Bringing Logic Programming to primary school: a teacher training course

Laura Andrea Cecchi[1],*, Jorge Pablo Rodríguez[1]

[1]*Grupo de Investigación en Lenguajes e Inteligencia Artificial (G.I.L.I.A.)*
Facultad de Informática, Universidad Nacional del Comahue
Neuquén, Argentina

Abstract

The early incorporation of Computer Science (CS) into mandatory education is crucial for developing Computational Thinking (CT) and Logical Thinking (LT) in children. Logic Programming (LP), a paradigm based on formal logic, is essential for fostering specialized knowledge in STEM, particularly in CS and to develop CT and LT.

The insufficient teacher training hinder effective LP teaching. To address this, initiatives such as primary school LP experiences, specialized teacher training, implementation support, and raising awarenes among policymakers are recommended. This work introduces a teacher training course to integrate LP into primary education. We describe the course design and the session plan.

We conducted a primary school teacher training course in Neuquén, Argentina. Teachers, unfamiliar with logic concepts and programming, provided positive feedback. They agreed that LP is viable for primary education. Despite limited empirical evidence, the results were encouraging and showed that effective integration of LP through gamification is possible.

Keywords

Logic Programming Education, Prolog, Computational Thinking, Logical Thinking, Primary Education, Teacher professional development

1. Introduction

The early incorporation of Computer Science (CS) into mandatory education has attracted considerable attention. The pivotal shift from ICT-oriented subjects to rigorous CS concepts is widely acknowledged by the international scientific community as crucial for cultivating Computational Thinking (CT) and Logical Thinking (LT) in children [1, 2].

Logic programming (LP) is a programming, database and knowledge representation paradigm based on formal logic. Different concepts such as Computing, Thinking and Logic [3] converge in this paradigm. Thus, LP is an essential practice for developing specialised knowledge in STEM disciplines, particularly in CS, as well as for fostering CT and LT.

However, CS at primary education level is not a mandatory school subject in several countries, for instance in the mayority of the Argentina provinces [4]. Despite increasing adoption in Europe, less than half of countries have included CT skills as part of their current compulsory education curricula [1]. That means that there is no CS subject in the primary school curriculum, even though many institutions have been equipped with technological infrastructure.

Thus, teachers often lack the essential disciplinary knowledge, pedagogical content knowledge, and technological didactics to teach LP effectively. This shortfall in training means that educators are not fully equipped to engage students in meaningful LP learning experiences. Furthermore, educational authorities often lack a deep understanding of the full scope and potential of LP, particularly in regards to Explainable Artificial Intelligence. This gap in knowledge can lead to underestimating the importance of LP, resulting in missed opportunities to incorporate it effectively into the primary school curriculum and enhance students' computational and logical thinking skills.

PEG 2024: 2nd Workshop on Prolog Education, October, 2024, Dallas, USA.
*Corresponding author.
lcecchi@fi.uncoma.edu.ar (L. A. Cecchi); j.rodrig@fi.uncoma.edu.ar (J. P. Rodríguez)
0000-0001-5236-6715 (L. A. Cecchi); 0000-0002-4697-6477 (J. P. Rodríguez)

To address the effective integration of LP into compulsory education, the following initiatives are starting points to be considered:

1. Carrying out experiences in primary school to teach LP aimed at raising awareness among teachers [5].
2. Deploying specialized training programs to bridge the knowledge gap for in-service teachers.
3. Supporting the implementation processes in schools.
4. Raising awareness among educational policymakers about the potential of integrating LP into primary education.

In this work, we introduce a teacher training course to bring LP to primary school. We detail the course design focusing on LP knowledge, teaching resources for teaching LP, and pedagogical knowledge about LP. Next, we describe the session's plan and discuss an initial experience carried out in Argentina.

