NOVA SCHOOL OF
SCIENCE & TECHNOLOGY

DEPARTMENT OF
COMPUTER SCIENCE

MARIA CAROLINA SEBASTIÃO FERNANDES

BSc in Computer Science

AN ONLINE TOOL FOR LEARNING LOGIC

MASTER IN COMPUTER SCIENCE AND ENGINEERING

NOVA University Lisbon
February 14, 2024

NOVA SCHOOL OF
SCIENCE & TECHNOLOGY

DEPARTMENT OF
COMPUTER SCIENCE

AN ONLINE TOOL FOR LEARNING LOGIC

MARIA CAROLINA SEBASTIÃO FERNANDES

BSc in Computer Science

Adviser: Ricardo Gonçalves
Assistant Professor, NOVA University Lisbon

Co-adviser: João Costa Seco
Associate Professor, NOVA University Lisbon

MASTER IN COMPUTER SCIENCE AND ENGINEERING

NOVA University Lisbon
February 14, 2024

Abstract

In today's educational landscape, the demand for effective online tools for education is increasingly pronounced. Existing platforms often struggle to provide students with a dynamic and interactive learning experience while offering immediate feedback on their progress. This presents a significant challenge that calls for innovative solutions.

This research addresses the need for a comprehensive online platform for logic education, which is a usual part of Computer Science courses syllabus, by proposing the development of an adaptive learning environment using Moodle. The primary objective is to create a platform that seamlessly integrates diverse logic exercises, supports various logic paradigms, and provides real-time feedback to students. Leveraging the flexibility and extensibility of Moodle, the platform aims to enhance student engagement and foster critical thinking and problem-solving skills.

The proposed approach involves a multifaceted exploration of logic education methodologies, the identification of shortcomings in existing platforms, and the design and development of a prototype platform. Through rigorous evaluation and testing, the effectiveness and usability of the platform will be assessed, to provide actionable recommendations for future implementations.

The anticipated results of this research include the development of an online platform that improves logic education in online settings. By offering students an interactive learning experience, the platform is expected to empower them to master logic principles effectively, ultimately enhancing their academic success and professional development.

Resumo

No panorama educativo atual, a procura de ferramentas em linha eficazes para o ensino lógico é cada vez mais acentuada. As plataformas existentes têm muitas vezes dificuldade em proporcionar aos alunos uma experiência de aprendizagem dinâmica e interativa, ao mesmo tempo que oferecem 'feedback' imediato sobre o seu progresso. Isto representa um desafio significativo que exige soluções inovadoras.

Esta investigação aborda a necessidade de uma plataforma em linha abrangente para o ensino da lógica, que é uma parte importante dos cursos de Informática, propondo o desenvolvimento de uma plataforma adaptativa para o ensino da lógica, propondo o desenvolvimento de um ambiente de aprendizagem adaptativo utilizando o Moodle. O objetivo principal é criar uma plataforma que integre sem problemas diversos exercícios de lógica, suporte vários paradigmas de lógica e forneça 'feedback' em tempo real aos alunos. Tirando partido da flexibilidade e extensibilidade do Moodle, a plataforma visa melhorar o envolvimento dos alunos e promover o pensamento crítico e a capacidade de resolução de problemas.

A abordagem proposta envolve uma exploração multifacetada de metodologias de ensino da lógica, a identificação das deficiências das plataformas existentes e a conceção e desenvolvimento de uma plataforma protótipo. De uma plataforma protótipo. Mediante uma avaliação e testes rigorosos, a eficácia e a usabilidade da plataforma serão avaliadas, visando fornecer recomendações acionáveis para futuras implementações.

Os resultados esperados desta investigação incluem o desenvolvimento de uma plataforma 'online' o ensino da lógica em ambientes 'online'. Ao oferecer aos estudantes uma experiência de aprendizagem coesa e imersiva, espera-se que a plataforma os capacite para dominar eficazmente os princípios da lógica, melhorando assim o seu sucesso académico e desenvolvimento profissional.

Contents

List of Figures

1

Introduction

1.1 Motivation

The evolution of learning environments in today's educational landscape necessitates a reevaluation of the tools and resources available to students. As educational practices increasingly transition to online platforms, the demand for effective and comprehensive resources for logic education becomes more pronounced. Among the critical requirements is the need for a centralized environment that provides students with an efficient and streamlined approach to study and practice logic exercises while offering immediate feedback on their progress.

Logic education has relied on disparate resources and tools, often lacking integration and cohesion. Students frequently find themselves navigating multiple platforms and materials to access exercises, study materials, and feedback mechanisms, leading to inefficiencies and fragmented learning experiences.

In response to these challenges, there is a compelling need for a unified platform that consolidates all aspects of logic education, offering students a seamless and integrated learning environment.

Moreover, the importance of immediate feedback cannot be overstated in the context of logic education. Students benefit greatly from timely insights into their performance, allowing them to identify areas for improvement and refine their problem-solving skills iteratively [8].

However, many existing tools fall short in providing such feedback, relying on manual intervention from instructors or lacking real-time assessment capabilities. The instructors would also benefit from having the teaching centralized in one platform where they could easily provide material for the students to practice and also be able to assess the students.

By addressing these needs and challenges, an innovative online platform for logic education can significantly enhance students' learning experiences and outcomes and the instructor's teaching experience. By providing a comprehensive suite of logic exercises, study materials, and immediate feedback mechanisms within a single environment, students can benefit from a more efficient and effective approach to mastering subjects for example logic exercises such as first-order formulation.

By providing an easier way to manage a course, and student's progress, assess students, and provide material for them to practice within a single environment, teachers can benefit from a more efficient, effective, and less time-consuming approach to teaching.

Such a platform not only streamlines the learning process but also empowers students to take ownership of their learning journey, fostering autonomy and self-directed learning skills and facilitating the workload on the instructor improving is teaching experience.

1.2 Problem Formulation

However, the current array of tools available for logic education in online settings reveals a substantial gap. Many existing solutions are confined to rigid structures, limiting their capacity to adapt to evolving pedagogical needs. Installable programs, while effective to some extent, often lack the flexibility necessary for accommodating diverse exercises and supporting various logic paradigms.

The inherent static nature of many tools poses challenges in keeping pace with the different tools that are lectured in a course specifically in the field of logic.

New exercise types, especially those related to advanced topics like first-order logic, struggle to find a place in these environments. Educators face hurdles when attempting to integrate innovative exercises and expand the curriculum beyond traditional propositional logic and when trying to help their students practice and learn in a way that they have immediate interactive feedback without needing the teacher to give the solutions.

1.3 Research

The primary objective of this research is to comprehensively address the identified needs and challenges within logic education in online settings through the development of an innovative online platform to leverage Moodle. This research endeavors to undertake a multifaceted approach, encompassing the investigation of the current landscape of logic education tools and platforms available in online environments, identification of limitations and shortcomings inherent in current solutions, exploration of potential enhancements and features, design and development of a prototype, rigorous evaluation of its effectiveness and usability, and provision of actionable recommendations for future extensions. **Research Questions:**

- 1. What methodologies are commonly employed for the assessment of logic exercises in educational systems?

- 2. What challenges do existing learning tools face?

The literature review will systematically explore relevant studies, theoretical frameworks, and practical applications, enriching the research with a robust theoretical foundation and empirical evidence.

1.4 Document Structure

The document is organized into 6 sections, each addressing specific aspects of the research topic and contributing to a comprehensive understanding of logic education in online settings. In Chapter 1 we provide an overview of the motivation behind the study and formulating the central problem. It outlines the research objectives, highlighting the need to explore existing logic education tools and develop innovative solutions to address their limitations. In Chapter 2 we provide a literature review of multiple existing logic tools explored. In Chapter 3 we present the E-learning environment and its importance and impact for both students and teachers. Explore Moodle passing through studies and applications of Moodle and its more technical details. In Chapter 4 specific logic exercises commonly used in logic education are explored in detail for a better understanding of such. In Chapter 5 we explore some potential enhancements in online learning environments. In Chapter 6 outlines the plan for designing and implementing an innovative online platform for logic education using Moodle. A detailed plan and Gantt chart illustrate the project timeline and milestones, providing a roadmap for future development.

2

Existing Logic Tools

A thorough analysis of current tools has provided a clear view of the educational technology landscape, underscoring the need for an adaptable and comprehensive approach. In this domain, a detailed exploration of diverse e-learning tools and methodologies has revealed a spectrum of strategies for online education.

2.1 Logic for Fun

One of the tools discovered and explored was "Logic for Fun" [11] consists of an online teaching tool, developed around 20 years ago, focused on logical modeling or formalization. The tool is designed to assist students in expressing logical problems and puzzles, emphasizing many-sorted first-order logic by addressing their difficulties. The tool invites users to express logical problems and puzzles, with a solver providing solutions based on the user's input and is designed to be used as an adjunct to courses in logic, critical reasoning, artificial intelligence, or other related fields. Each user can enroll in a course through credentials given by the teacher.

