Towards Explainable AI: Reasoning With Controlled Natural Language

Sofia Gavasi and Nico Roos

Maastricht University
Department of Advanced Computing Sciences
Maastricht, The Netherlands
s.gavasi@student.maastrichtuniversity.nl
roos@maastrichtuniversity.nl

Abstract. This paper investigates how formal logical reasoning can be made accessible and explainable through natural language. It introduces a system that integrates the semantic tableau method with a Controlled Natural Language, enabling inference over structured English inputs while preserving logical rigor. The system performs syntactic and semantic preprocessing, classifies sentence structures, and applies adapted tableau rules to construct visual proof trees. It supports extended reasoning patterns, including the handling of explicit exceptions. Experimental evaluation on logic tasks, synthetic datasets, and user studies demonstrates that the system can reason accurately within its constraints and improve user comprehension and trust through transparent explanations. These results highlight the potential of logic-based, linguistically grounded systems for advancing explainable AI.

Keywords: Explainable AI · Reasoning · Controlled Natural Language · Natural Language Inference · Semantic Tableau.

1 Introduction

As artificial intelligence systems become increasingly embedded in decision-making processes across a range of domains, there is a growing demand for transparency in how these systems reach their conclusions. Many existing approaches, particularly those based on statistical learning and black-box architectures [1], are capable of generating answers, but provide limited insight into the reasoning behind them. This raises concerns about accountability, interpretability, and user trust. While post-hoc explanation techniques have been developed to mitigate these issues, they often fail to reveal the actual inferential structure behind a model's decision [2].

In contrast, symbolic reasoning systems offer a fundamentally different approach. These systems derive conclusions through the application of formal logic to explicitly stated premises, enabling complete traceability of the reasoning process [3]. This method is, in principle, inherently interpretable, due to the fact that each inference step can be examined and justified. However, symbolic systems are not without limitations. Their explanatory outputs are often expressed

in formal logic notations, which, although precise, may be inaccessible to end users unfamiliar with symbolic representations. As a result, the challenge is not only to enable logical reasoning, but to do so in a manner that is both rigorous and communicatively transparent.

Early question-answering systems addressed this by incorporating natural language reasoning through logic-based representations. A classic example is Chat-80 [4], a Prolog-based system from the 1980s that answered English questions about geography . It would parse a natural language query into a Prolog logical form, then use logical inference to derive an answer. Although efficient, the reasoning took place in the domain of formal logic and was completely hidden from the user.

Recent advances in natural language processing have moved toward neural approaches, especially large-scale transformer-based language models [5]. These models have achieved impressive results on tasks like natural language inference and question answering. However, as surveys such as *Natural Language Reasoning: A Survey* [5] emphasize, their reasoning processes remain partially unreliable, meaning that they can arrive at correct conclusions in some cases while failing unpredictably in others due to gaps in consistency, robustness, or logical validity. These systems often generate fluent, seemingly logical responses, but small changes in input can lead to inconsistent behavior, exposing how the models rely on statistical pattern matching rather than true logical competence. As these studies suggest, the next phase of progress requires moving beyond raw task performance and toward systems that support transparent, consistent, and trustworthy reasoning.

Motivation: This paper aims to enable reasoning directly in natural language, so that users unfamiliar with formal logic can still understand and trust the system's conclusions. The central objective is not merely to process language, but to support *explainable decision making* by producing human-readable justifications that mirror how people naturally reason. The motivation is to bridge the gap between rigorous logical inference and human-friendly communication. This approach aligns with the broader goals of explainable AI: building systems that can show they work in a way users find intuitive.

A key challenge in reasoning over unrestricted natural language is ambiguity, which computers struggle to resolve without contextual guidance. To mitigate this, the system employs a Controlled Natural Language (CNL) [6] that balances expressiveness with structural clarity, ensuring each input has a single, unambiguous interpretation. Users are guided to phrase inputs in this structured form, which are then processed. The resulting representations are evaluated using adapted semantic tableau rules [7] along with a user interface that visualizes the reasoning process to enhance interpretability.

Paper Outline: The next section provides the background information on the semantic tableau method and on Controlled Natural Language (CNL). Section 3 presents our system design. Section 4 describes the experimental evaluation of the system developed. Finally, Section 5 concludes the paper.

2 Preliminaries

2.1 The Semantic Tableau Method

The semantic tableau method [8] is a proof procedure used to determine the validity or satisfiability of logical formulas by systematically attempting to construct a counter-model. That is, it tries to construct a model for a set of formulas and the negation of a desired conclusion by breaking down complex formulas into simpler components based on syntactic rules and arranges them into a branching, tree-like structure called the tableau. Every branch of the tableau is a possible model. If no model can be found, i.e., every branch leads to a contradiction, the conclusion holds. A branch that leads to a contradiction is said to be closed, and open otherwise. This method offers a visually intuitive way to explore logical consequence and is foundational in both manual and automated reasoning systems [9].

The semantic tableau method was selected as the foundation for this system because of its structural clarity and suitability for explainable automated reasoning. In contrast to resolution-based or sequent calculus approaches, tableau decompose logical formulas in a step-by-step, tree-like fashion that reflects the intuitive process of identifying contradictions or validating arguments [9]. Cognitive science research, notably the *Mental Model Theory of Reasoning* proposed by Johnson-Laird [10] and related studies [11], suggests that such hierarchical structures closely correspond to the way humans engage in deductive reasoning by iteratively breaking complex problems into simpler components. This alignment with human cognitive patterns makes semantic tableau particularly well-suited for applications seeking to bridge formal logic with user-transparent reasoning processes.