2. Course design

The teacher training course is based on an active and collaborative pedagogical approach, designed to introduce programming, promote the teaching of LP, and foster the development of CT and LT in primary school students. This course is part of the initiatives that seeks to promote the broad incorporation of CS into the primary school curriculum.

The training itinerary of the course includes content that addresses three dimensions:

LP Knowledge: What should and could be taught.

Teaching resources for teaching LP: Which teaching resources should be used and why

Pedagogical Knowledge about LP: How it should be taught.

Each of them is intertwined with experiences through the participatory design of teaching proposals for their classrooms, blending LP, CT, and LT with topics in the primary school curriculum.

2.1. LP Knowledge

Identifying *what should and could be taught to children* about LP plays a pivotal role in the educational development and can guide the integration of these topics into the curriculum.

Initially, we introduce the concept of relationship between objects. Following this, we proceed to explain what constitutes a relationship, identifying them in texts, modelling them, determining their participating objects, and defining their arity.

To simplify the understanding of relationship between objects and to facilitate easier modelling of them, we utilise a graphical notation. In Figure 1, we graphical represents the relationship *a person wears clothes*, using prefix notation as *wears(Person,Clothes)*. In this notation, rectangles are used to represent objects, and a rectangle with rounded corners indicates a relationship.

Within this course design approach, knowledge about LP is viewed as a continuum, where it is possible to identify structures that describe a cohesive body of knowledge (practices and concepts). This allows for effective intervention in a particular type of problem [5]:

- Ground Facts: This structure enables the modelling of relationships between objects to construct a knowledge base in an LP.
- Variables and queries: This body of knowledge is developed to allow querying a knowledge base and discovering the collection of objects that meet a specific condition.
- Rules: This cognitive structure's development assists in forming logical relationships between facts, serving as a cognitive mechanism that enables the development of valid arguments. It also serves as a programming tool that allows for the inference of knowledge from known facts.
- Recursive rules: This structure is introduced in a simple way, using recursive processes in daily life. In particularly, we present it through an example inspired by Kowalski in [6]: *Jorge likes everything Laura likes.*

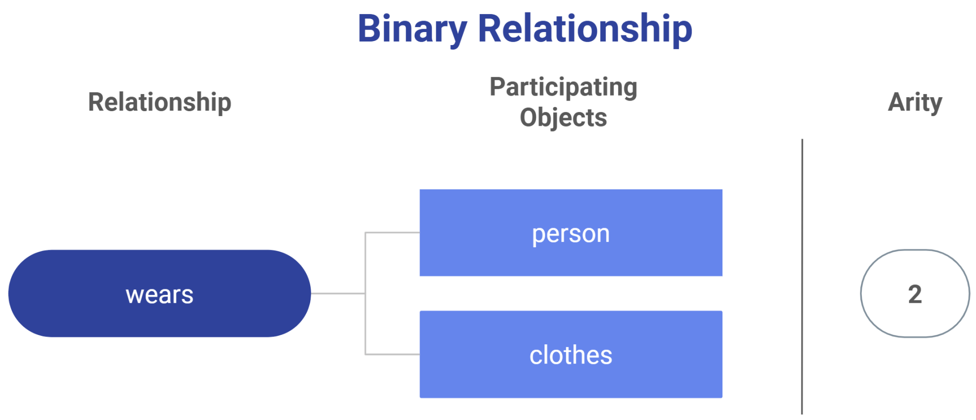

Figure 1: Graphical notation for modelling the relationship wears(Person,Clothes).

2.2. Teaching resources for teaching LP

To ensure the viability of teaching LP, appropriate teaching resources are required. In the training course, we introduce a block-based language for developing programs in the Logic Programming Paradigm: Blockly Prolog [7].

A block-based environment helps focus on the logic of problem-solving and the identification of LP's conceptual components. This environment allows for connecting basic elements of the LP concepts, such as facts and rules, with colours, independent of the text on each block's label. When the concept construction stabilises, it is possible to proceed with an abstract formalisation process in LP terms.