The exercises are categorized into five levels: Beginner, Intermediate, Advanced, Expert, and Logician because students like the idea of progressing through levels. The levels progress in difficulty, covering topics such as basic vocabulary declaration, logic puzzles, advanced logical treatment, state-transition problems, and applications of logic. The tool offers several advantages over traditional formalization exercises, including correct solutions, immediate feedback, and increased student engagement.

2.1.1 System Flow

The site's workflow, illustrated in 2.1 , facilitates a dialogue-driven interaction between students and the system. Students are presented with challenges in natural language, responding by formulating formal encodings. These are evaluated by the solver, with feedback provided in the form of error messages or solutions. The iterative process continues until the student chooses to conclude or pause. Work can be saved for future reference. Interpreting models involves validating constraints against truth standards. If

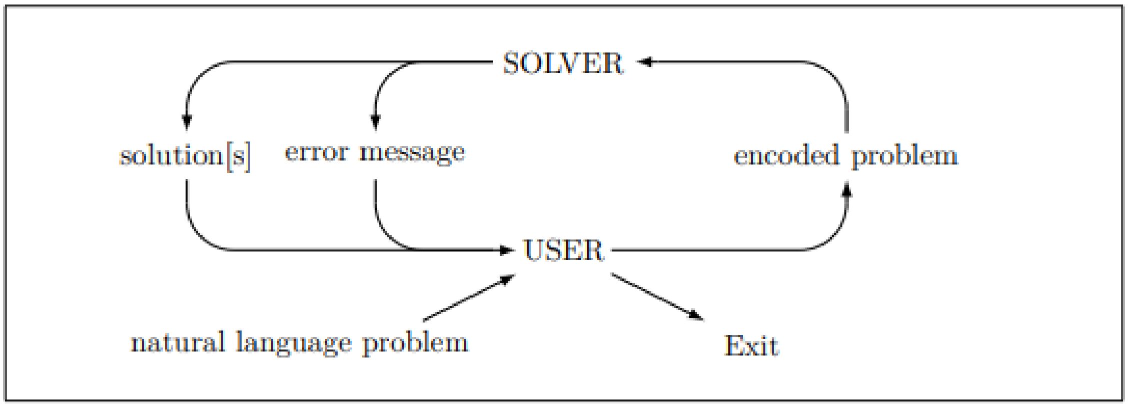


Figure 2.1: Logic for Fun Flow

the solver fails to find a model within the time limit, a "No solution found" message is returned. Multiple models may be provided, with users able to adjust this limit. Each model is presented explicitly, aiding student understanding.

Note that because the language lacks free variables, there's no necessity to treat variable valuations separately from complete interpretations that vary based on the values of names. Therefore, separate handling of variable valuations is unnecessary.

2.1.2 Error Handling and Feedback

Error handling and feedback are very important when evaluating the effectiveness of a tool. Logic for Fun has fundamentally two kinds of errors, syntactic or semantic producing different feedbacks to the user.

Errors are categorized into syntactic and semantic types. Errors of syntax are caught by the parser or the type checker and the solver provides explicit messages, vocabulary used, and parse tree details. Semantic errors are harder to classify, and unexpected or no solutions indicate potential misunderstandings at the semantic level. The tool relies on the search for solutions to identify and report errors.

2.1.3 Diagnosis Tool

The Diagnosis tool automates the process of isolating incorrect constraints in cases where the encoded problem globally has no solution. The tool provides two types of information to the solver: Approximate Models and Unsatisfiable Cores.

In the Approximate Model, the solver can be asked for an assignment of values satisfying as many constraints as possible, treating unsatisfied constraints as "soft." The violated constraints are listed, and the user can mark some as "hard" to resolve, iteratively approaching a solution. The solver can operate in two modes: depth-first branch and bound (providing optimal approximations or detecting unsatisfiable hard constraints) and a local search method like WalkSAT (providing reasonable models without guarantees).

In the Unsatisfiable Cores, the solver can identify a minimal subset of constraints with no solution. Any subset must contain a contradiction, providing potentially useful information for repairing the problem encoding.

2.1.4 Conclusion

While Logic for Fun shows promise in addressing our needs, it does show some limitations such as reduced scope of exercises, implementing mainly first-order exercises, low adaptability for a teacher to integrate into a course and potential issues in the solver performance if there are a high number of students.

Some relevant aspects of this tool are the use of build-in connectives and quantifiers that remove deviation and simplify the data processing, insights into the solver process such as depth-first branch and bound and local search, the separation of errors into two types could be helpfull for a easier detetion, funtions are typed by the user to reduce large propositions, progressive difficulties benefitial for students engagement and imediate feedback provided.

2.2 Iltis

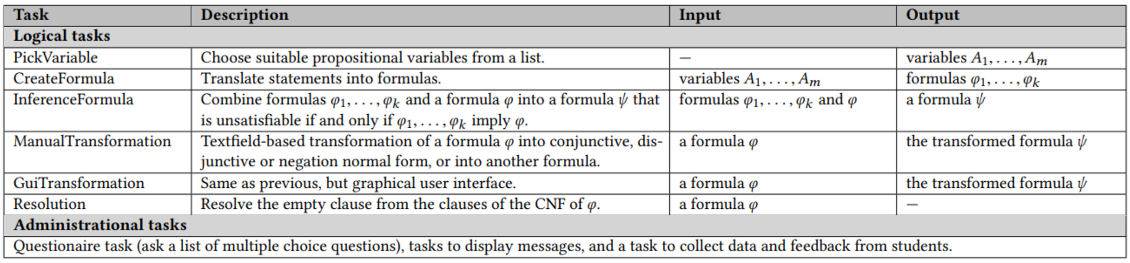
Another tool discovered and explored was “Iltis” [11]. The Iltis project focuses on providing a web-based, interactive system for teaching logical methods in computer science. At the same time, it aims to support learning propositional logic, modal logic, and first-order logic while providing immediate feedback to students. Their key objectives were to support the learning process of modeling knowledge and inferring new knowledge using propositional logic and provide immediate feedback and support to students.

The tool exercises are composed of a sequence of tasks specified in XML format. The tasks follow a model-view-controller pattern, upon user input, the task view generates an action that is sent to the controller. The task controller executes the action, if applicable, and then returns feedback to the task view which is then displayed. There are two types of tasks implemented, logical tasks and administrational tasks, figure 2 displays a list of the currently implemented tasks. This task-based structure allows for the modular inclusion of new types of tasks through developers.

The tools used in its development were Java and Google Web Toolkit in the backend and HTML and Javascript for the webpage.

The immediate feedback is generated by feedback generators and a task can have more than one assigned to it such as checking variables availability and one to find the error in the formulation. The tool has an advanced feedback mechanism that uses reversion rules to go from the student’s solution to the correct solution to find the errors.

From the teachers’ perspective, they have the flexibility to specify exercises in XML, combining small tasks (figure 2) to create comprehensive exercises. Using this system teachers can provide various levels of feedback to their students based on correctness, appropriate use of variables, and in-depth analysis of logical operators and formula parts.



Task	Description	Input	Output
Logical tasks			
PickVariable	Choose suitable propositional variables from a list.	–	variables $A_1, \ldots, A_m$
CreateFormula	Translate statements into formulas.	variables $A_1, \ldots, A_m$	formulas $\varphi_1, \ldots, \varphi_k$
InferenceFormula	Combine formulas $\varphi_1, \ldots, \varphi_k$ and a formula φ into a formula ψ that is unsatisfiable if and only if $\varphi_1, \ldots, \varphi_k$ imply φ.	formulas $\varphi_1, \ldots, \varphi_k$ and φ	a formula ψ
ManualTransformation	Textfield-based transformation of a formula φ into conjunctive, disjunctive or negation normal form, or into another formula.	a formula φ	the transformed formula ψ
GuiTransformation	Same as previous, but graphical user interface.	a formula φ	the transformed formula ψ
Resolution	Resolve the empty clause from the clauses of the CNF of φ.	a formula φ	–
Administrational tasks			
Questionaire task (ask a list of multiple choice questions), tasks to display messages, and a task to collect data and feedback from students.			

Figure 2.2: Iltis tasks implemeted

2.2.1 Conclusion

There are some relevant features from this web-based tool to be taken into consideration when developing our own, such as the concept of modular exercises, its feedback generators according to each task which reduce the complexity, and its array of tasks implemented.

Although this tool has some logic exercises implemented(propositional logic, modal logic, partial coverage of first-order logic), the teacher needs to specify the tasks in XML, this would only be possible for teachers with a certain knowledge. Regarding the management of students, this tool does not display anything.