2.2 Reasoning With Natural Language

Reasoning in natural language has historically been studied through syllogistic logic, originating with Aristotle's *Prior Analytics* (circa 350 BCE) [12], where structured reasoning involving quantifiers like *"All"*, *"Some"*, and*"None"* was first formalized . While powerful, syllogisms are limited to categorical statements and lack the expressiveness of modern logic.

In contemporary AI and computational linguistics, Natural Language Inference (NLI) or Recognizing Textual Entailment (RTE) tasks aim to determine whether a given hypothesis follows from a premise in natural language [13]. These tasks are central to benchmarks such as SNLI [14] and have been used to evaluate transformer-based language models.

One method to formally bridge logic and natural language is through the use of Lambda Logical Forms (LLFs), where natural language is translated into λ-calculus expressions capturing scope and structure [15]. Additionally, Combinatory Categorial Grammar (CCG) provides a syntax-semantics interface that helps derive logical form directly from linguistic constituents [16].

2.3 Controlled Natural Language

Controlled Natural Language (CNL) provides a compromise between the expressive richness of natural language and the precision of formal logic. It defines a constrained subset of natural language with unambiguous syntax and semantics, suitable for computational interpretation and formal reasoning tasks [17].

Notably, CNLs such as Attempto Controlled English (ACE) [18] have been applied successfully in fields including knowledge representation, requirements engineering, and legal reasoning, where both interpretability and formal rigor are essential. [19] In addition to these research-oriented applications, controlled natural languages have also been adopted in governmental and regulatory contexts. For example, the Dutch Tax and Customs Administration employs a controlled language known as Regelspraak [20] to reformulate tax legislation into a formally precise yet readable format that can be directly executed by computer systems to determine income tax obligations.

The system developed for this work accepts user input in a grammar that was defined using Backus–Naur Form (BNF) [21]. Rather than explicitly defining a fixed vocabulary of allowed words, the grammar specifies the syntactic structures of permissible statements.

3 System Design

To understand the functioning of the system, it is useful to begin with a concrete example of an inference. Consider the following set of premises:

- *"All fruits that are unripe are not tasty."*
- *"If a fruit is not tasty then Lucy is not satisfied."*
- *"Some apples are unripe and all apples are fruit."*

From these premises, the conclusion we aim to derive is that *"Lucy is not satisfied."*

To derive the conclusion, the first essential step involves comprehending the structure and meaning of each sentence. This foundational process is referred to as *Text Processing*. *Text Processing* constitutes the core of the system, as it enables it to determine which rule of the semantic tableau method should be applied to a given sentence. Furthermore, it facilitates the derivation of new sentences based on existing ones and ensures that this procedure is repeated iteratively until a sentence can no longer be transformed.

The second foundational component of the system is the actual *Solver*, which constitutes the main logical flow of the inference process. The *Solver* organizes the premises and the conclusion into a tree-like structure analogous to the semantic tableau. It then iteratively applies logical rules, creating new branches when required, and systematically checks for contradictions.

The procedure terminates under one of two conditions: either all branches are closed, indicating that each branch contains a contradiction, or no further modifications can be applied to the sentences, in which case some branches remain open.

3.1 Text Processing

The text analysis component of the system is built upon spaCy [22], an open-source library for advanced NLP in Python. By leveraging spaCy's capabilities, the system is able to effectively process and interpret natural language inputs, enabling the transformation of complex sentences into structured representations amenable to formal logical reasoning. To illustrate this, consider the first premise:

"All fruits that are unripe are not tasty."

Using spaCy, the system parses this sentence to extract key linguistic features. For instance:

- **Tokenization:** The sentence is divided into the following tokens: *All*, *fruits*, *that*, *are*, *unripe*, *are*, *not*, *tasty.*
- **Part-of-Speech Tagging:** *All* – determiner (DET), *fruits* – noun (NOUN), *that* – relative pronoun (PRON), *are* – auxiliary verb (AUX), *unripe* – adjective (ADJ), *are* – verb (AUX), *not* – negation particle (PART), *tasty* – adjective (ADJ).
- **Dependency Parsing:** The main clause is structured around the copular verb "are" connecting the subject "All fruits that are unripe" to the complement "not tasty". The relative clause "that are unripe" modifies the noun "fruits".

This syntactic analysis allows the system to recognize the sentence as a universal negative statement. The relative clause is identified and linked to the head noun, enabling a precise interpretation of the logical structure underlying the sentence.

To enable the analysis and categorization of different sentence types, the *Text Processing* module follows a structured sequence of steps:

1. **Preprocessing:** This stage involves several linguistic adjustments to enhance parsing accuracy, including spelling normalization, grammatical corrections (such as verb form adjustments), pronoun substitution, compound noun merging, and phrase reformatting.
2. **Sentence Segmentation:** The text is split into individual sentences based on periods (“."), so that each clause can be analyzed independently.
3. **Detection of Logical Operators:** Phrases such as *"It is not the case that"*, *"It is not true that"*, and *"if and only if"* are identified and labeled accordingly.
4. **Transformation of Relative Clauses:** Sentences of the form *"All _ who/that _ _ "* (e.g., *"All fruits that are unripe are not tasty"*) are particularly complex to process. These are restructured into conditional forms such as: *"If one is a fruit and one is unripe, then one is not tasty"*.
5. **Logical Connective Identification:** Conjunctions, disjunctions, and conditional statements (e.g., “and", “or", “if-then") are detected and annotated to segment and categorize different logical clauses. The sentence *"If a fruit*

is not tasty then Lucy is not satisfied.", will be split into *"A fruit is not tasty"* and *"Lucy is not satisfied"*, and they will be labeled as connected by a conditional statement

6. **Subject/Object Disaggregation:** In cases where a clause contains multiple subjects or direct objects (e.g., *"Lucy and Mary are happy"*) the sentence is separated into atomic components: *"Lucy is happy"* and *"Mary is happy"*.
7. **Atomic Clause Analysis:** In the final step, each basic clause is analyzed in detail:
   - Quantifiers are identified and assigned to subjects or direct objects (e.g., *"**Some** apples are unripe"*),
   - The positions of subjects, verbs, and (if applicable) direct objects or predicate adjectives are determined,
   - Negations are detected (e.g., *"Lucy is **not** satisfied"*).