The environment only permits connecting blocks that result in syntactically correct constructions, avoiding teaching and handling Prolog's textual syntax. This aids in cognitively modelling the production of valid solutions from an LP perspective, illustrating their structure and making these construction methods explicit.

Furthermore, Blockly Prolog has a drag-and-drop interface which is easy to use, especially for younger learners and it is a web environment that avoids the inconvenience of installing the software.

Programming tools must meet certain characteristics [8, 9, 10, 11]: *low floor* (easy to get started), *high ceiling* (opportunities to create increasingly complex projects over time) and *wide walls* (supporting many different types of projects so people with many different interests and learning styles can all become engaged). Blockly Prolog meets these features.

In Figure 2, we present a summary of the knowledge structures and an example code in Blockly Prolog.

2.3. Pedagogical Knowledge about LP

The pedagogical proposal we introduce in this training course for teaching LP to children is a gamification approach. Its main components are an immersive narrative and clues or challenges that helps to teach the knowledge structures described above.

The immersive narrative serves as a means to articulate the teaching and learning proposal around the creation of fictional worlds, incorporating LP content along the journey. These narratives offer immersive experiences that encourage students to take on the roles of protagonists in the stories, tales, or adventures. Searching for people or objects, investigating a crime, and proposing actions to tackle potential environmental problems are examples of possible narratives.

A challenge or clue is small problem embedded in the narrative, a piece of the story that we need to solve to progress through the story. Throughout the narrative, consecutive challenges or clues of brief resolution and increasing complexity are presented.

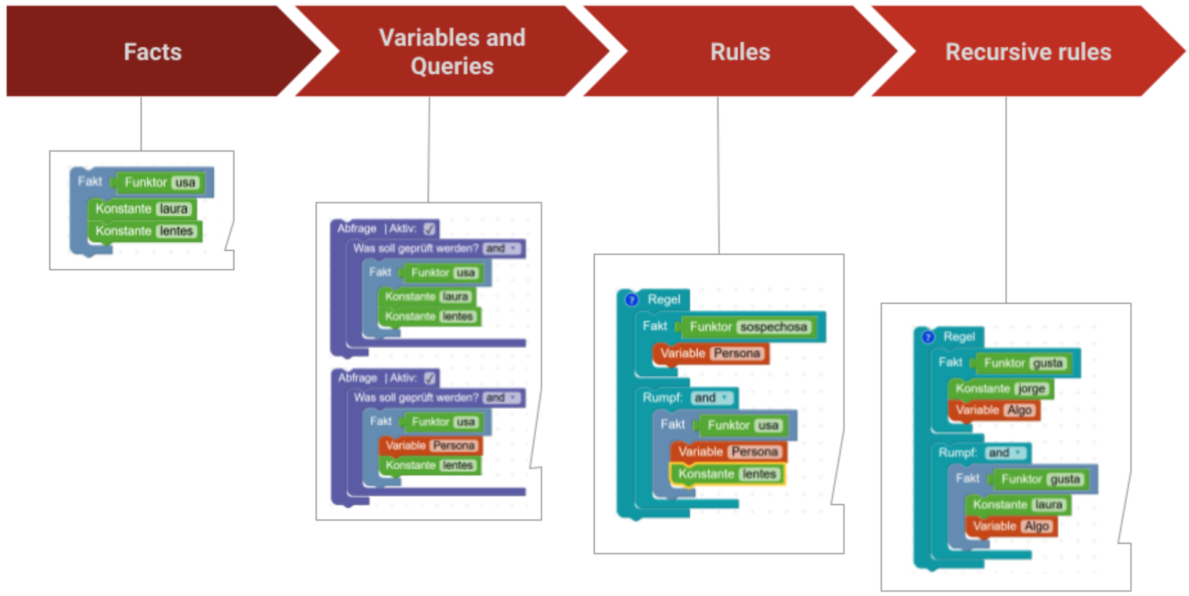

Figure 2: Knowledge structures and an example code in Blockly Prolog.