2.3 Lechef library and ViLLE

Lechef is a javascript framework that enables the creation of exercises for practicing and understanding the behavior of the basic logical gates, and circuits made from those gates. The library supports two types of exercises: simulation, where students manipulate input and output values for given circuits, and design, where students construct circuits based on provided specifications.

The Lechef library itself is a stand-alone library and can be used on any web page, but it is not possible to store the student answers and grades. This problem can be overcome if integrated into a learning environment such as ViLLE [11].

Lechef library provides automatically assessed exercises on logic circuits with visual and immediate feedback. This feedback is shown for each input and output as shown in (figure 3) Regarding the teachers, the automatic assessment provides reduced time and resources to grade and manage the students. Furthermore, using Lechef integrated into ViLLE they can view every student submission, and their feedback and create and personalize their course content.

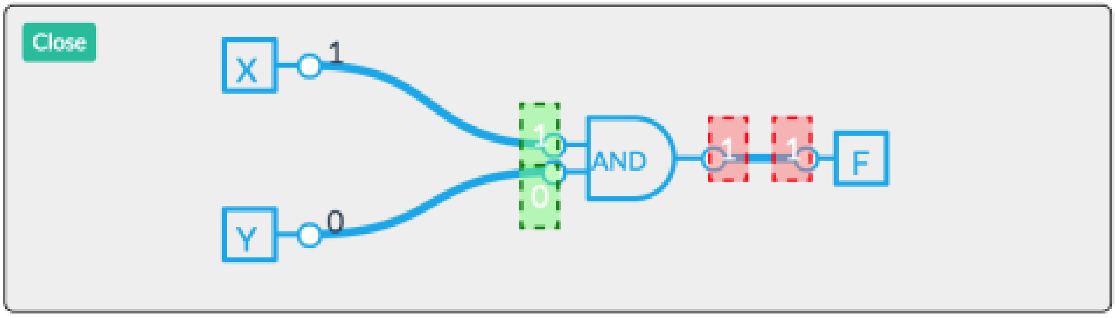


Figure 2.3: Lechef logic circuit feedback

2.3.1 Conclusion

While Lechef primarily focuses on basic logic gates, its structure, and design could inspire incorporating first-order logic exercises aligning with the goal of this research. The integration of a Javascript library with a ViLLE environment produces a more consistent tool enabling personalization, the management of students and the storage of data. Regarding the visual exercises such as graphs, the feedback in each input and output shown in these tools could be helpful.

2.4 Learning Mathematics through Programming Techniques

Online mathematics courses offer a dynamic and interactive learning experience, leveraging multimedia resources, interactive simulations, and adaptive learning systems to engage students in the exploration of mathematical concepts. From algebra to calculus, these courses provide a wealth of opportunities for students to visualize abstract ideas, practice problem-solving skills, and collaborate with peers in virtual learning environments.

The tool explored in this section [4] focuses on high school students understanding of mathematical concepts and problem-solving through programming techniques, specifically focusing on Python due to this language's common usage in artificial intelligence development.

It incorporates an expert system, simulating human expertise in a given area, to aid students in understanding and solving mathematical problems through programming, including a conversational bot developed with the Microsoft Bot Framework and Cognitive Services language understanding intelligent services API. It utilizes the "repl.it" platform, allowing students to edit and execute Python code through an online console, and also integrates machine learning and production rules to understand user intentions and provide appropriate responses. Nine intentions, including loop, conditional, functions, and others, are defined to enhance the learning experience.

The tool, assessed through pre-course and post-course evaluations, demonstrated a positive impact on students' mathematical reasoning, logical programming skills, and knowledge of programming syntax. [4]

2.4.1 Conclusion

The tool shows potential for implementation in the Computer Science syllabus and it's effective to improve students' perception of programming languages and enhance their problem-solving skills. Our primary focus is more on logic exercises such as first-order propositions. Regarding some of the tools used in its development, the chatbot option implemented with Microsoft Bot Framework introduces a new possibility of incorporation of artificial intelligence in learning environments, and the "repl.it" platform could be relevant for the potential implementation of the code-related exercises.

2.5 Summary

Each of the tools has unique characteristics that differentiate them, while Iltis and Logic for Fun focus more on logic exercises (first-order, propositional logic, modal logic) of our interest, the Lechef library integrated into ViLLE and the mathematical tool have more divergent exercise types from our goal still based on logic.

The Lechef library integrated with ViLLE allows teachers to manage students and the storage of data and shows an example of feedback in visual exercises.

The mathematical tool offers potential for implementation in the Computer Science syllabus, the possibility to improve a learning tool by incorporating chatbot and artificial intelligence tools, and a tool for implementing code-related exercises.

Logic For Fun and Iltis tools offer interactive learning experiences with immediate feedback, but they differ in their exercise structures, feedback mechanisms, and teacher flexibility. While "Logic for Fun" provides a more intuitive interface and automated error diagnosis, "Iltis" offers modular exercises and detailed feedback generators. Both tools lack the inclusion of student management.

The "Logic for Fun" exercises can be categorized into different learning difficulties which are helpful for student engagement and are implemented statically while the "Iltis" are implemented through the composition of tasks giving the teacher a more flexible personalization of the course. This structure influences the feedback mechanisms, even tho the diagnosis tool and solver are useful for the first exercise structure in the modular exercises each task has its own feedback generator which reduces the implementation complexity and computational weight. Although, through Iltis, teachers can combine small tasks to create comprehensive exercises, it requires XML knowledge.

3

E-learning and Moodle

E-learning has become a cornerstone of modern education, revolutionizing the way students access and engage with educational content. At the forefront of this revolution is Moodle, a versatile and open-source Learning Management System (LMS) that empowers educators and institutions to create dynamic online learning environments.

This chapter provides an overview of E-learning and explores the key features and trends in adoption of Moodle. By examining the role of technology in shaping pedagogical practices, we uncover the transformative impact of Moodle in facilitating interactive and accessible learning experiences. Through case studies and examples, we illustrate the diverse applications of Moodle across different educational contexts, highlighting its effectiveness in enhancing teaching and learning outcomes.

3.1 E-learning

E-learning, short for electronic learning, refers to the use of electronic technologies, primarily the Internet, to deliver educational content and facilitate learning. It encompasses a wide range of activities, including online courses, virtual classrooms, interactive tutorials, digital simulations, and educational games.

In e-learning, students typically access course materials, lectures, and assignments through digital platforms such as learning management systems (LMS), websites, or mobile applications. These platforms provide a flexible and convenient way for learners to engage with educational content anytime, anywhere, using various devices like computers, tablets, or smartphones.

It offers flexibility, allowing learners to study at their own pace and schedule, from anywhere with internet access. E-learning platforms often include interactive features and multimedia elements for a more engaging learning experience. With personalized learning options, e-learning caters to individual needs and learning styles, making education more accessible and effective.

Online education has many benefits, such as improved professor-student communication, increased student satisfaction in courses, the interaction between students, attendance percentage in comparison to traditional presential classes, homework submission percentage in e-learning in comparison to e-mail submissions, improved group development and homogeneity, and broader access to knowledge [5] [15].

It's proved that higher grades are associated with more frequent platform access and platform visits and remained high even after the course completion, demonstrating sustained student interest [5].

For developing countries, the implementation and use of technological tools is more complicated than in developed countries, since the adoption of virtual learning tools brings new educational and social challenges. The adoption of e-learning tools in developing countries requires specific attention to the social, economic, and technological characteristics of these regions. In higher education institutions in developing countries, continuous training in the use of information and communication technologies (ICTs) is important and this training should be accompanied by strategies to promote the advantages of using virtual learning tools so that users are motivated to be engaged in contexts of flexible and cooperative learning with virtual tools [15].

3.2 Moodle

Moodle, short for Modular Object-Oriented Dynamic Learning Environment [13] [14], is a widely [3] used Course Management System (CMS) designed to create dynamic online learning communities and enhance face-to-face learning experiences [6]. Commonly referred to as a Learning Management System (LMS) or Virtual Learning Environment (VLE) [13], Moodle is adopted globally by universities, organizations [14], and various departments, including Medical Instruction [3], Physics, and foreign languages [6].

As an open-source software platform, Moodle is free to use [14] and highly customizable, supporting thousands of courses and offering extensibility through third-party plugins [3] [13]. One of Moodle's strengths lies in its user-friendly interface, which facilitates intuitive navigation. Content organization is streamlined through the use of modular blocks and structured sections, enhancing the overall learning experience [14]. However, its user-friendliness can be a challenge for some users, particularly those with limited technical knowledge or difficulty understanding course materials [3].

The platform significantly enhances teaching and learning processes, particularly through its robust features for homework submission and evaluation. However, it's important to note that data storage can pose challenges, especially in classes generating large files [6]. Addressing this issue requires careful consideration of storage management strategies to prevent excessive data accumulation over time if the homework feature is implemented.