3.2 Solver

Following the text processing phase, the system proceeds to logical reasoning through a dedicated component, referred to as the *Solver*. The name reflects its functional role: much like formal logic solvers, it takes a set of premises and a proposed conclusion and systematically determines whether the conclusion logically follows. It acts as a reasoning engine that evaluates the satisfiability and consistency of the input statements.

The reasoning process is organized around a tree-based structure, where each node contains one or more logical statements. These statements are expanded iteratively according to the rules of the semantic tableau method, creating branches that represent alternative logical possibilities.

Each node in the tree represents a logical state in the reasoning process and contains a set of true statements and a set of false statements. The statements in the false part of a node can be interpreted as if they were prefixed with *"It is not the case that"*.

When the solver is invoked with an inference, the premises are added to the set of true statements, while the conclusion is introduced as a false statement. This configuration enables the system to test whether the conclusion necessarily follows from the premises by attempting to close all branches of the tableau. If all branches lead to contradictions, the conclusion is logically valid.

Application of Rules To carry out the inference process, the solver relies on a set of logical rules derived from the classical semantic tableau method. These rules determine how complex logical formulas are decomposed into simpler components, which are then systematically analyzed to construct or close branches in the tableau.

While the underlying structure of the semantic tableau rules follows classical logic, their application to natural language inputs required several modifications. In formal logic, operations such as negation can be applied directly using symbolic notation (e.g., $\neg P$). However, natural language expressions demand more

advanced handling, as there is no uniform way to express negation across different sentence types.

For instance, removing the negation of the statement *"Nobody is happy"* makes it *"Somebody is happy"*, which requires identifying the quantifier and transforming it into an existential form. However, removing the negation of a simple declarative statement like *"I am not happy"* involves deleting a negation token, resulting in *"I am happy"*. These transformations are not only syntactically distinct but also context-dependent, and thus require linguistic analysis beyond the scope of symbolic logic alone.

To address this, the system relies on the *Text Processing* module to classify sentence structures and determine the appropriate form of negation or logical transformation. This enables the solver to apply tableau rules accurately, while preserving the grammatical and semantic integrity of natural language inputs.

Another example of a difference from the standard semantic tableau rules is that, in the case of conjunctions and disjunctions, it is necessary to distinguish between two forms: those that occur within sentences or clauses, and those that occur within terms. In natural language, an example of the former would be *"Some apples are unripe* ***and/or*** *all apples are fruit"*, whereas an example of the latter would be *"Apples* ***and/or*** *pears are unripe"*, or *"I eat apples* ***and/or*** *pears"*.

Statement Processing in Each Iteration At each iteration, the solver processes statements from unprocessed leaf nodes in the tableau. For each statement in the sets of true and false formulas, the following steps are executed:

1. **Linguistic Analysis:** The statement is analyzed using the *Text Processing* component to extract its logical structure and identify relevant features.
2. **Rule Application:** Based on the classification assigned during text processing, the appropriate semantic tableau rule is applied. For instance, the sentence *"If a fruit is not tasty then Lucy is not satisfied"* is identified as a conditional statement. Consequently, the rule for implications is applied, resulting in the derivation of two statements: *"A fruit is not tasty"* and *"Lucy is not satisfied."* These are then placed in two separate child nodes, reflecting the branching nature of the implication.
3. **Branch Evaluation:** Each newly created child node is examined for contradictions. A branch is considered closed if it contains a contradiction (i.e., a statement and its negation). If no further rules can be applied and no contradictions are found, the branch remains open.

To manage complexity and prevent excessive branching, the solver applies only one rule per iteration. This controlled approach ensures a systematic exploration of the logical space and facilitates efficient inference.

Quantifier Handling The handling of universal and existential quantifiers represents one of the most complex aspects of the solver.

Existential statements require the introduction of new constants into the logical framework. Examples of such statements include *"There is a person"*, *"I love someone"*, or *"Some apples are unripe"*. These statements assert the existence of at least one individual satisfying a given property, necessitating the use of new constants during logical expansion.

Moreover, the introduction of constants is also necessary when dealing with negated universal statements. For instance, the sentence *"It is not the case that all humans are happy"* appears in the false set of a tableau node as *"All humans are happy"*. In this context, a new constant (e.g., *"Alice"*) must be introduced to instantiate the universal quantifier, resulting in the sentence *"(It is not the case that) Alice is happy"*.

Each introduced constant is added to a list of used values, which will then be applied to all universal statements encountered throughout the inference process. Constants already present in the initial input, such as *"Lucy"* in the original inference example, are also included in this list.

Universal statements are more challenging to handle, as they require reapplication each time a new constant is introduced. For example, consider the statement *"All apples are fruit"*. This must be instantiated for every constant in the system, including newly introduced ones. If the constant *"Alice"* is introduced, the universal statement yields the implication *"If Alice is an apple, then Alice is a fruit"*. The general form *"If one is an apple, then one is a fruit"* is preserved and reapplied whenever a new constant is added.