Challenges or clues are the smallest parts into which a narrative is divided. They offer children quick-resolution problems, aim to achieve something and involve the use of LP practices and concepts. Each challenge includes a description of the situation in the story that contains the information required for constructing the resulting product.

To illustrate the gamification approach, we introduce the *Playing detectives* case to primary school teachers. The proposed game is based on detective fiction storytelling in which the goal is to solve a robbery that occurred at the airport. Participants of this gamification play the rol of detectives (see Figure 3). The game is structured into stages. At each stage, children must face different challenges that, when solved, enable them to advance to the next stage.

A set of clues in natural language will be given to the detectives, showing certain situations that will allow them to circumscribe the suspicious persons. Students must code the clues in Prolog and finally query the logic program on *who is the thief* (see Figure 3).

3. Session's plan

The teacher training course consisted in a series of five 3-hour sessions. Each session explores a different body of knowledge, connecting it to topics in the school curriculum and to CT and LT practices. Teachers use their experience and knowledge about LP to create possible learning environments for their classroom.

The teacher training course has two purposes; on the one hand, it seeks to develop skills and knowledge to teach LP to their students. On the other hand, it aims to help design lessons to teach LP in contextualized scenarios. In this direction, primary school teachers must design a classroom activity for each session. This activity should involve the LP content taught in that session and cover a curriculum subject suitable for their grade level.

Session 01

LP Knowledge: Relations, Objects and Arity. Basic logical connectives. Analysis of Spanish texts and modeling of relationships between objects. Facts, constants and atomic queries.
Pedagogical Knowledge about LP: Block based programming. Establishing relationships between objects