Despite these challenges, Moodle was chosen due to its modularity, extensibility, and large community of users [12]. It stands out as a viable competitor to major commercial systems due to its extensive feature set and scalability [6]. Additionally, Moodle's cost-effectiveness makes it an attractive option for educational institutions seeking efficient learning solutions [6].

3.2.1 Features

Moodle stands out among learning management systems by offering a comprehensive array of modules designed to enhance the learning experience [3]. These include assignments, quizzes, surveys, chat rooms, and workshops dedicated to course development [3]. Additionally, Moodle boasts a multitude of capabilities such as forums, private journals for student-teacher communication, resources, and assignment display sections [13].

The platform further extends its functionality with support for repositories and portfolios, conditional activities, seamless integration with web services, enhanced file handling, and robust roles and permissions management [5].

As educational technology evolves, Moodle keeps pace with emerging trends. It facilitates personalized learning experiences tailored to individual learning styles and integrates seamlessly with social media platforms. Adaptive mechanisms for accommodating diverse learning styles are also a key feature, along with the growing popularity of blended learning models and the emphasis on learning analytics for informed decision-making [3].

Among its notable features, Moodle offers the convenience of Mail Merge functionality for sending personalized emails [6]. Additionally, courses can be easily packaged into a single zip file using the Backup function, streamlining content management and distribution [13].

3.2.2 Teacher Considerations

Moodle offers a wide range of capabilities tailored to support teachers in delivering effective and engaging online courses. These include features such as forums for fostering interactive discussions, private journals enabling confidential communication between students and teachers, quizzes for assessing student understanding, and access to a variety of educational resources. Additionally, Moodle provides a dedicated section for displaying assignments, and streamlining the management of course materials [13].

One of Moodle's key strengths lies in its modular design, which encompasses themes, activities, interface languages, database schemas, and course formats. This modular approach allows for easy customization and expansion of Moodle's functionality, enabling educators to tailor their online courses to meet specific needs. Furthermore, Moodle's compatibility with other systems facilitates seamless integration with existing educational platforms, enhancing its versatility and adaptability [13].

Teachers utilizing Moodle enjoy comprehensive control over course settings, including the choice of course formats and a flexible array of course activities. With full user logging and tracking capabilities, educators can monitor student progress and engagement effectively, ensuring a personalized learning experience [13]. Moreover, teachers have the freedom to modify courses as needed, including the ability to restrict access for other instructors. The comprehensive suite of course activities available in Moodle, including forums, quizzes, resources, choices, surveys, assignments, chats, and workshops, empowers educators to create dynamic and interactive learning environments [13].

3.2.3 Technical considerations

Moodle can be built on the web application stack of PHP combined with MySQL, making it highly versatile and accessible across various platforms [6] [13] [14].

Written primarily in PHP and utilizing an associated SQL database, Moodle offers compatibility with over ten different database brands, including Oracle, IBM DB2, Microsoft SQL Server, and more [13]. Additionally, Moodle supports the use of external databases for authentication purposes, allowing any database with at least two fields to serve as an authentication source [13].

In terms of authentication and security, Moodle offers a range of mechanisms through plug-in authentication modules, enabling seamless integration with existing systems. Email address verification via confirmation is supported, along with SSL, certificates, and TLS for enhanced security measures [13]. Furthermore, teachers can enhance site security by adding an "enrollment key" to restrict access [13].

For developers seeking to customize Moodle's functionality for specific purposes, the platform offers extensive flexibility and support. Developers can tailor Moodle to meet the unique requirements of their learning environment, ensuring a tailored and optimized user experience [14].

3.2.4 Moodle Plugins

A Moodle plugin is an additional piece of software that extends the functionality of Moodle. Moodle plugins allow users to customize and enhance their Moodle sites by adding new features, activities, blocks, themes, and other components.
Here's a breakdown of some common types of Moodle plugins:

- **Activity Plugins:** These plugins add new interactive activities to Moodle courses, such as quizzes, assignments, workshops, surveys, and more.

- **Resource Plugins:** Resource plugins allow users to incorporate various types of content into their courses, including documents, videos, URLs, and interactive multimedia.

- **Block Plugins:** Blocks are elements displayed in the sidebar or main content area of Moodle courses. Block plugins provide additional functionality, such as calendars, course navigation, activity tracking, and communication tools.

- **Question Type Plugins:** These plugins expand the range of question types available in Moodle quizzes, allowing instructors to create assessments with diverse formats, such as multiple-choice, essay, matching, and interactive drag-and-drop.

- **Theme Plugins:** Theme plugins customize the appearance and layout of Moodle sites, providing options for branding, design, and user interface customization.

- **Authentication Plugins:** Authentication plugins offer alternative methods for user authentication, such as LDAP, OAuth, and single sign-on (SSO), integrating Moodle with external authentication systems.

- **Enrollment Plugins:** Enrollment plugins determine how users are added to courses in Moodle. They can include self-enrollment, manual enrollment by teachers or administrators, enrollment based on user attributes, and more.

- **Report Plugins:** Report plugins generate custom reports and analytics on various aspects of Moodle usage, including user activity, course participation, grades, and completion status.

Moodle's plugin architecture allows for extensive customization and scalability, enabling users to tailor their Moodle sites to meet specific educational needs and requirements. The Moodle plugin directory provides a centralized repository where users can discover, download, and install a wide range of plugins contributed by the Moodle community.

3.2.4.1 Plugins for exercises

There are several Moodle plugins specifically designed to provide exercises and interactive activities for students to complete within Moodle courses. These plugins offer a variety of question types, assessment formats, and interactive features to engage students and enhance their learning experience. Here are some popular
Moodle plugins for exercises and assessments:

- **Quiz Module:** The built-in Quiz module in Moodle allows instructors to create quizzes with various question types, including multiple-choice, true/false, short answer, essay, matching, and more. It supports adaptive quizzes, randomized questions, and immediate feedback for students.

- **H5P Plugin:** The H5P plugin enables the integration of interactive content created with the H5P authoring tool into Moodle courses. H5P supports a wide range of interactive exercises, such as interactive videos, presentations, quizzes, games, and branching scenarios.

- **Lesson Module:** The Lesson module in Moodle provides a structured way to create interactive lessons with branching pathways, multimedia content, and various question types. Instructors can create self-paced tutorials, simulations, or interactive stories to engage students in active learning.

- **Questionnaire Module:** The Questionnaire module allows instructors to create surveys, polls, and questionnaires to gather feedback from students or conduct research within Moodle courses. It supports a variety of question types and customizable survey designs.

- **Virtual Programming Lab (VPL) Plugin:** The VPL plugin allows instructors to set up virtual programming environments within Moodle courses, where students can write and execute code in various programming languages. It provides tools for automatic assessment, code debugging, and real-time feedback on student submissions.

- **Interactive Content Plugins (e.g., Drag-and-Drop, Hotspot, Cloze):** There are various third-party plugins available in the Moodle plugin directory that offer interactive exercises and activities, such as drag-and-drop interactions, hotspot identification, cloze (fill-in-the-blank) questions, and more.

3.2.5 Moodle APIs

Moodle APIs (Application Programming Interfaces) are a set of programming interfaces and protocols that allow developers to interact with Moodle's functionality programmatically. These APIs enable developers to extend, customize, and integrate Moodle with other systems, applications, and services. There can be found a list of APIs in the following link.

Several of Moodle's APIs can help create exercises or interactive learning activities within Moodle courses. Here are some APIs that you might find useful:

- **Activity Modules API:** The Activity Modules API allows developers to create custom activities or exercises that can be added to Moodle courses. This API can be used to develop interactive exercises, quizzes, games, simulations, or any other type of learning activity to include in a course.

- **Question Bank API:** The Question Bank API provides methods for managing questions and question categories within Moodle. This API can be used to create, modify, and organize questions for use in quizzes, assignments, and other assessments. This API can be particularly useful if developing custom question types or in case of need to import questions from external sources.

- **Web Services API:** The Web Services API enables external systems to interact with Moodle remotely over HTTP. This API can be used to create exercises or activities in external applications and integrate them with Moodle courses. For example, developing a custom learning tool or assessment platform that communicates with Moodle via the Web Services API to sync student progress and results.

- **External Tool API:** The External Tool API allows you to integrate external learning tools and applications seamlessly into Moodle courses. This API can be used to develop custom exercises or activities in external tools and embed them within Moodle using LTI (Learning Tools Interoperability) integration. This approach provides flexibility and interoperability for creating interactive learning experiences.

- **Gradebook API:** The Gradebook API provides methods for managing grades and grade-related data within Moodle courses. This API can be used to create custom grading rubrics, calculate grades for exercises or assessments, and automate grading tasks. This API can be helpful if you need to customize the grading process for your exercises.

