Logic and Answer Set Programming in High School: Two Learning Unit Experiences

Kristian Reale[1,*]

[1]Department of Mathematics and Computer Sciences, University of Calabria, Rende (CS), Italy

Abstract

This paper presents the design and implementation of two learning units for teaching Answer Set Programming (ASP) to high school students, with a focus on declarative problem solving approaches in Artificial Intelligence. These units were carried out in classroom settings at two different schools, aiming to develop logical reasoning and problem-solving skills through declarative programming. The paper discusses the teaching methods employed and the outcomes observed in both educational contexts. The findings provide practical information for educators who are interested in considering the integration of logic programming into high school curricula.

Keywords

Answer Set Programming, High School, Logic Programming

1. Introduction

In the modern society where an ongoing digital transition involves importantly culture and economy, also the school education is called to evolve in order to align with this progression and promote ongoing enhancement. In recent years, the subject of *Computer Science* has been incorporated into educational programs and has gradually gained space in school curricula around the world, evolving in accordance with technological progress. The STEM approach (Science, Technology, Engineering, Mathematics) [1] has played a key role in this shift, introducing students to topics like coding, data, networking, and Artificial Intelligence (AI) from an early age. More recently, even schools with a humanistic focus, such as Classical High Schools, have begun exploring how digital and computational thinking can enhance their educational mission by introducing new interdisciplinary programs with the aim to connect humanities and technology, encouraging dialogue between philosophy, literature, science, and AI.

A key element of this evolution is the teaching of logic, which provides a critical foundation for structured thinking and problem solving [2, 3] and allows students to develop reasoning skills to navigate complex digital environments. In this setting, Answer Set Programming (ASP) [4], an advanced logic-based paradigm, is a good candidate for enhancing logic skills since it is an expressive and versatile logic programming paradigm for the definition of complex computational problems in a clear and fully declarative manner. The declarative nature and expressive power of ASP have led to the development, within the scientific community, of support systems and efficient solvers such as Clingo [5] and DLV [6]. This has made it possible to exploit ASP in a wide range of applications in areas such as planning, workflow, optimization problems, and more.

However, despite the recognized importance of logic, ASP remains rarely taught in schools, although it is widely included in university courses. Nevertheless, while some teaching experiences have been proposed, as well as advanced tools to assist students in learning ASP [7, 8, 9, 10], it has received little attention in High Schools, especially among adolescents who focus primarily on traditional study methods typical of secondary education.

This paper describes the design and implementation of two educational activities aimed at introducing Answer Set Programming into High School curricula, exploring the potential of logic programming as a

PEG 2025: 3rd Workshop on Prolog Education, September 2025, Rende, Italy.

*Corresponding author.

kristian.reale@unical.it (K. Reale)

0000-0002-5988-2429 (K. Reale)

technological and conceptual bridge between different disciplines: scientific, classical, and professional. The educational experiences were implemented through two Italian distinct educational activities carried out as part of National-Level projects for high school institutions. Both projects aimed to delve into topics that are typically not covered in standard high school curricula and explore new advanced scientific subjects often reserved for university courses or based on scientific research findings.

2. Overview of the Conducted Activities

The target institutions for the activities were two high schools of the province of Reggio Calabria, Italy. In particular, the first high school is known as “IIS Francesco La Cava” (IIS La Cava) located in Bovalino (Reggio Calabria). This school offers distinct fields of study: Vocational Institute for Commercial Services, Classical High School, and Scientific High School. The second institution is known as “IIS Pietro Mazzone” (IIS Mazzone) located in Roccella Ionica (Reggio Calabria). This institution provides some fields of study, including Scientific High School.