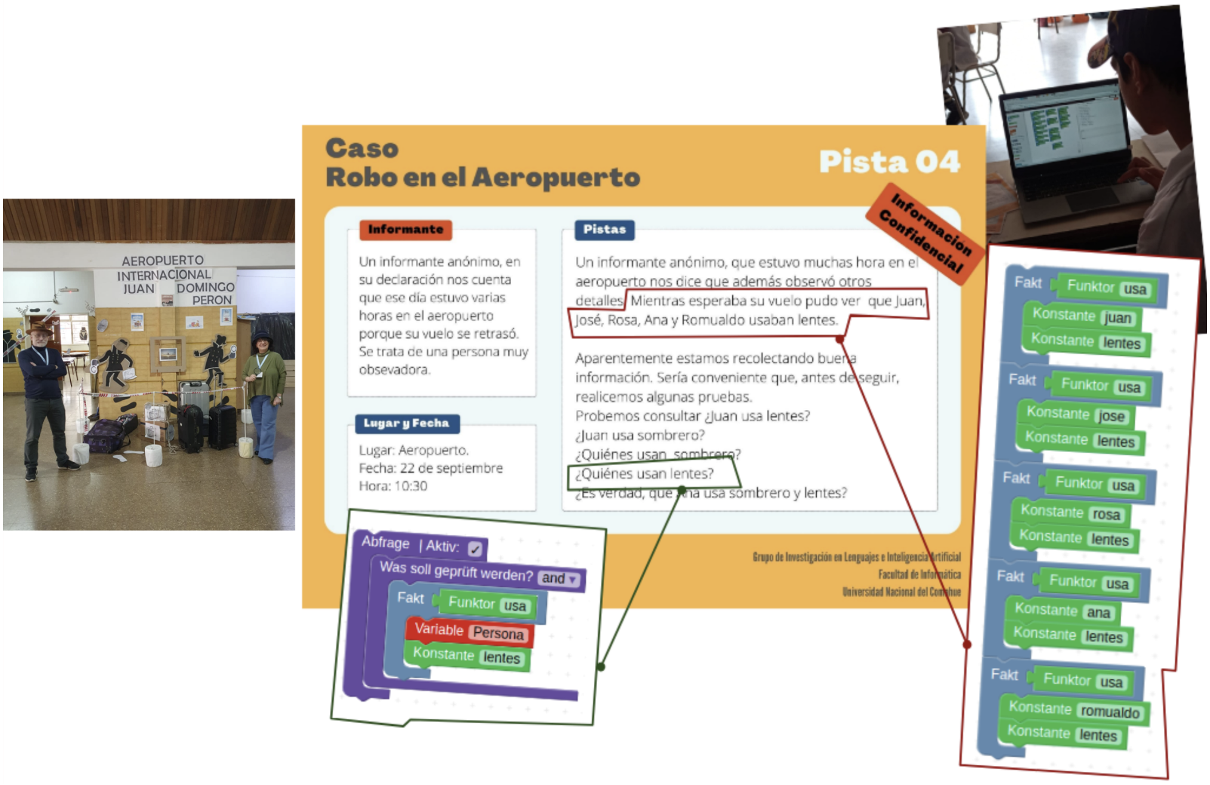

Figure 3: *Playing the rol of detectives*: Clue 4 in natural language and part of the Prolog program that solves it [12].

(LT). Recognizing patterns in similar problems (CT). Participatory design of lessons using relationships between objects.
Teaching resources for teaching LP: Blockly Prolog: introduction to the programming environment. Facts, constants and ground atomic queries in Blockly Prolog.

Session 02

LP Knowledge: Variables and conjunctive queries with variables.
Pedagogical Knowledge about LP: Problem based learning [13, 14]. Participatory design of lessons using variables and conjunctive queries with variables.
Teaching resources for teaching LP: Blockly Prolog: variables and conjunctive queries with variables.

Session 03

LP Knowledge: Rules and recognizing simple recursive rules
Pedagogical Knowledge about LP: Identifying arguments: relationship between the premises that support a conclusion and the conclusion itself (LT). Generate abstractions (CT). Problem based learning. Participatory design of lessons using non-recursive rules.
Teaching resources for teaching LP: Rules, Mathematical and logical operations and relationships in Blockly Prolog.

Session 04

LP Knowledge: In this session, we focus on connecting the topics introduced and consolidating knowledge.

Pedagogical Knowledge about LP: Gamification. Project Based Learning [15] (LT and CT).
Teaching resources for teaching LP: Transitioning from block-based to text-based Prolog.

Session 05

In this session, we will engage in the participatory design of didactic sequences aimed at teaching LP within specific scenarios. The suggestion is to use ecological challenges as topics. LP Knowledge, Pedagogical Knowledge about LP and teaching resources for teaching LP are intertwined in a particular sequence intended for a contextualized context.

In this regard, participatory design actively involves teachers in the creation of didactic sequences for teaching LP. This approach ensures that the designed product meets the needs of their students and is practical for use [16]. Methods based on participatory design also enable teachers to iteratively improve their designs while contributing to a deeper understanding of LP and how to support learning.

4. Preliminary Results and Conclusions

We carried out a first experience of the primary schools teachers training course at Neuquén, Argentina. Initially, 53 primary school teachers attended the workshop, but only 32 completed it. Teachers were unfamiliar with logic concepts and had no previous knowledge about programming. 16 teachers participated in the post-experience survey, from which it is possible to extract positive opinions regarding the contents and didactic of the course that they carried out.

All 16 agreed that LP is a viable paradigm for primary school children and that the topics can be integrated into their lessons. A majority of 81.3% of teachers believe that LP can be taught in sixth and seventh grade, targeting children aged 11 or 12. Additionally, 56% think it can be taught in fifth grade, 37% in fourth grade and 25% in second grade.

Regarding the course, most of the participants stated that the methodology was suitable and appropriate for them, even thought they noted that the five in-person sessions were insufficient. Among the topics addressed —unary relationship, binary relationship, facts, rules, and Blockly Prolog— the most challenging for them was rules.