3.2.6 Customization and Interactivity

Moodle makes use of the Mustache template system to render most of its HTML output, and in some other cases too. Templates are defined as plain text, which typically includes HTML and a range of Mustache tags and placeholders. The Mustache placeholders are replaced with actual values during the render of the page. Mustache templates can be rendered both server-side in PHP and client-side using JavaScript. Themes can override the templates defined in other components if required.

Moodle is a template-based system to which content must be added. This makes Moodle's interface very intuitive and allows for easy navigation. The whole page is presented in a "flat view" format. It is laid out in small blocks and organized around sections following a topic or weekly outline. Each section has its tools such as lessons, quizzes, assignments, and forums which are all linked to a built-in grade book (see section on assessment below). All blocks on a page can be individually arranged, and the elements within each section can be easily moved around or hidden [13].

An activity is a general name for a group of features in a Moodle course. Usually, an activity is something that a student will do that interact with other students and or the teacher. In Moodle terminology, an Activity, such as Forums or Quizzes, properly means something students can contribute to directly, and is often contrasted to a Resource such as a File or Page, which is presented by the teacher to them. However, the term activity is sometimes for convenience and is also used to refer to both Activities and Resources as a group.

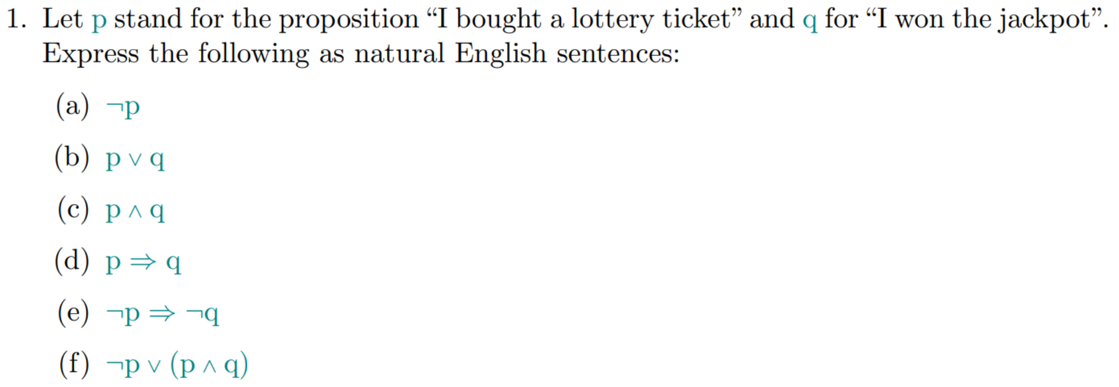
4

Logic Exercises

This chapter provides an exploration of key elements within the realm of logic exercises. It covers a spectrum of topics including propositional logic, Boolean Algebra, truth tables, the Horn algorithm, and first-order logic, and presents potential exercises for future implementation.

4.1 Propositional logic

Propositional logic, also known as sentential logic, is a branch of formal logic that deals with propositions or statements that can be either true or false. It involves studying the logical relationships between these propositions using logical operators such as AND, OR, NOT, IMPLIES, and IF AND ONLY IF. Propositional logic does not concern itself with the internal structure or meaning of the propositions themselves; instead, it focuses on how these propositions are combined and manipulated to form more complex statements. It serves as the foundation for more advanced branches of logic and is widely used in various fields, including mathematics, computer science, philosophy, and linguistics. We can see an example in (figure X).



1. Let p stand for the proposition “I bought a lottery ticket” and q for “I won the jackpot”. Express the following as natural English sentences:
   (a) $\neg p$
   (b) $p \lor q$
   (c) $p \land q$
   (d) $p \Rightarrow q$
   (e) $\neg p \Rightarrow \neg q$
   (f) $\neg p \lor (p \land q)$

Figure 4.1: Propositional Logic exercise

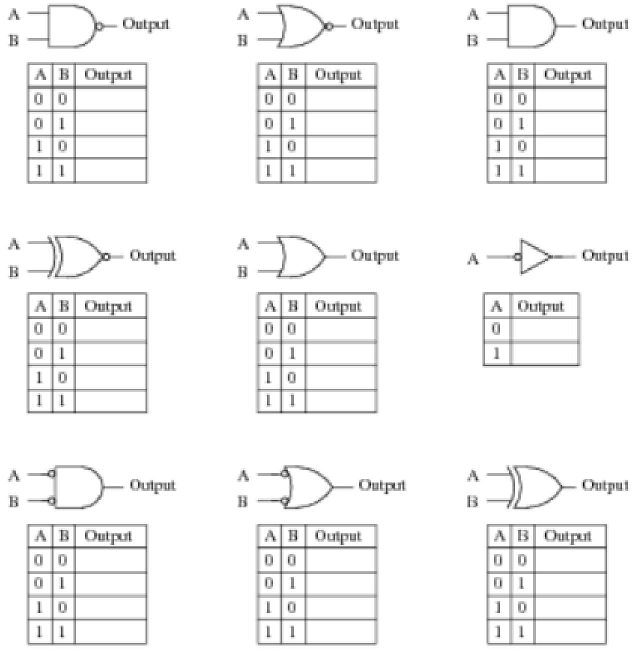
In previous work, there was developed a custom logic tool without connection with

Moodle that implemented exercises related to propositional logic. These exercises include tasks such as filling in blank truth tables and transforming natural language sentences into logical expressions [2]. Therefore our primary exercise to be implemented will be related to this exercise.

4.2 Boolean Algebra

Boolean Algebra, named after the mathematician George Boole, is a branch of mathematics and a formal system used to analyze and manipulate logical expressions. It deals with variables that can only take two values, typically denoted as 0 (false) and 1 (true).In Boolean Algebra, operations such as AND, OR, and NOT are defined to operate on these binary variables. Some aditional operations can also be found such as XOR (exclusive OR), NAND (NOT AND), NOR (NOT OR), XNOR (exclusive NOR)

Boolean Algebra finds wide applications in digital logic design, computer science, circuit design, and database querying, among other areas. It provides a formal and systematic way to represent and analyze logical expressions, making it a fundamental tool in various fields of mathematics and engineering.



A	B	Output
0	0	
0	1	
1	0	
1	1	

A	B	Output
0	0	
0	1	
1	0	
1	1	

A	B	Output
0	0	
0	1	
1	0	
1	1	

A	B	Output
0	0	
0	1	
1	0	
1	1	

A	B	Output
0	0	
0	1	
1	0	
1	1	

A	Output
0	
1	

A	B	Output
0	0	
0	1	
1	0	
1	1	

A	B	Output
0	0	
0	1	
1	0	
1	1	

A	B	Output
0	0	
0	1	
1	0	
1	1	

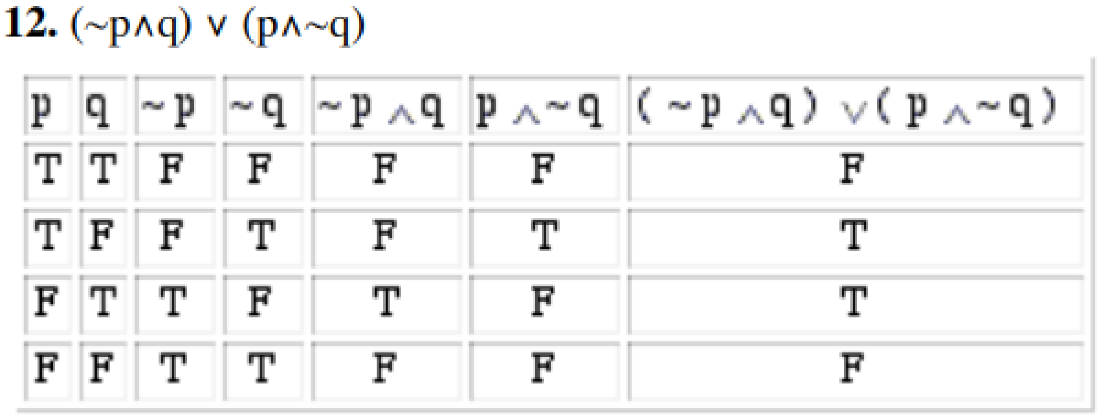
Figure 4.2: Boolean Algebra - Logic Gates exercise

4.3 Truth tables

Truth tables are a systematic way of representing the truth values of logical expressions. They list all possible combinations of truth values for the variables in an expression and determine the resulting truth value of the expression for each combination.

In a truth table:

- Each column represents a variable or a combination of variables.
- Each row represents a unique combination of truth values for the variables.
- The final column represents the truth value of the logical expression based on the truth values of the variables in that row.
- Truth tables are particularly useful for evaluating complex logical expressions, determining the validity of arguments, and simplifying Boolean expressions. They provide a clear and organized method for analyzing the behavior of logical operations and expressions.