The first educational activity, conducted at the “IIS Mazzone”, is titled **“The Logic of Artificial Intelligence"** and is part of the *Piano Lauree Scientifiche*[1], a national initiative aimed at generating interest and enthusiasm for science. The activity, which had a duration of just 12 hours as established by the main formal initiative, was designed to introduce high school students to the fundamentals of Logic-Based Artificial Intelligence, with a particular focus on declarative problem solving through the use of Answer Set Programming. The course combined theoretical instruction with practical lab sessions, encouraging students to work in groups to apply what they had learned by designing simple AI systems based on a declarative approach. The final challenge consisted of developing a logic-based AI agent capable of playing a card game, offering students a hands-on opportunity to engage with key concepts in logical reasoning.

The second education activity, conducted at the “IIS La Cava”, is called **“Smart Village: Planning the Paths of Thought in a Digital Town"** and is part of a government-funded educational program for student skill development[2]. The activity, which had a duration of 30 hours as defined by the main institutional initiative, was designed around the idea of developing a tourist navigation system for the town of Bovalino, integrating various technologies and programming approaches suitable for high school learners. The final part of the activity focused on Answer Set Programming, with the aim of teaching students how to use the language to build the intelligent reasoning component of the system. This allowed them to experience how logic-based AI works in practice, and how declarative programming can solve real-world problems like finding the best route in a town while avoiding roadblocks.

3. Learning Plan Design and Implementation

In this Section, the learning units developed for each educational activity are introduced. Particular attention was given to the academic background of the students and prior knowledge, which were analyzed to design appropriate and meaningful learning experiences for each specific school context.

3.1. The Logic of Artificial Intelligence Learning Unit

For the “The Logic of Artificial Intelligence” educational activity, the following Learning Unit was proposed and carried out.

Learning Unit: The Logic of Artificial Intelligence

[1] https://www.istruzione.it/dg_studente/piano_nazionale_lauree_scientifiche.shtml

[2] PON Project “Smart Village: Un Paese Nel Digitale" Module “Pianifichiamo i Percorsi del Pensiero" codice identificativo “10.2.2A-FSEPON-CL-2018-535"

Conducted at:	IIS “Mazzone"
Duration:	12 hours
Number of students:	14
Age range:	15–18 years
Institutes of origin and background:	All the students come from the *Scientific High School, Applied Sciences track* and have Basic Logic and Object-Oriented Programming in C++ as skills.
Objective:	Development of an Artificial Intelligence system for a card game of their choice
Course Topics:	- Introduction to both Symbolic and Data-Driven Artificial Intelligence. - Difference between procedural and declarative languages. - Logic and Answer Set Programming. - Practical demonstration: how to build an AI sistem for the “Half card game”.
Evaluation Method:	Final multiple-choice questionnaire covering the concepts presented in the course, including an introduction to Artificial Intelligence, concepts of propositional and first-order logic, general questions on applying logic in everyday life, and the rules and constraints of ASP. Group work focused on the development of a mini project of their choice: - Two groups developed AI for the game of Briscola - One group attempted to build AI for the game of Scopa
Final Results:	Final questionnaire with over 75% correct answers. One fifth-year student wrote the entire AI for Briscola with some errors. Another group of two students completed the AI system for Briscola. The remaining students started well but gave up after defining a few rules.

As shown in the learning unit, at “IIS Mazzone” High School, 14 students from the Scientific High School field, aged 15-18, participated in this educational experience. All of them possessed programming skills, particularly in C++, and some of them had some knowledge of Java.

The contents were presented to the students as follows. The first part of this activity introduced Artificial Intelligence, emphasizing the distinction between Symbolic AI, which generally exploits a declarative approach, and Machine/Deep Learning. Since the primary objective of the activity was to use Logic in Artificial Intelligence, the entire emphasis was placed on the declarative problem solving approach, which typically utilizes Answer Set Programming. The activity was divided into two components: a lecture and a laboratory session. During the lecture portion, slides were presented to the students, providing an overview of Artificial Intelligence as a scientific field, with a particular focus on the application of logic in AI. Specifically, Propositional Logic and a brief introduction to First Order Logic were explained. Following that, an introduction to ASP was provided, beginning with a distinction between procedural programming languages and declarative programming languages, which is the underlying nature of ASP. The laboratory session involved dividing the students into groups and assigning them the task of experimenting with the ASP language by creating an Artificial Intelligence system capable of playing a card game of their choice.