Despite the lack of sufficient empirical evidence concerning the effectiveness of the course proposed in this research, the results are encouraging. An observation of the quality and relevance of the productions created at the end of the course was used to study the effectiveness of the teaching methods and the progress of the teachers involved.

In each production, it was observed that the knowledge of LP is disciplinarily consistent, that the pedagogical design is appropriate for its students and that the chosen teaching resources are able to support the activity.

The didactic sequence designed by the teachers describe a wide range of curricular topics, including the region's fauna, the thinning of the ozone layer, Earth's subsystems, endangered animals, and the classification of musical instruments, among others.

Participatory design can lead to lessons that are better suited to the needs of students. The sequences developed show a high fit to the curricular and societal contexts for which they were built. The projects produced aligned with the course requirements, consistently integrating LP knowledge through a well-structured gamification approach.

References

[1] S. Bocconi, A. Chioccariello, P. Kampylis, V. Dagienė, P. Wastiau, K. Engelhardt, J. Earp, M. A. Horvath, E. Jasutė, C. Malagoli, et al., Reviewing Computational Thinking in Compulsory Education, Technical Report, Joint Research Centre (Seville site), 2022.
[2] Y. B. Kafai, C. Proctor, A revaluation of computational thinking in k–12 education: Moving toward computational literacies, Educational Researcher 51 (2022) 146–151.

[3] R. Kowalski, Computational logic and human thinking: how to be artificially intelligent, Cambridge University Press, 2011.

[4] J. Rodríguez, M. Cortez, La posición de las Ciencias de la Computación en el diseno curricular para la escuela secundaria argentina: Una revisión sistemática, Electronic Journal of SADIO (EJS) 19 (2020) 136–150.

[5] J. P. Rodríguez, L. A. Cecchi, Logic Programming in Primary School: Facing Computer Science at an Early Age, in: 2024 L Latin American Computer Conference (CLEI), 2024, pp. 1–9. doi:10.1109/CLEI64178.2024.10700103.

[6] R. Kowalski, Logic for problem solving, Department of Computational Logic, Edinburgh University, 1974.

[7] Niklas Holtz, Homepage Blockly Prolog - Universität Oldenburg, 2024. http://www.programmierkurs-java.de/blocklyprolog/.

[8] S. A. Papert, Mindstorms: Children, computers, and powerful ideas, Basic books, 1980.

[9] M. Guzdial, Programming environments for novices, in: S. Fincher, M. Petre (Eds.), Computer science education research, Taylor & Francis, 2005, pp. 137–164.

[10] S. Grover, R. Pea, Computational thinking in k–12: A review of the state of the field, Educational researcher 42 (2013) 38–43.

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

[12] L. A. Cecchi, J. P. Rodríguez, V. Dahl, Logic programming at elementary school: Why, what and how should we teach logic programming to children?, in: D. S. Warren, V. Dahl, T. Eiter, M. Hermenegildo, R. Kowalski, F. Rossi (Eds.), Prolog - The Next 50 Years, number 13900 in LNCS, Springer, 2023.

[13] D. E. Allen, R. S. Donham, S. A. Bernhardt, Problem-based learning, New directions for teaching and learning 2011 (2011) 21–29.

[14] J. Fagerlund, P. Häkkinen, M. Vesisenaho, J. Viiri, Computational thinking in programming with Scratch in primary schools: A systematic review, Computer Applications in Engineering Education 29 (2021) 12–28.

[15] A. Saad, S. Zainudin, A review of Project-Based Learning (PBL) and Computational Thinking (CT) in teaching and learning, Learning and Motivation 78 (2022) 101802. Https://doi.org/10.1016/j.lmot.2022.101802.

[16] B. DiSalvo, J. Yip, E. Bonsignore, D. Carl, Participatory design for learning, in: Participatory design for learning, Routledge, 2017, pp. 3–6.