To manage this process, the solver uses a dictionary that tracks all introduced constants, the universal statements they have been applied to, and the specific node in the inference tree where each application took place, crucial for maintaining correct behavior across branches.

In summary, when a universal statement is encountered at a given node, the solver checks whether it has been applied to all relevant constants in the node's ancestry. If not, the statement must be reapplied for the missing constants to ensure logical completeness.

3.3 Interface

The system features a Streamlit-based interface designed for transparency and explainability. An introductory section explains the system and includes optional details on semantic tableau. A grammar guide helps users write valid, unambiguous CNL inputs.

Users can either select from a set of predefined inference tasks or input their own custom premises and conclusions. Once an inference is submitted, the system generates a logic tree using semantic tableau rules and visualizes each reasoning step. Every node in the tree includes a structured explanation: it lists the current true and false statements, specifies the logical rule applied, highlights the sentence being modified, and shows the resulting transformations. Contradictions and open branches are explained with textual justifications.

4 Experiments

4.1 Validity Experiments

Reasoning with the Syllogism Dataset from Kaggle

Introduction and Method To evaluate the reasoning capabilities of the system on structured inference tasks, an experiment was conducted using an external dataset of syllogisms. The dataset, sourced from Kaggle [23], consists of 55 pairs of premises followed by two proposed conclusions, along with a classification label indicating which conclusions logically follow. The purpose of this experiment was to assess whether the solver could accurately reason through syllogistic structures expressed in controlled natural language, demonstrating its ability to handle standardized inference patterns.

Minimal transformation of the dataset[1] was required, and in most cases, the statements did not require adjustments. However, because the system does not possess background world-knowledge, occasional supplementation of the premises was necessary. For instance, the inference with premises *"Some dogs are brown. All brown animals are common"*, and conclusion *"Some dogs are common"*, could not be solved without knowing that *"All dogs are animals"*. For such cases, this knowledge was added to the original inference as a premise.

Results When the necessary background information was made explicit through the added statements, the system demonstrated strong performance in matching the labeled outcomes, and all inferences were solved correctly.

Discussion This experiment demonstrates that the system is capable of reasoning over structured syllogistic patterns expressed in controlled natural language. The relatively standardized structure of the syllogism dataset made it well-suited to the system's current design, requiring only minimal intervention to adapt the premises for processing.

The necessity of explicitly stating background knowledge highlights a key characteristic of the system: it operates exclusively on the information presented within the premises, without inferring unstated facts. This transparency is an advantage in terms of explainability but also suggests an area for future improvement, such as integrating controlled domain knowledge to support more natural reasoning.

Reasoning with Synthetic Datasets

Introduction and Method To further assess the flexibility and robustness of the system, custom datasets were generated using large language models (LLMs) and a controlled grammar specification. A formal grammar was defined using Backus-Naur Form (BNF).[2]. A graphical specification of the grammar, outlining

[1] The adapted Kaggle dataset can be found at: https://doi.org/10.34894/0ANEPG

[2] The formal BNF grammar specification and LLM generated datasets can be found at: https://doi.org/10.34894/0ANEPG

the permitted structures for input sentences, can be found in Appendix A. The goal of this experiment was to investigate how the system would perform when reasoning over automatically generated natural language inferences that adhered strictly to the grammar rules.

The grammar was provided to two LLMs, GPT-4o[24] and DeepSeek[25], along with a few examples illustrating the desired structure of the premises and conclusions. Each model was tasked with generating a dataset[2] of 100 inference examples: 50 classified as simpler inferences and 50 classified as more complex. Each example consisted of a set of premises, a proposed conclusion, and a validity label indicating whether the conclusion logically followed from the premises.

While the general structure of the generated datasets conformed to the BNF grammar, some corrections were necessary before evaluation. Minor issues, such as proper nouns not being capitalized, occasionally caused problems in text processing and had to be manually corrected to ensure accurate parsing by the system. No alterations were made to the logical structure of the examples.

Results After minor corrections to address capitalization inconsistencies, the system successfully evaluated all inferences from the generated datasets. The solver correctly classified each conclusion according to its labeled validity.

Discussion This experiment highlights the solver's ability to reason accurately over natural language statements generated under a controlled syntactic framework. The use of a formal grammar enabled the creation of a large and diverse set of test cases without manual sentence crafting, showcasing the potential for scalable dataset generation.

Observations regarding the generated datasets reveal interesting differences between the two LLMs. The DeepSeek-generated dataset was simple and repetitive, while ChatGPT's dataset had more complex logic but included nonsensical yet grammatically correct sentences. Despite lacking real-world meaning, these sentences were still useful for testing logical validity.

Overall, this experiment suggests that with a well-defined grammar and minimal preprocessing, the solver can reliably handle automatically generated reasoning tasks, further validating its potential for controlled natural language applications.

4.2 Application Experiments

Reasoning with Logic Puzzles

Introduction and Method To explore the potential applications of the reasoning system beyond standard inference validation, an experiment was conducted using a selection of logic puzzles. These puzzles were sourced from publicly available educational and recreational resources [26, 27, 28, 29, 30]. The goal of this experiment was to investigate whether the solver could be adapted to tackle structured reasoning challenges found in logic puzzles, demonstrating its flexibility and applicability in a broader range of natural language reasoning tasks.

It is important to note that logic puzzles represent a category of reasoning problems that are often more complex than those typically encountered in practical decision-making scenarios.