12. (~p∧q) ∨ (p∧~q)

p	q	~p	~q	~p ∧q	p ∧~q	(~p ∧q) ∨(p ∧~q)
T	T	F	F	F	F	F
T	F	F	T	F	T	T
F	T	T	F	T	F	T
F	F	T	T	F	F	F

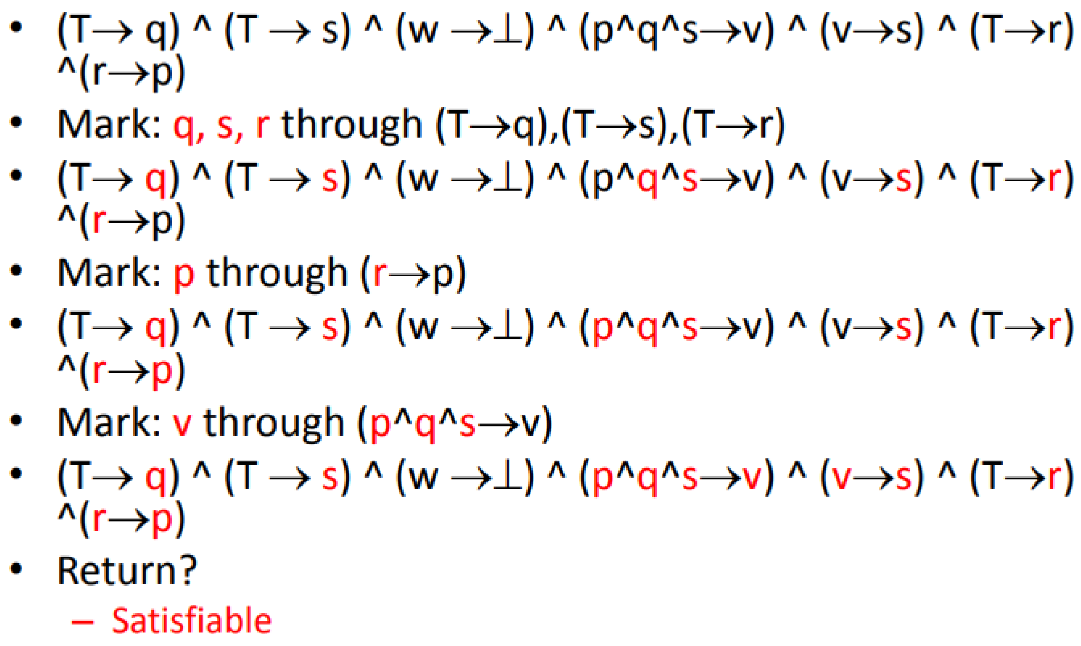
Figure 4.3: Truth tables exercise

4.4 Horn algorithm

A Horn clause is a logical formula containing at most one positive literal (an atom that is asserted to be true) on its left-hand side. A clause with at most one positive literal is also called a definite clause. For example:

The Horn-SAT problem is the problem of determining whether a given set of Horn clauses is satisfiable, i.e., whether there exists an assignment of truth values to the variables that makes all the clauses true simultaneously.

The Horn algorithm is a specific algorithm used to solve the Horn-SAT problem. It exploits the special structure of Horn clauses to efficiently find a satisfying assignment if one exists. The basic idea is to perform a process called Horn clause resolution, which is similar to the more general resolution method used in automated theorem proving.



- (T→ q) ^ (T → s) ^ (w →⊥) ^ (p^q^s→v) ^ (v→s) ^ (T→r) ^(r→p)
- Mark: q, s, r through (T→q),(T→s),(T→r)
- (T→ q) ^ (T → s) ^ (w →⊥) ^ (p^q^s→v) ^ (v→s) ^ (T→r) ^(r→p)
- Mark: p through (r→p)
- (T→ q) ^ (T → s) ^ (w →⊥) ^ (p^q^s→v) ^ (v→s) ^ (T→r) ^(r→p)
- Mark: v through (p^q^s→v)
- (T→ q) ^ (T → s) ^ (w →⊥) ^ (p^q^s→v) ^ (v→s) ^ (T→r) ^(r→p)
- Return?
  - Satisfiable

Figure 4.4: Horn alghoritm exercise

4.5 First-order logic

First-order logic (FOL) is a formal system for expressing statements about objects and their properties, relationships, and quantified conditions. It extends propositional logic by introducing variables, quantifiers, and predicates representing properties or relations between objects. FOL is widely used in mathematics, computer science, philosophy, and artificial intelligence for reasoning and formalizing knowledge.

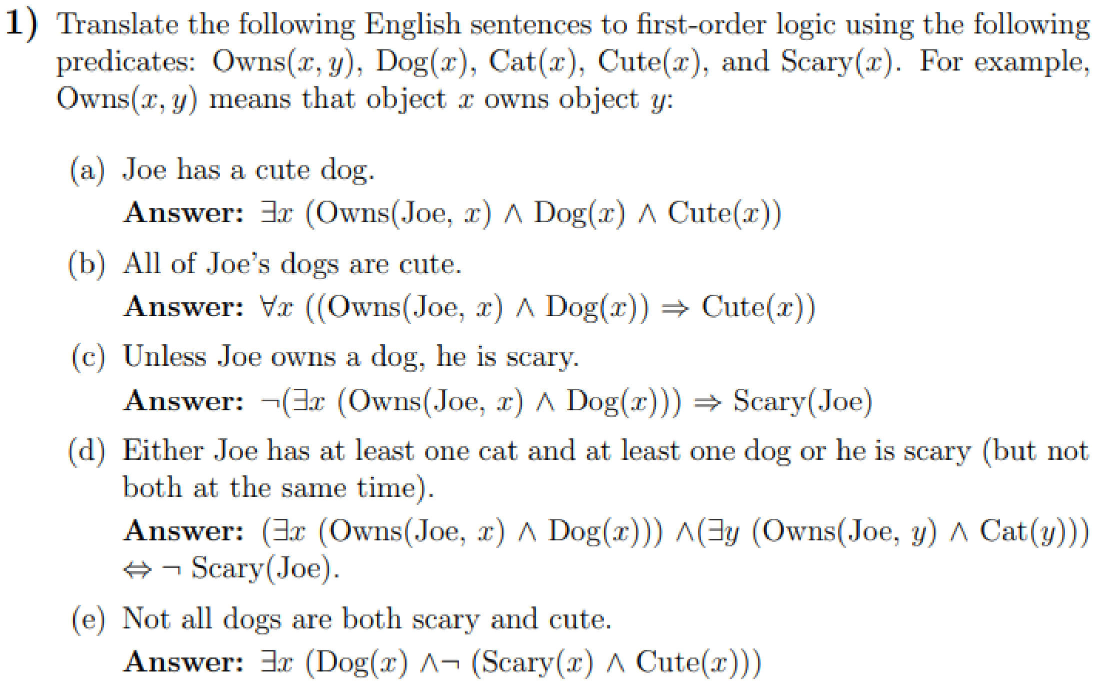
4.6 Other exercises

In addition to the exercises already implemented, there are several other potential logic exercises that could be considered for implementation in our tool. These exercises include:

1. **Normal Form of Prenex:** Practice converting logical formulas into prenex normal form, which is a standard form in predicate logic that allows for clearer quantifier placement.

2. **Normal Form of Skolem and Unification:** Explore Skolem normal form and unification algorithms, essential concepts in mathematical logic and automated theorem proving.

3. **Prolog:** Introduce Prolog, a logic programming language commonly used in artificial intelligence and computational linguistics, allowing students to practice writing and executing logic-based programs.



1) Translate the following English sentences to first-order logic using the following predicates: Owns(x, y), Dog(x), Cat(x), Cute(x), and Scary(x). For example, Owns(x, y) means that object x owns object y:

(a) Joe has a cute dog.
Answer: $\exists x$ (Owns(Joe, x) $\wedge$ Dog(x) $\wedge$ Cute(x))

(b) All of Joe's dogs are cute.
Answer: $\forall x$ ((Owns(Joe, x) $\wedge$ Dog(x)) $\Rightarrow$ Cute(x))

(c) Unless Joe owns a dog, he is scary.
Answer: $\neg$($\exists x$ (Owns(Joe, x) $\wedge$ Dog(x))) $\Rightarrow$ Scary(Joe)

(d) Either Joe has at least one cat and at least one dog or he is scary (but not both at the same time).
Answer: ($\exists x$ (Owns(Joe, x) $\wedge$ Dog(x))) $\wedge$($\exists y$ (Owns(Joe, y) $\wedge$ Cat(y))) $\Leftrightarrow$ $\neg$ Scary(Joe).

