as its objective to specify an automaton, presents the following narrative: *"The final module must be reprogrammed. According to the new security policies, design a module that accepts all words ending in 'ab'."* In this task, students design a finite automata that accepts a given regular language and present it to their peers, justifying its structure and testing its correctness. The concept covered is **automaton specification**. The objective is to consolidate concepts learned previously through the design of a finite automata based on specific language requirements.

The sequence of tasks aligns with the core principles of PBL by providing students with authentic and contextualized challenges that require active inquiry, hypothesis testing, and collaborative problem-solving. Each mission begins with a motivating narrative that introduces a problem to be explored, such as detecting intrusions or validating access, placing students in the role of decision makers within a meaningful scenario. Rather than passively receiving content, learners are encouraged to build their understanding of automata concepts by engaging with concrete problems, formulating strategies, and iterating on their reasoning based on feedback from simulations or peers.

5. Final Remarks

This work-in-progress paper presented a proposal of activity to introduce automata, specifically finite automata, into Basic Education, structured around the PBL methodology. Through a sequence of contextualized tasks, students are challenged to investigate, test, and solve problems embedded in a cyber defense narrative, progressively building an understanding of formal models. The flexibility of the activity, adaptable to both plugged and unplugged formats, makes it suitable for a wide range of school environments, including those with limited access to technological infrastructure. This is particularly relevant in countries such as Brazil, where Computer Science Education is still being consolidated within the official curriculum.

Future work includes consolidating the proposal and expanding it to address more advanced automata models, such as pushdown automata, and incorporating additional active learning strategies, including gamification and project-based learning. Finally, validation with the target audience is planned.

References

Barrows, H. S. (1996). Problem-based learning in medicine and beyond: A brief overview. *New Directions for Teaching and Learning*, 1996(68):3–12.

Battal, A., Afacan Adanır, G., and Gülbahar, Y. (2021). Computer Science Unplugged: A Systematic Literature Review. *Journal of Educational Technology Systems*, 50(1):24–47.

Berbel, N. A. N. (2011). As metodologias ativas e a promoção da autonomia de estudantes. *Semina: Ciências Sociais e Humanas*, 32(1):25–40.

Brazil (2022). Computação: Complemento à BNCC. Disponível em: `https://portal.mec.gov.br/docman/fevereiro-2022-pdf/236791-anexo-ao-parecer-cneceb-n-2-2022-bncc-computacao/file`. Accessed: Jun. 18, 2025.

Ching, Y.-H. and Hsu, Y.-C. (2024). Educational Robotics for Developing Computational Thinking in Young Learners: A Systematic Review. *TechTrends*, 68(3):423–434.

Hartmann, W., Nievergelt, J., and Reichert, R. (2001). Kara, finite state machines, and the case for programming as part of general education. In *Proceedings IEEE Symposia on Human-Centric Computing Languages and Environments (cat. no. 01th8587)*, pages 135–141. IEEE.

Hopcroft, J. E., Motwani, R., and Ullman, J. D. (2001). Introduction to Automata Theory, Languages, and Computation. *ACM Sigact News*, 32(1):60–65.

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.

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.

Kite, V., Park, S., and Wiebe, E. (2021). The Code-Centric Nature of Computational Thinking Education: A Review of Trends and Issues in Computational Thinking Education Research. *Sage Open*, 11(2):21582440211016418.

Mogensen, T. Æ. (2024). *Introduction to Compiler Design*. Springer Nature.

Ramos, M. V. M. (2009). Ensino de linguagens formais e autômatos em cursos superiores de computação. *Revista de Computação e Tecnologia (ReCeT)*, 1(1):22–34.

Reeve, J. (2009). Why Teachers Adopt a Controlling Motivating Style Toward Students and How They Can Become More Autonomy Supportive. *Educational Psychologist*, 44(3):159–175.

Resnick, M. (2017). *Lifelong Kindergarten: Cultivating Creativity Through Projects, Passion, Peers, and Play*. MIT press.

Resnick, M., Maloney, J., Monroy-Hernández, A., Rusk, N., Eastmond, E., Brennan, K., Millner, A., Rosenbaum, E., Silver, J., Silverman, B., et al. (2009). Scratch: Programming for All. *Communications of the ACM*, 52(11):60–67.

Silva, J. V., Silva Junior, B. A., Cavalheiro, S. A. C., and Foss, L. (2024). Exploring Automata Theory with an Educational Activity Using Graph Grammar for K-12 Education. In *Simpósio Brasileiro de Informática na Educação (SBIE)*, pages 329–342. SBC.

UNESCO Institute for Statistics (2012). International Standard Classification of Education: ISCED 2011. Technical report, UNESCO Institute for Statistics, Montreal, Canada. Accessed: Jul. 11, 2025.