Before the puzzles could be processed by the solver, some transformation of the original statements was required. Many puzzles employed complex quantifications, such as *"exactly two"* or *"at least one"*, which are difficult to handle within the current semantic tableau framework. Puzzles relying heavily on quantities were therefore excluded from the experiment. For the selected puzzles, premises were carefully reformulated to fit the controlled natural language format expected by the solver. This included eliminating ambiguities, explicitly stating subjects, and simplifying compound sentences while preserving the logical intent of the original puzzle. Each adapted puzzle was paired with a set of premises and three proposed conclusions: one that was logically valid and two that were invalid. The resulting dataset[3]. had 27 inferences. The objective was to assess whether the solver could correctly identify the valid conclusion and reject the invalid ones.

Results The system successfully processed all adapted logic puzzles, correctly distinguishing between valid and invalid conclusions in each case. In every scenario, the solver determined validity correctly without error, demonstrating that with appropriate sentence transformations, it is capable of solving logic puzzles formulated in controlled natural language.

Discussion This experiment demonstrates that the reasoning system can be applied beyond controlled inference datasets, extending to practical reasoning tasks such as solving classic logic puzzles. Although significant manual adjustments to the input sentences were necessary, the core logical structure of the puzzles could be preserved, and the solver was able to reason over them successfully.

However, the experiment also highlighted certain limitations. A substantial number of logic puzzles had to be excluded because they relied on complex quantified reasoning, which the current implementation does not support. Additionally, the adaptation process itself required careful reformulation of sentences to fit the controlled input requirements, suggesting that a fully automated pipeline for broader natural language applications would require further developments, such as quantity-handling and more advanced paraphrasing capabilities.

Despite these challenges, the successful application to logic puzzles highlights the flexibility and adaptability of the system. It illustrates that the solver is not limited to narrowly defined inference tasks but can be extended to solve a variety of structured reasoning problems, provided that inputs are appropriately controlled.

User Testing

Introduction and Method To evaluate the overall accessibility of the reasoning system, a user study was conducted using a Google Form-based questionnaire

[3] The logic puzzles dataset can be found at: https://doi.org/10.34894/0ANEPG

and a user interface.[4] Following initial feedback from a small pilot group, the revised form was distributed via WhatsApp group chats.

The questionnaire begins with a brief instructional section that introduces participants to the system interface and the core concepts of semantic tableau used in the reasoning process. This is followed by a guided example that walks the user through a single inference scenario using the solver, accompanied by natural language explanations at each node of the inference tree. The user is then able to test out the system on 15 examples of inferences[5]. Subsequently, participants are asked to compare the reasoning process of the system on a specific inference to a parallel explanation generated by GPT-4o, which was prompted to solve the same inference in a step-by-step manner. The objective of this comparison is not to evaluate performance between the two systems, but rather to investigate how users perceive and understand natural language explanations as opposed to more formal, symbolic ones, such as the one generated by the LLM.

Throughout the form, participants respond to questions designed to measure their understanding of the reasoning process, their ability to recognize logical contradictions, and their perception of the system's trustworthiness and transparency.

The study pursues two central objectives: (1) to determine whether the system's interface and explanation style support transparency and user understanding and (2) to compare user comprehension of a natural language-based reasoning explanation versus a symbolic logic-based one, particularly contrasting responses from participants with and without formal training. Statistical analyses were conducted in Python with SciPy[22] and Matplotlib[31].

Results A total of 72 responses were collected through the user study of users with and without a background in logic. User comprehension of the system was evaluated following a guided example in which participants interacted with the solver. In response to the question *"How much were you able to understand and follow along the process of the reasoning?"*, ranked on a scale from 1 to 5, the average comprehension scores are shown in Table 1.

Table 1. Comprehension scores after using the solver (scale 1–5)

Group	**N**	**Mean Score**	**Standard Deviation**
Overall	72	4.54	0.60
No logic background	33	4.39	0.66
With logic background	39	4.67	0.53

Participants were also asked whether *"seeing each step of the reasoning process made the system feel more trustworthy or transparent"*. All participants responded affirmatively to this question.

[4] The interface is found at: https://thesis-awxqzd5xxn7z6jwj7mhgzk.streamlit.app/
[5] The 15 inference examples can be found at: https://doi.org/10.34894/0ANEPG

Comprehension was also rated after viewing a symbolic explanation. The results are summarized in Table 2.

Table 2. Comprehension scores: Formal Language vs. Solver (scale 1–5)

Group	**Formal Mean**	**Formal SD**	**Solver Mean**	**Solver SD**
Overall	3.56	0.96	4.46	0.69
No logic background	3.30	0.98	4.30	0.77
With logic background	3.77	0.90	4.59	0.59

To assess the difference in perceived comprehension between the symbolic and natural language explanations, a Wilcoxon signed-rank test was conducted. This non-parametric test was chosen because the data consisted of paired, ordinal responses on a 1–5 scale, and did not follow a normal distribution. Normality was assessed using the Shapiro-Wilk test, which is well-suited for small to moderate sample sizes and indicated that the assumption of normality was violated. The test results, shown in Table 3, revealed a highly significant difference in favor of the solver.

Table 3. Paired sample tests comparing solver and symbolic logic comprehension

Test	**Statistic**	p**-value**
Wilcoxon signed-rank	$W = 28.0000$	< 0.000000005

Additional comparisons were conducted between participants with and without a background in logic. To compare comprehension scores across independent groups, Mann-Whitney U tests were used. These tests are appropriate for comparing non-normally distributed ordinal data across two independent samples. The results for each explanation type are shown in Table 4.