Prior to commencing the task, a practical demonstration was conducted to guide the students on how to create an Artificial Intelligence system for the *Seven and a Half* [11] card game using a 40-card italian deck. In the *Seven and a Half* game, the rules state that players draw cards trying to get as close as possible to 7.5 points without going over. The picture card "the King of Coins" is special, and drawing too much can result in a bust. In the following, the ASP program of the *Seven and a Half* game proposed to the students is described, which consists of a single move of the game.

r_1 : `askCard | doNotAskCard.`
r_2 : `anyCardInTable :- table(Number, Suite).`
r_3 : `sumTableValues(0) :- not anyCardInTable.`
r_4 : `sumTableValues(S) :- S = #sum{Value, Number, Suit:`

```
                          numericCardValue(Number, Suit, Value),
                          table(Number, Suit)}.
r5 : tableSum(S) :- sumTableValues(S).
r6 : totalSum(S) :- tableSum(S1), owns(Number, Suit),
                          numericCardValue(Number, Suit, V), S = S1 + V.
r7 : :- askCard, tableSum(S), S >= 8.
r8 : doNotAskCard :- owns(king, coins).
r9 : doNotAskCard :- table(king, coins).
r10 : askCard :- tableSum(S), fuzzy(Range), S < Range, not doNotAskCard.
r11 : fuzzy(Range) :- #rand(4, 6, Range).
```

The proposed ASP program runs in a loop at every turn of the AI system, with the input facts being updated to reflect the current cards on the table and the card held by the AI system.

Rule r_1 is a disjunctive rule that allows the system to ask for another card or not. It represents the main output of the system, indicating the possible decision of whether to take a card or not. Rule r_2 says that if there is at least one card on the table, then $anyCardInTable$ is true. Used later to determine whether the table is empty. Rule r_3 and r_4 calculate the total value of cards currently on the table by summing the value of each card. Since in this case rule r_3 is redundant because rule r_4 already computes 0 when the predicate $table$ is false, it is kept to help students explicitly model the case of an empty table. Rule r_5 and r_6 compute the total score the player would have if they added their card (owns) to the current table sum (*totalSum* = *value* on *table* + *value* of card in hand). Rule r_7 is a constraint that forbids asking for a card if the table sum is 8 or more. Since 7.5 is the maximum, going over is a bust. Rules r_8 and r_9 state that if the player owns the King of Coins or the King of Coins is already on the table, the system should not ask for another card to avoid risking over 7.5. Rules r_{10} and r_{11} introduce a fuzzy element through randomness by assigning a random threshold (between 4 and 6) to the *fuzzy(Range)* predicate. This threshold is then used in rule r_{10} to determine whether the player should ask for a card, effectively simulating variability in risk tolerance.

For their project, some students chose to develop an AI system for the game Briscola [12], and one group even managed to build an AI strong enough to beat the instructor in a play session. Another group attempted to create an ASP program capable of playing Scopa [13], but without success. This difference in outcomes reflects the previous experience and background of the students: while those with stronger logical and programming skills were able to design an ASP program from scratch, others, less accustomed to formalizing problems into rules, could only understand and adapt existing programs rather than develop a new one independently. Nevertheless, although not all students were able to build an ASP program from scratch, all of them could read a pre-written one and performed well on the final questionnaire.

3.2. Smart Village: Planning the Paths of Thought in a Digital Town Learning Unit

For the “Smart Village: Planning the Paths of Thought in a Digital Town” educational activity, the following Learning Unit was proposed and carried out.