(e) Not all dogs are both scary and cute.
Answer: $\exists x$ (Dog(x) $\wedge\neg$ (Scary(x) $\wedge$ Cute(x)))

Figure 4.5: First order logic example

4. **Semantic Consequence:** Dive into the concept of semantic consequence, where one sentence logically follows from another, providing exercises to assess students' understanding and application of logical entailment.

5. **Logic Equivalence:** Offer exercises on determining logical equivalence between logical expressions, helping students develop their skills in recognizing equivalent forms of logical statements.

6. **Normal Forms:** Cover various normal forms in logic, such as conjunctive normal form (CNF) and disjunctive normal form (DNF), providing exercises to transform logical formulas into these standard forms.

7. **Deduction:** Deduction in logic exercises refers to the process of logical reasoning used to reach a conclusion or inference from given premises or information.

By incorporating these additional exercises into our tool, we can provide students with a comprehensive learning experience in logic, covering a wide range of topics and enhancing their logical reasoning skills.

5

Exploring Enhancements in Online Learning Environments

As the landscape of education evolves, the quest for optimizing online learning environments continues to drive research and innovation. Within this pursuit, several areas emerge as potential avenues for exploration and improvement. This section delves into extra dimensions beyond the primary focus of research, aiming to shed light on topics such as gamification, user interface (UI) impact, information extraction, and the crucial role of feedback in online learning platforms. Through an examination of relevant papers and methodologies employed, we unravel the intricacies of these areas and their significance in enhancing the efficacy and user experience of e-learning tools.

5.1 Gamification

Gamification is the process of incorporating game-like elements, such as points, badges, and leaderboards, into non-game contexts to increase engagement, motivation, and participation. It leverages principles of game design to make tasks more enjoyable and to encourage desired behaviors. Gamification is widely used in various fields, including education, marketing, fitness, and workplace training, to enhance user experience and drive desired outcomes.

In [9] the study provides insights into the complex reactions of students to gamification elements, particularly badges, in an online learning context. The paper investigates the introduction of badges as a gamification element in an online learning environment, specifically focusing on a course with automatically assessed exercises. The study incorporates student surveys, submission data, and log records to assess the impact of badges on student engagement, behavior, and learning outcomes. Regarding the students, the majority of them expressed positive sentiments about the visual appeal of badges, but opinions on the completion logic were varied, approximately 82While the numerical data didn't show a significant detrimental impact on learning outcomes, the written feedback reflected diverse student perceptions of badges. The study recommends several considerations for educators implementing badges, including providing opt-out options, incorporating social elements, ensuring novelty, clarifying grading impact, and avoiding hindrances to platform usability. These findings offer insights into the nuanced responses of students to gamification elements, particularly badges, in an online learning context and provide valuable recommendations about the challenges associated with implementing badges in educational settings both subjects to take in consideration to improve the learning and teaching experience of a tool.

5.2 Impact of UI

A user interface (UI) serves as the point of interaction between a user and digital devices or software applications, encompassing visual elements like screens, pages, icons, and interactive aspects such as buttons, forms, and menus. The primary goal of a well-designed UI is to facilitate effective communication and interaction, enabling users to navigate, control, and comprehend the system or application's functionality. By providing an intuitive and visually appealing environment, a well-crafted UI enhances the overall user experience.

In [8], four common design issues are identified: system response time, help facilities, error information handling, and command labeling. The paper also discusses three golden rules for UI design, emphasizing placing the user in control, reducing user memory load, and maintaining interface consistency. The paper highlights how important it is for students to be actively involved and proactive in their education. It specifically focuses on addressing challenges in the design of e-learning platforms, like how to make learning materials interesting and motivating. The suggested strategies propose a teamwork approach between software and students. This involves using speech interfaces, using casual communication styles, including animated teaching assistants, using different colors, giving students more control, and even adding background music. All these strategies aim to make learning more interesting and motivating for students.

5.3 Extracting Information

In this section, we delve into the theme of extracting valuable information from student interactions within e-learning environments, with a specific focus on two noteworthy papers Paper 4 and Paper 3. These papers explore the application of data mining techniques to analyze student responses, revealing insights that are beneficial for both educators and learners. [1] introduces a model that utilizes classification and association algorithms to scrutinize student mistakes, demonstrating the potential of hierarchical classification in clustering students with similar errors. Additionally, it addresses the necessity for enhanced dendrogram visualization and proposes automation for extracting and displaying relevant patterns to educators. [10] introduces the Logic-ITA, a web-based Intelligent Teaching Assistant tailored for formal proofs in propositional logic, showcasing its impact on student performance through personalized learning experiences. Both papers employ data mining approaches, such as SQL queries and association mining, to extract meaningful patterns and relationships from student mistakes. These findings have implications for enhancing online learning tools and providing valuable insights for educators in e-learning environments.

Methodologies Employed

- Data Mining Algorithms: Provide a detailed overview of the data mining algorithms used, including classification and association algorithms.
- Logic Tutor Integration: Explain how the proposed model was applied to the Logic Tutor, detailing the process of analyzing student responses from both web-based tools and traditional exercises.
- Hierarchical Classification Success: Elaborate on the success of hierarchical classification in grouping students with similar mistakes, and discuss its significance.
- Introduction to Logic-ITA: Detail the design and functionalities of Logic-ITA as a web-based Intelligent Teaching Assistant.
- Database Utilization: Explain how Logic-ITA collects and stores student answers in a database, emphasizing its role in extracting pedagogically relevant information.

Findings and Results [1] Dendrogram Visualization: Discuss the authors' insights into the need for improved dendrogram visualization and their suggestions for algorithmic development in this regard. Automation: Explore the authors' intentions to automate the process of extracting information, showcasing the potential benefits of displaying only relevant patterns to teachers.

[10] Implementation at the University of Sydney: Share the positive outcomes observed in student performance at the University of Sydney, including increased homework and exam scores compared to a control group. Data Mining Approaches in Action: Explain how SQL queries and association mining were employed to analyze student mistakes, providing concrete examples of their impact on understanding class progress and patterns in errors.

Teaching Implications and Relevance SQL Queries and Descriptive Statistics: Discuss how SQL queries offer descriptive statistics on common mistakes and misused rules, aiding teachers in identifying areas of difficulty. Association Mining Insights: Explore how association mining reveals relationships between mistakes, providing valuable insights into students' struggles with specific concepts. Overall Impact on Learning: Summarize the positive outcomes and impact on learning observed in both papers, emphasizing their relevance to the broader research goal of improving educational tools' effectiveness.

Key Concepts and Their Significance Logic-ITA: Reinforce the significance of Logic-ITA as a web-based Intelligent Teaching Assistant for formal proofs. SQL Queries and Association

Mining: Reiterate the importance of these data mining techniques in extracting meaningful insights from student interactions. Impact on Learning: Emphasize the observed positive outcomes in student performance and how these findings contribute to the overarching goal of improving educational tools.

5.4 Feedback

In this section, we delve into the critical aspect of feedback within the realm of online learning, focusing on the role of constructive feedback in enhancing students' proficiency. Through the examination of relevant literature, we explore how feedback-oriented online exercises contribute to the development of well-organized answers, particularly in the context of physiology education.

he [7] investigates how online exercises can enhance students' skills in constructing thoughtful responses. These exercises, which incorporate multiple-choice questions with feedback, aim to develop higher-order cognitive skills among students.

Assessing the impact of these exercises reveals a promising correlation between student engagement and enhanced performance, especially in short-answer questions related to the endocrine system. However, challenges exist in implementing such exercises, including technical issues and varying perceptions of difficulty among students.

To address these challenges, several mitigation strategies can be employed:

- Acknowledge and provide clear instructions for technical issues.
- Ensure technical support is readily available.
- Implement adaptive learning strategies to personalize the learning experience.
- Clearly communicate the purpose and relevance of online exercises.
- Incorporate feedback mechanisms to provide immediate guidance.
- Offer a variety of exercise formats to accommodate diverse learning preferences.
- Encourage peer collaboration to enhance motivation and support.
- Design exercises with a gradual progression of difficulty.
- Provide additional resources for students who may find certain concepts challenging.
- Solicit feedback from students to continuously improve the exercises and strike an optimal balance.

6 Development Plan

In this chapter, we outline the plan for designing and implementing our online platform for logic education using Moodle. A detailed plan and Gantt chart illustrate the project timeline and milestones, providing a roadmap for future development.

6.1 Plan

Our future work delves into the implementation of a web-based tool leveraging the E-learning features through Moodle, with immediate feedback mechanisms, and course management in a single environment.

To achieve our goal it's fundamental to create our tool structure using Moodle instance in our server and connecting a database, the database planned to use is MySql being the most common database found for connection with Moodle but in the case of adaptation, Moodle offers compatibility with other databases. The connection to a database will ensure data persistence and data for future analyses.

After setting up our project our goal it's to personalize it. We have a large array of features and options for personalization, therefore depending on the goal we can use APIs, and existing Plugins, create personalized plugins, etc.