Table 4. Between-group comparisons (logic background vs. none)

Explanation Type	**Test**	**Statistic**	p**-value**
Formal Language	Mann-Whitney U	$U = 813.0000$	0.0451
Solver	Mann-Whitney U	$U = 769.0000$	0.1100

Finally, participants were asked: *"Do you think explaining each step in natural language, instead of using symbolic logic, makes it more accessible and trustworthy?"*. Table 5 shows that most participants answered "Yes," especially those

without a logic background, while those with a logic background were somewhat more split but still mainly positive.

Table 5. Perceived accessibility of explanations (logic background vs. none)

Response	No logic background	With logic background
Partially	4	17
Yes	29	22

A chi-square test of independence was conducted to assess whether participants' background in logic was associated with their responses. The results are summarized in Table 6.

Table 6. Chi-square test for association between background and natural language preference

Statistic	Degrees of Freedom	p-value
$\chi^2 = 7.1124$	1	0.0077

Discussion The findings from the user study support several observations regarding the accessibility of the system. Participants generally reported a high level of comprehension after using the solver, with an average score of 4.54 out of 5. This suggests that the step-by-step natural language explanations were broadly accessible, even to users without formal training in logic, whose average comprehension score (4.39) remained relatively close to that of trained participants (4.67). The unanimous agreement that the step-by-step structure enhanced transparency further reinforces the system's potential as a trustworthy interface for reasoning tasks.

The comparison between the solver and GPT-generated symbolic explanations provides additional insight. Users consistently rated the natural language explanation as more understandable than the symbolic logic one, with an average difference of nearly one full point (4.46 vs. 3.56). To evaluate whether this difference was statistically significant, a Wilcoxon signed-rank test was conducted. The results confirmed a highly significant difference in favor of the solver's natural language explanations. This suggests that such an explanation strategy may better support user comprehension than formal symbolic logic, particularly in general or mixed-expertise audiences.

Interestingly, while the between-group comparison showed a statistically significant difference in comprehension for GPT's symbolic explanations, favoring participants with a background in logic, this was not the case for the solver-generated explanations. The absence of a significant difference in solver com-

prehension between trained and untrained users indicates that the system may bridge the accessibility gap typically posed by logic-heavy interfaces.

Furthermore, all participants expressed a favorable view of the natural language explanation format, with 64% selecting *"Yes"* and the remaining 36% selecting *"Partially"*, and no participant responded negatively to this approach. The chi-square test revealed a statistically significant association between users' background in logic and their expressed preference: participants without formal training were more likely to respond with a clear *"Yes"*, while those with a background in logic more frequently selected *"Partially"*.

Despite the overall positive results, it is important to recognize limitations that constrain the generalizability of these findings. The sample comprised 72 participants, most of whom likely share a similar demographic profile, namely, university affiliation and a baseline level of academic literacy. This relative homogeneity may have contributed to the high comprehension scores and favorable perceptions of the natural language explanations. Furthermore, as participation was voluntary and not externally verified, there was no mechanism to ensure the absence of response bias or self-selection effects. These factors suggest that the observed outcomes, while promising, may not fully capture the experience of more diverse populations.

5 Conclusion

This paper set out to explore how logical reasoning can be made more explainable by operating directly in natural language. This work demonstrates that formal reasoning can be effectively aligned with natural language to support transparent and user-friendly inference. By adapting the semantic tableau method for controlled language input, the system enables explainable reasoning without sacrificing logical precision. The results of the user study indicate that transparency is important to support both understanding and trust. Even users without formal training in logic reported a high level of comprehension. This contributes to advancing explainable AI by showing how symbolic methods can be made accessible and interpretable through language-based interaction.

In addition to explainability, the system is capable of addressing core Natural Language Inference (NLI) tasks such as entailment, contradiction, and neutrality, provided the statements comply with the Controlled Natural Language. The NLI-tasks belong to the standard capabilities of semantic tableau methods.

One of the main limitations of the system is its dependence on controlled natural language. While this design choice was implemented to allow unambiguous interpretation, it also restricts the system's applicability to real-world language, which is often informal and context-dependent.

Future work could focus on expanding the expressive power of the controlled natural language, such as complex forms of quantification. Another important direction is the integration of background knowledge using external ontologies or knowledge graphs, enabling the system to draw on implicit facts during the inference process, instead of having to explicitly specify them in the input.

References

[1] Vikas Hassija et al. "Interpreting Black-Box Models: A Review on Explainable Artificial Intelligence". In: *Cognitive Computation* 16.1 (2024), pp. 45–74. DOI: 10.1007/s12559-023-10179-8.

[2] Daniel Vale, Ali El-Sharif, and Muhammed Ali. "Explainable artificial intelligence (XAI) post-hoc explainability methods: risks and limitations in non-discrimination law". In: *AI and Ethics* 2.4 (Nov. 2022), pp. 815–826. DOI: 10.1007/s43681-022-00142-y.

[3] Uzma Nawaz, Mufti Anees-ur-Rahaman, and Zubair Saeed. "A review of Neuro-Symbolic AI integrating reasoning and learning for advanced cognitive systems". In: *Intelligent Systems with Applications* (2025), p. 200541. ISSN: 2667-3053. DOI: 10.1016/j.iswa.2025.200541.

[4] David H. D. Warren and Fernando C. N. Pereira. "An efficient easily adaptable system for interpreting natural language queries". In: *Comput. Linguist.* 8.3–4 (July 1982), pp. 110–122. ISSN: 0891-2017.

[5] Fei Yu et al. "Natural Language Reasoning, A Survey". In: *ACM Comput. Surv.* 56.12 (Oct. 2024). ISSN: 0360-0300. DOI: 10.1145/3664194.