Learning Unit: Smart Village: Planning the Paths of Thought in a Digital Town

Conducted at:	IIS “La Cava”
Duration at:	30 hours
Number of students:	18
Age range:	15–18 years
Institutes of origin and background:	*Scientific High School, Applied Sciences track:* 4 students mainly skilled on Basic Logic and Object-Oriented Programming with C++; *Classical High School*: 12 students without any Computer Science background in their school program; *Vocational Institute for Commercial Services:* 2 students mainly skilled with just Basic Computer Science.
Objective:	Build a navigator that calculates the route from point A to point B in the city of Bovalino, avoiding closed roads.
Course Topics:	- Introduction to both Symbolic and Data-Driven Artificial Intelligence. - Smart Devices and IoT systems. - Computational thinking and basic programming in Scratch (for Classical High School students). - Microsoft Access. - App Inventor. - Logic and Answer Set Programming
Evaluation Method:	Final multiple-choice questionnaire covering programming concepts in Smart Devices, Scratch, App Inventor, and Java, as well as the concepts of Artificial Intelligence, propositional and first-order logic, rules and constraints of ASP, and finally, practical applications on applying logic in everyday life. Group work focused on the development of a mini project. The division into groups is based on their skills (one group for each Institute): *Scientific High School, Applied Sciences track:* building the ASP program to determine the best route; *Classical High School*: importing Maps coordinates of Points of Interest (POI) into Microsoft Access; *Vocational Institute for Commercial Services:* designing the app using App Inventor.
Final Results:	Final questionnaire with more than 75% correct answers. The students from the *Classical High School* learned to read ASP programs, but were not able to build one from scratch. The students from the *Vocational Institute for Commercial Services* showed significant gaps in their understanding of logic. The students from the *Scientific High School* wrote the ASP program for calculating the route, although with some errors.

The proposed Learning Unit involved 18 students aged 15 to 18, including 12 from a Classical High School, 4 from a Scientific High School, and 2 from a Vocational Institute for Commercial Services, all of whom participated in this educational experience. Among them, only the Scientific High School students had some advanced computer science skills, particularly in programming with C++. The Vocational Institute for Commercial Services students had basic computer science skills, and, finally, the Classical High School students had no computer science background due to the absence of this subject in their curriculum.

The learning unit began with an introduction to Artificial Intelligence (3 hours) and emphasized the significance of equipping devices like smartphones (smart devices) with AI capabilities (3 hours). The attention then shifted to laboratory activities regarding practical application of Answer Set Programming, where the students collaboratively, and with my guidance, had to build an Intelligent Android App Navigator to assist users in navigating the town of Bovalino. Specifically, the Navigator aimed to provide users with the shortest route from point A to point B within the town while considering road closures and diversions. During the laboratory session, the concept of algorithms was first introduced, along with exercises utilizing the tool Scratch [14] (3 hours) . The introduction of Scratch served as a valuable foundation for understanding the tool App Inventor [15] (3 hours), which allows the creation of real Android apps using a similar methodology to Scratch. The focus then transitioned to Java programming (6 hours), and finally, to the utilization of the ASP language for the development phase of the navigation algorithm (10 hours). The remaining two hours were dedicated to completing the final questionnaire.

For the definition of the ASP program that calculates the best route, students were first introduced to the ASP language through explanations and examples, such as the well-known 3-coloring problem. Since ASP is a declarative language that promotes readability, almost all students were able to understand the meaning of the logic rules. The ASP program for the navigator was then built under my guidance and almost all students were able to follow and understand it, although only those from the Scientific High School proposed small variants of some rules.

The final results of the Learning Unit highlighted some differences: students of the *Vocational Institute for Commercial Services* showed gaps in their understanding of logic, reflecting the difficulties they already had in regular curricular subjects. Students of the *Classical High School*, instead, did not build the program from scratch, as they were not used to writing instructions for a computer. Nevertheless, despite these gaps, almost all students were able to read a pre-written ASP program and performed well on the final questionnaire.