Moodle APIs such as Gradebook API with methods for managing grades and grade-related data and Activity Modules API that allows developers to create custom activities or exercises that can be added to Moodle courses and Plugins such as Block for placement, Question Type for exercises, Theme Plugins for appearance purposes, authentication plugins for integrating Moodle with external authentication systems to have students and teachers authentications, Enrollment for setting up the enrollment method, Resource to incorporate content and Activity plugins for interactive activities such as assignments could be valuable.

There are some elements from existing tools that will be taken into consideration relative to the exercise structure such as the exercise structure, the feedback methods, and some extra features. The exercise structure chosen was the modular structure, where the exercises are composed of tasks each with its feedback generators this will improve the complexity. Some extra features are for example the use of built-in connectives and quantifiers that remove deviation and simplify the data processing and exercises with different difficulty levels, the latter more optional.

Our first exercise to be implemented will be related to propositional logic exercises developed in a previous work [2]. This exercise will suffer an adaptation to the chosen exercise structure and potentially to the chosen feedback structure.

Testing is an important part of developing a software system, therefore, our goal is to keep testing while we progress. Taking that into consideration, after a successful implementation of Moodle connection and personalization, one exercise integration, and Feedback there will be executed more exhaustive testing in the whole system.

One crucial part of our platform is the ability to have teachers manage their students and course through evaluations and exercises and for our students to have access to a centralized online learning environment, to achieve that we need to have at least two roles: Teacher and Student, for a better security we will also have Admin role. The users will have to authenticate and have different use cases according to their role, therefore our tool will have authentication developed.

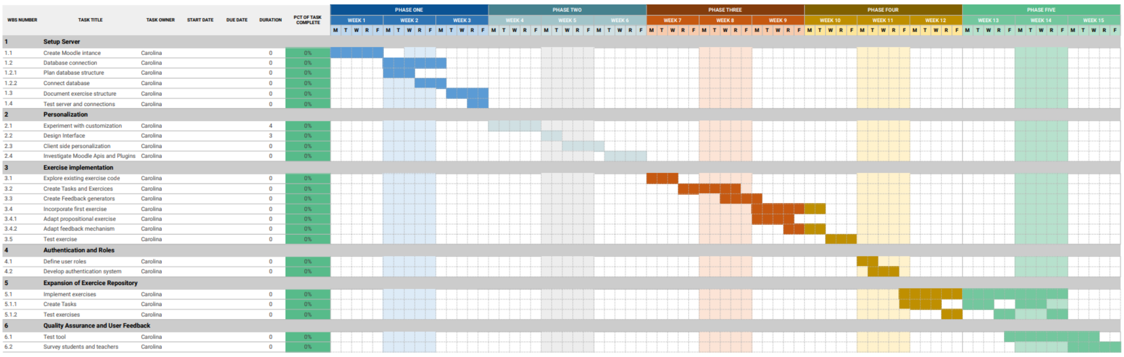
We concluded that User Interface and User Experience in online learning tools are important when maximizing the student engagement and learning process. Thus, to fulfill our goal it's crucial to take into consideration the interface and user experience, this includes feedback, appropriate UI, and additional features for better navigation such as explanations, and error messages.

To have a tool with an elevated scope of logic exercises in one place we will be implementing more exercises, depending on the exercises that may be needed for the usage of Moodle features and potentially our own Moodle plugin.

It's important to have a functional platform and to ensure the correct function there will be tests to identify problems, which will be categorized by urgency and be resolved accordingly. To test in a more realistic environment we will test our platform with both students and teachers and survey their opinion.

6.2 Gantt

In this section we have a separation of tasks in a Gantt Chart.



WBS NUMBER	TASK TITLE	TASK OWNER	START DATE	DUE DATE	DURATION	PCT OF TASK COMPLETE
1	**Setup Server**					
1.1	Create Moodle intance	Carolina			0	0%
1.2	Database connection	Carolina			0	0%
1.2.1	Plan database structure	Carolina			0	0%
1.2.2	Connect database	Carolina			0	0%
1.3	Document exercise structure	Carolina			0	0%
1.4	Test server and connections	Carolina			0	0%
2	**Personalization**					
2.1	Experiment with customization	Carolina			4	0%
2.2	Design Interface	Carolina			3	0%
2.3	Client side personalization	Carolina			0	0%
2.4	Investigate Moodle Apis and Plugins	Carolina			0	0%
3	**Exercise implementation**					
3.1	Explore existing exercise code	Carolina			0	0%
3.2	Create Tasks and Exercices	Carolina			0	0%
3.3	Create Feedback generators	Carolina			0	0%
3.4	Incorporate first exercise	Carolina			0	0%
3.4.1	Adapt propositional exercise	Carolina			0	0%
3.4.2	Adapt feedback mechanism	Carolina			0	0%
3.5	Test exercise	Carolina			0	0%
4	**Authentication and Roles**					
4.1	Define user roles	Carolina			0	0%
4.2	Develop authentication system	Carolina			0	0%
5	**Expansion of Exercice Repository**					
5.1	Implement exercises	Carolina			0	0%
5.1.1	Create Tasks	Carolina			0	0%
5.1.2	Test exercises	Carolina			0	0%
6	**Quality Assurance and User Feedback**					
6.1	Test tool	Carolina			0	0%
6.2	Survey students and teachers	Carolina			0	0%

Figure 6.1: Gantt Chart

Bibliography

[1] B. Aguado, A. Merceron, and A. Voisard. “Extracting information from structured exercises”. In: *Proceedings of the 4th International Conference on Information Technology Based Higher Education and Training ITHET03, Marrakech, Morocco*. 2003 (cit. on pp. 23, 24).

[2] B. M. AMARAL. “ONLINE INTERACTIVE TOOL FOR LEARNING LOGIC”. In: (2022) (cit. on pp. 18, 27).

[3] H. Athaya et al. “Moodle Implementation for E-Learning: A Systematic Review”. In: *Proceedings of the 6th International Conference on Sustainable Information Engineering and Technology*. 2021, pp. 106–112 (cit. on pp. 11, 12).

[4] W. Auccahuasi et al. “Interactive online tool as an instrument for learning mathematics through programming techniques, aimed at high school students”. In: *Proceedings of the 6th International conference on information technology: IoT and Smart City*. 2018, pp. 70–76 (cit. on pp. 8, 9).

[5] D. Benta, G. Bologa, and I. Dzitac. “E-learning platforms in higher education. Case study”. In: *Procedia Computer Science* 31 (2014), pp. 1170–1176 (cit. on pp. 11, 12).

[6] D. Benta et al. “University level learning and teaching via e-learning platforms”. In: *Procedia Computer Science* 55 (2015), pp. 1366–1373 (cit. on pp. 11–13).

[7] J. Carnegie. “Use of feedback-oriented online exercises to help physiology students construct well-organized answers to short-answer questions”. In: *CBE—Life Sciences Education* 14.3 (2015), ar25 (cit. on p. 25).

[8] B. Faghih, D. M. R. Azadehfar, P. Katebi, et al. “User interface design for e-learning software”. In: *arXiv preprint arXiv:1401.6365* (2014) (cit. on pp. 1, 23).

[9] L. Haaranen et al. “How (not) to introduce badges to online exercises”. In: *Proceedings of the 45th ACM technical symposium on Computer science education*. 2014, pp. 33–38 (cit. on p. 22).

[10] H. Hoppe et al. "A web-based tutoring tool with mining facilities to improve learning and teaching". In: *Artificial intelligence in education: Shaping the future of learning through intelligent technologies* 97.201 (2003), p. 49 (cit. on pp. 23, 24).

[11] V. Karavirta et al. "Interactive exercises for teaching logic circuits". In: *Proceedings of the 2016 ACM Conference on Innovation and Technology in Computer Science Education*. 2016, pp. 101–105 (cit. on pp. 4, 6, 7).

[12] C. Limongelli, F. Sciarrone, and G. Vaste. "Personalized e-learning in Moodle: the Moodle_LS System". In: *Journal of e-Learning and Knowledge Society* 7.1 (2011), pp. 49–58 (cit. on p. 11).

[13] V. Nedeva. "The possibilities of e-learning, based on Moodle software platform". In: *Trakia Journal of Sciences* 3.7 (2005), pp. 12–19 (cit. on pp. 11–13, 16).

[14] S. Saraswat. "Customization and implementation of LMS moodle". In: *International Journal of Scientific and Research Publications* 4.5 (2014), pp. 1–4 (cit. on pp. 11, 13).

[15] A. Valencia-Arias, S. Chalela-Naffah, and J. Bermúdez-Hernández. "A proposed model of e-learning tools acceptance among university students in developing countries". In: *Education and Information Technologies* 24 (2019), pp. 1057–1071 (cit. on p. 11).