[6] Adam Wyner et al. "On Controlled Natural Languages: Properties and Prospects". In: June 2010, pp. 281–289. ISBN: 978-3-642-14417-2. DOI: 10.1007/978-3-642-14418-9_17.

[7] Lex Hendriks and A.O. Kazakci. "A method for design reasoning using logic: From Semantic Tableaux to Design Tableaux". In: *ICED 11 - 18th International Conference on Engineering Design - Impacting Society Through Engineering Design* 2 (Jan. 2011), pp. 275–286.

[8] Evert W. Beth. *The Foundations of Mathematics: A Study in the Philosophy of Science*. Amsterdam: North-Holland, 1959.

[9] Marcello D'Agostino et al., eds. *Handbook of Tableau Methods*. Springer, 1999. ISBN: 978-94-017-1754-0.

[10] P. N. Johnson-Laird. *Mental Models: Towards a Cognitive Science of Language, Inference, and Consciousness*. Cambridge, MA: Harvard University Press, 1983. ISBN: 978-0-674-56882-2.

[11] David Copeland. "Theories of categorical reasoning and extended syllogisms". In: *Thinking and Reasoning* 12 (Nov. 2006), pp. 379–412. DOI: 10.1080/13546780500384772.

[12] Aristotle. *Prior Analytics*. Trans. by Robin Smith. Indianapolis, IN: Hackett Publishing Company, 1989. ISBN: 978-0-87220-064-7.

[13] Ido Dagan et al. "Recognizing Textual Entailment: Models and Applications". In: *Synthesis Lectures on Human Language Technologies* 6.4 (2013), pp. 1–220. DOI: 10.2200/S00509ED1V01Y201305HLT023.

[14] Samuel R. Bowman et al. "A large annotated corpus for learning natural language inference". In: *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. Ed. by Lluís Màrquez, Chris Callison-Burch, and Jian Su. Lisbon, Portugal: Association for Computational Linguistics, 2015, pp. 632–642. DOI: 10.18653/v1/D15-1075.

[15] Patrick Blackburn and Johan Bos. *Representation and Inference for Natural Language: A First Course in Computational Semantics*. Stanford, CA: CSLI Publications, 2005. ISBN: 978-1-57586-496-9.

[16] Mark Steedman. *The Syntactic Process*. Cambridge, MA: MIT Press, 2000. ISBN: 978-0-262-19420-4.

[17] Tobias Kuhn. "A Survey and Classification of Controlled Natural Languages". In: *Computational Linguistics* 40.1 (Mar. 2014), pp. 121–170. DOI: 10.1162/COLI_a_00168.

[18] Norbert E. Fuchs. "First-Order Reasoning for Attempto Controlled English". In: *Proceedings of the Second International Workshop on Controlled Natural Language (CNL 2010)*. Ed. by Michael Rosner and Norbert E. Fuchs. Vol. 7175. Lecture Notes in Computer Science. Berlin / Heidelberg, Germany: Springer, 2012, pp. 73–94. ISBN: 978-3-642-31174-1.

[19] Norbert E. Fuchs et al. "Attempto Controlled English: A Knowledge Representation Language Readable by Humans and Machines". In: *Reasoning Web: First International Summer School 2005, Msida, Malta, July 25-29, 2005, Revised Lectures*. Ed. by Norbert Eisinger and Jan Małuszyński. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005, pp. 213–250. ISBN: 978-3-540-31675-6. DOI: 10.1007/11526988_6. URL: https://doi.org/10.1007/11526988_6.

[20] Mischa Corsius et al. "RegelSpraak: a CNL for Executable Tax Rules Specification". In: *Proceedings of the Seventh International Workshop on Controlled Natural Language (CNL 2020/21)*. Ed. by Tobias Kuhn et al. Amsterdam, Netherlands: Special Interest Group on Controlled Natural Language, Sept. 2021. URL: https://aclanthology.org/2021.cnl-1.6/.

[21] Peter Naur. "Report on the Algorithmic Language ALGOL 60". In: *Communications of the ACM* 3.5 (1960), pp. 299–314. DOI: 10.1145/367236.367262.

[22] Pauli Virtanen et al. "SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python". In: *Nature Methods* 17 (2020), pp. 261–272. DOI: 10.1038/s41592-019-0686-2.

[23] warcoder. *Syllogism Data*. https://www.kaggle.com/datasets/warcoder/syllogism-data. 2021.

[24] OpenAI. *ChatGPT-4o*. https://openai.com/index/hello-gpt-4o. 2024.

[25] Daya Guo et al. "DeepSeek-R1: Incentivizing Reasoning Capability in LLMs via Reinforcement Learning". In: *arXiv preprint arXiv:2501.12948* (2025). URL: https://arxiv.org/abs/2501.12948.

[26] Centre for Innovation in Mathematics Teaching. *Mathematics Enhancement Programme: Year 7, Unit 1 – Introduction*. 2000.

[27] Denise Gaskins. *Lewis Carroll's Logic Challenges*. 2010. URL: https://denisegaskins.com/2010/10/11/lewis-carroll%E2%80%99s-logic-challenges/.

[28] Presh Talwalkar. *Brazilian Olympiad Pinocchio's Green Hats Viral Question*. 2022. URL: https://mindyourdecisions.com/blog/2022/06/22/brazilian-olympiad-pinocchios-green-hats-viral-question/.

[29] G. N. Hile. *3E Lewis Carroll Puzzles*. 2025. URL: https://math.hawaii.edu/~hile/math100/logice.htm.