In the following, the ASP program developed during the activity is described.

f_1 : `start(1).`
f_2 : `end(7).`
r_1 : `path(X, Y) | noPath(X, Y) :- arc(X, Y).`
r_2 : `visited(X) :- start(X).`
r_3 : `visited(Y) :- path(X, Y), visited(X).`
r_4 : `:- end(X), not visited(X).`
r_5 : `:- interruption(X), visited(X).`
r_6 : `:~ path(X, Y). [1@1, X, Y]`

The fact f_1 and f_2 define the starting node (node 1) and the ending node (node 7) of the desired shortest path where a node is a Point of Interest (POI). Rule r_1 states that for every arc (connection) between nodes X and Y, the program nondeterministically, chooses whether to include it in the path (*path(X, Y)*) or exclude it (*noPath(X, Y)*). Rule r_2 and r_3 state that the starting node is considered visited and if there is a path from node X to node Y and X has been visited, then Y is also marked as visited. This rule recursively marks all reachable nodes along the selected path as visited. The constraint r_4 states that the end node must be visited. The constraint r_5 prevents visiting nodes marked as interruptions. Finally, the weak constraint r_6 tries to minimize the total occurence of the predicate *path* in order to obtain the shortest path.

4. Discussion and Conclusions

The outcomes of both educational activities were satisfactory, because students enthusiastically participated and achieved positive results. All students quickly grasped the principles of logical reasoning thanks to the declarative nature of ASP, which facilitated their understanding and application of the language. This experience not only enhanced their computational thinking skills but also significantly strengthened their critical thinking abilities. By distinguishing between classical imperative programming paradigms and logic-based declarative languages, students developed a more structured and analytical mindset essential for problem-solving.

The guidance provided during the initial laboratory phase gradually diminished as students became proficient in logical thinking within a few hours. Final questionnaires revealed positive feedback, showing that even students without prior computer science experience could effectively engage with logical reasoning through ASP. Furthermore, using ASP to develop an intelligent automatic navigator and an agent capable of playing a card game helped students appreciate the central role of declarative, logic-based approaches in advancing Artificial Intelligence. This logical foundation offers a valuable alternative to traditional, more complex AI methods reliant on neural networks.

These outcomes highlight how prior knowledge and active engagement influence performance of students. The ones with stronger backgrounds, such as those from Scientific High Schools, were able to create functional AI programs with impressive results, while others faced challenges with fundamental concepts like logic and program construction. This underscores the importance of targeted support and tailored teaching strategies when introducing complex subjects like Artificial Intelligence and Answer Set Programming. The success of some students in developing an AI system capable of playing Briscola and defeating also the instructor in a game session, demonstrates how hands-on, project-based learning fosters deeper understanding, critical thinking, and motivation.

This experience confirms that subjects initially perceived as difficult by high school students can become accessible through practical, hands-on activities. Providing opportunities for students to create their own games (such as the card games developed in this project) and real-world tools like the App Navigator (familiar from their daily smartphone use) served as powerful motivators. This approach enabled students to actively engage with complex concepts by building tangible products, thereby reinforcing their logical and critical thinking skills. Moreover, the rapid acquisition of a declarative logic language like ASP by the students suggests its strong potential for inclusion in high school computer science curricula, where fostering logic and critical thinking is essential for preparing students to navigate the challenges of the digital age.

Acknowledgments

The school experiences have been partially supported by: (i) PON Project “Smart Village: Un Paese Nel Digitale” Module “Pianifichiamo i Percorsi del Pensiero” codice identificativo “10.2.2A-FSEPON-CL-2018-535”; (ii) Piano Lauree Scientifiche 2017/18 previsto dal D.M. 1047/2017 con l’"Accordo di Partenariato” in via di definizione tra l’Università degli Studi di Pavia/Università di Milano e l’Università della Calabria. Other thanks: Caterina Autelitano, the School Principal of “IIS La Cava” High School in Bovalino, played a crucial role in enabling the PON project; Francesco Bonaparte provided support in the definition of the project and its successful continuation; Rosita Fiorenza, the School Principal of “IIS Mazzone” High School in Roccella Ionica, who made possible the collaboration with the University of Calabria for the PLS project.

Declaration on Generative AI

During the preparation of this work, the author used ChatGPT in order to: Grammar and spelling check. After using this tool, the authors reviewed and edited the content as needed and take full responsibility for the publication’s content.

References

[1] Global Sevilla, STEM Learning Methods, https://www.globalsevilla.org/stem-learning-methods, 2025. Accessed: 2025-06-27.
[2] E. McCain, Education in Logic: Building Critical Thinkers, 2024. URL: https://espressocoder.com/eduction-in-logic/, accessed: 2025-06-27.
[3] Lcom Team, Students as Logical Thinkers: How Computer Science Teaches Real-World Skills, 2024. URL: https://www.learning.com/blog/students-as-logical-thinkers/, accessed: 2025-06-27.

[4] M. Gelfond, V. Lifschitz, Classical Negation in Logic Programs and Disjunctive Databases. New Generation Computing 9, 365-385, New Generation Computing 9 (1991) 365–385. doi:10.1007/BF03037169.

[5] M. Gebser, R. Kaminski, B. Kaufmann, M. Ostrowski, T. Schaub, P. Wanko, Theory solving made easy with clingo 5, in: Proc. of ICLP Technical Communications, volume 52 of *OASIcs*, Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2016, pp. 2:1–2:15. URL: https://doi.org/10.4230/OASIcs.ICLP.2016.2. doi:10.4230/OASICS.ICLP.2016.2.

[6] M. Alviano, F. Calimeri, C. Dodaro, D. Fuscà, N. Leone, S. Perri, F. Ricca, P. Veltri, J. Zangari, The ASP System DLV2, in: M. Balduccini, T. Janhunen (Eds.), Logic Programming and Nonmonotonic Reasoning, Springer International Publishing, Cham, 2017, pp. 215–221.

[7] A. Dovier, P. Benoli, M. C. Brocato, L. Dereani, F. Tabacco, Reasoning in high schools: Do it with ASP!, in: CEUR Workshop Proceedings, volume 1645, CEUR-WS, 2016, pp. 205–213. URL: http://ceur-ws.org/.

[8] V. Nguyen, Y. Zhang, K. Jung, W. Xing, T. Dang, VRASP: A Virtual Reality Environment for Learning Answer Set Programming, 2020, pp. 82–91. doi:10.1007/978-3-030-39197-3_6.

[9] O. Febbraro, K. Reale, F. Ricca, ASPIDE: Integrated Development Environment for Answer Set Programming, in: J. P. Delgrande, W. Faber (Eds.), Logic Programming and Nonmonotonic Reasoning, Springer Berlin Heidelberg, Berlin, Heidelberg, 2011, pp. 317–330.

[10] E. Marcopoulos, Y. Zhang, onlineSPARC: A programming environment for Answer Set Programming, Theory and Practice of Logic Programming 19 (2019) 262–289. doi:10.1017/S1471068418000509.

[11] Pagat.com, Sette e Mezzo, 2007. URL: https://www.pagat.com/banking/sette_e_mezzo.html, Accessed: 2025-06-27.

[12] Pagat.com, Briscola, https://www.pagat.com/aceten/briscola.html, 2025. Accessed: 2025-06-27.

[13] Pagat.com, Scopa, https://www.pagat.com/fishing/scopa.html, 2025. Accessed: 2025-06-27.

[14] Lifelong Kindergarten Group, MIT Media Lab, Scratch, https://scratch.mit.edu/, 2025. Accessed: 2025-07-03.

[15] Massachusetts Institute of Technology (MIT), App Inventor, https://appinventor.mit.edu/, 2025. Accessed: 2025-07-03.