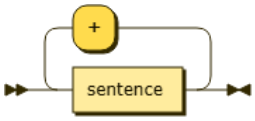
[30] Jay Bennett. *Riddle of the Week #43: Knights and Knaves, Part 1*. 2017. URL: https://www.popularmechanics.com/science/math/a14382121/riddle-of-the-week-43-knights-and-knaves-part-1/.

[31] J. D. Hunter. "Matplotlib: A 2D Graphics Environment". In: *Computing in Science & Engineering* 9.3 (2007), pp. 90–95. DOI: 10.1109/MCSE.2007.55.

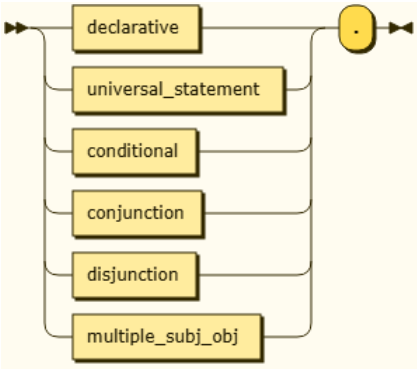
A Graphical Representation of the BNF Grammar

Generated using the Bottlecaps Railroad Diagram Generator: https://www.bottlecaps.de/rr

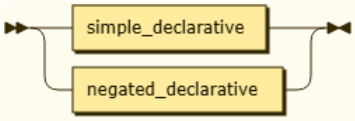
text:



sentence:

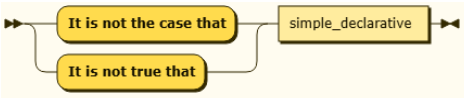


declarative:

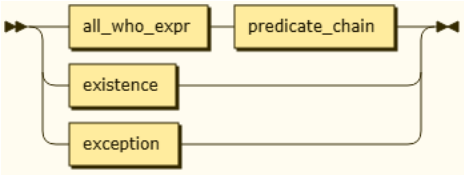


simple_declarative:

negated_declarative:

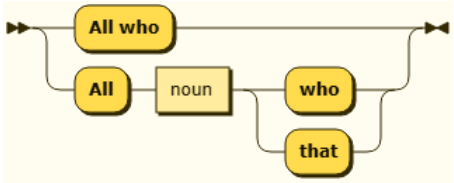


universal_statement:



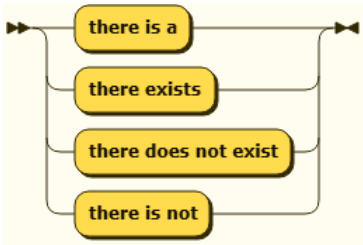
predicate_chain:

all_who_expr:

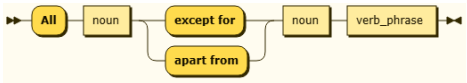


existence:

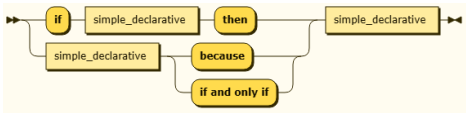
existence_expr:



exception:



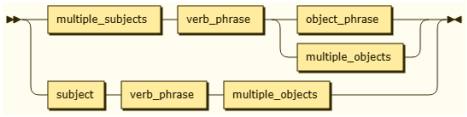
conditional:



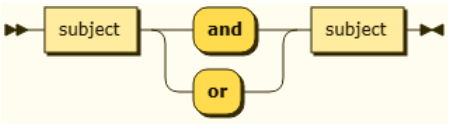
conjunction:

disjunction:

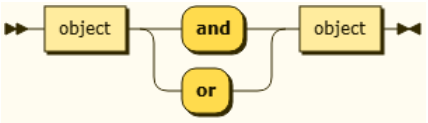
multiple_subj_obj:



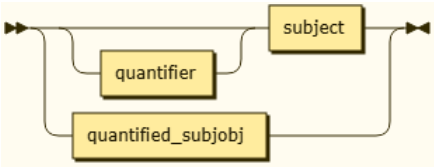
multiple_subjects:



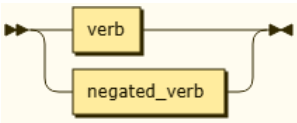
multiple_objects:



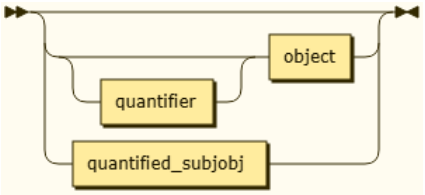
subject_phrase:



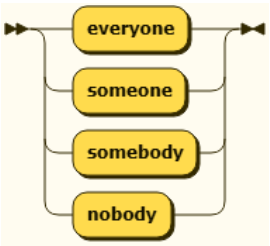
verb_phrase:



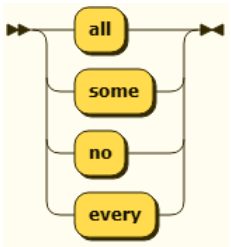
object_phrase:



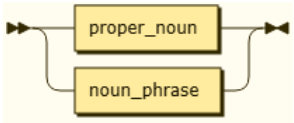
quantified_subjobj:



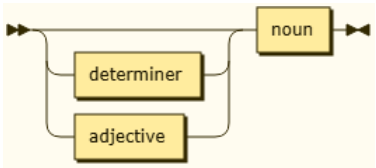
quantifier:



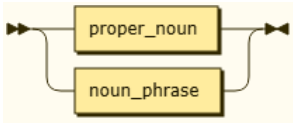
subject:



noun_phrase:



object:



determiner: