

Neuro-Symbolic Reasoning with External Inference Tools Mitigates Bias from Sub-Formula Level Logical Structure

Ryo Hasegawa, Yusuke Sakai, Hidetaka Kamigaito, Taro Watanabe

Nara Institute of Science and Technology (NAIST), Japan

{hasegawa.ryo.hp5, sakai.yusuke.sr9, kamigaito.h, taro}@is.naist.jp

Abstract

Logical reasoning is a core ability of modern language models, and various approaches have been proposed to enhance it. One effective method is neuro-symbolic approach, which makes a model output logic formulas and predict the answer based on them. However, fair comparison is challenging because existing methods use different experimental settings, such as post-processing and prompt styles. In addition, existing error analyses are too coarse to capture the impact of sub-formula level structures. In this paper, we present a systematic analysis of neuro-symbolic approaches under a unified setting with sub-formula level annotation. We reveal that some sub-formula level structures bias the model’s inference toward assuming the answer is true, and neuro-symbolic approaches with an external inference tool help mitigate this bias. On the other hand, these methods sometimes lack flexibility in representing sentence meaning as logic formulas¹.

1 Introduction

Modern large language models (LLMs) have acquired high linguistic abilities (OpenAI et al., 2024; OpenAI, 2025; Comanici et al., 2025; Yang et al., 2025), and logical reasoning is one of the essential abilities of LLMs for a wide range of downstream tasks (Cobbe et al., 2021; Wu et al., 2022; Lin et al., 2022; Zhao et al., 2023). As a result, various efforts have been made to evaluate and improve logical reasoning abilities (Morishita et al., 2024a,b; Liu et al., 2023; Jiang et al., 2025). For example, Chain-of-Thought (Wei et al., 2022; Kojima et al., 2022; Wang et al., 2023) encourages models to produce intermediate reasoning steps. In addition, reasoning models are trained to generate reasoning tokens and have demonstrated stronger reasoning ability (Yang et al., 2025; Guo et al., 2025; OpenAI, 2025).

However, several challenges remain. Models still struggle with complex reasoning (Parmar et al., 2024), compositional reasoning (Dziri et al., 2023), and out-of-domain generalization (Saparov et al., 2023). Moreover, LLMs are black-box systems, limiting their interpretability. Several approaches have been proposed to address these challenges, and one of them is neuro-symbolic approach (Pan et al., 2023; Olausson et al., 2023; Di et al., 2025; Ye et al., 2023). In these methods, LLMs generate logic formulas and inference with an external theorem prover or within the model. Since reasoning is expressed through explicit logic formulas explicitly, these methods can improve interpretability and performance.

Nevertheless, two limitations remain. First, existing analyses remain coarse-grained. Most previous studies evaluate the difficulty of problem solving with formula-level metrics, such as the number of reasoning steps. As a result, it remains unclear how sub-formula level structures, such as the negation status of the decisive atomic formulas, affect LLMs. Second, the mechanisms underlying the effectiveness of neuro-symbolic methods have not been clarified. Although various techniques have been proposed to improve performance, prior studies have adopted different types of post-processing and prompts, making it challenging to conduct systematic comparisons within a unified experimental setting. As a result, the respective effects of generation and inference processes are still not well understood.

Motivated by these limitations, we ask the following two research questions: *How do sub-formula factors affect the outputs of LLMs?* (RQ1). *On which factor does the performance depend: the formula generation process or the inference process?* (RQ2). To answer these questions, we conduct a systematic evaluation of multiple neuro-symbolic approaches with annotation which enable sub-formula level analysis. We newly annotate sub-

¹Our sub-formula level annotation is available at <https://github.com/RyHasegawa/SFAAC>.

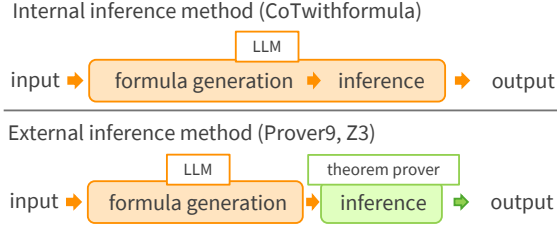


Figure 1: Categorization of neuro-symbolic methods. The inference process is conducted either within the LLM (*internal inference method*) or by an external theorem prover (*external inference method*).

formula factors to a synthetic dataset AAC (Betz et al., 2021), which consists of detailed variants of formulas but no annotation for sub-formula factors. We also use a human-written dataset FOLIO (Han et al., 2024) to compare with AAC. Our experiments reveal that, when not relying on external inference tools, sub-formula factors such as the negation status of the last atomic formulas or the number of atomic formulas in the consequent of the conclusion cause errors in the inference process. We also find neuro-symbolic methods with an external inference tool mitigate these errors. However, on human-written dataset, the generation process of neuro-symbolic methods leads to incorrect outputs because models struggle to map semantically equivalent words to the same predicates or terms.

2 Background and Related Work

Neuro-symbolic Methods. In neuro-symbolic approaches, LLMs are required to generate logic formulas and use them for inference. Figure 1 shows the two types: *internal inference methods*, and *external inference methods*. For **Internal inference methods**, generated formulas are not explicitly returned but used for internal inference within the LLM. One example is SymbCoT (Xu et al., 2024). It is a prompting strategy that makes models to translate natural language to symbolic format, e.g., think step-by-step like CoT, and check both the translation and reasoning chain. For **External inference methods**, they use automated theorem provers, and LLM outputs logic formula code acceptable for the prover. LINC (Olausson et al., 2023) uses Prover9 (McCune, 2005), and LoRP (Di et al., 2025) uses Prolog similarly. Logic-of-Thought (Liu et al., 2025) improves LINC by supplementing logic information. In LogicLM (Pan et al., 2023), various schemas such as Prover9 and Z3 (De Moura and Bjørner, 2008) are

selectively used depending on the task. These existing methods employ different prompting strategies, pre-processing, and post-processing techniques to enhance performance. LINC and LoRP execute multiple inference passes and utilize majority voting. Logic-of-thought involves the rephrase of prompts, while SymbCoT and LOGIC-LM incorporate a feedback loop. Therefore, a fair comparison under standardized conditions by existing methods remains impossible. This motivates us to conduct experiments under a unified prompt, pre-processing, and post-processing for fair analysis of formula generation process and inference process.

Datasets. Various datasets have been developed to evaluate LLMs’ logical reasoning ability. ParaRule (Clark et al., 2020) explicitly specifies the number of reasoning steps, and Proofwriter (Tafjord et al., 2021) expanded ParaRule by adding the reasoning steps generation task. AAC (Betz et al., 2021) organizes logic formulas multi-dimensionally, while LogicBench (Parmar et al., 2024) covers non-monotonic reasoning. PrOntoQA (Saparov and He, 2023) uses fictional knowledge to assess pure reasoning ability, whereas FOLIO (Han et al., 2024) uses human-written natural text. However, these datasets mainly analyze the effects of reasoning steps, formula counts, or model knowledge, and thus do not cover sub-formula factors. Therefore, we augment AAC with explicit annotations of sub-formula factors to enable a rigorous analysis of this factor.

Theorem Provers. The theorem provers in neuro-symbolic methods are commonly employed Prover9 (McCune, 2005) and Z3 (De Moura and Bjørner, 2008). Prover9 is paired with Mace4, and Prover9 attempts to find a proof, while Mace4 attempts to find a disproving counter-example. This refutation proves that Prover9 can provide a complete proof. In contrast, Z3 is a Satisfiability Modulo Theories (SMT), and the reasoning process is heuristic. While Prover9 is more rigorous for first-order logic, Z3 supports a wide range of logical tasks. It is not well understood whether differences between these theorem provers affect the results.

3 Analysis Methods

3.1 Fair Analysis for Sub-Formula Factors

Sub-formula factors can be defined in terms of atomic formulas, or atoms, such as $F(x)$, and compound formulas formed by logical operators, such

formulas	label	formula level factors			sub-formula level factors	
		# formulas	# atoms	# steps	negation of $H(x)$	# siblings of $H(x)$
1. $\forall x (F(x) \wedge I(x)) \rightarrow G(x), G(x) \rightarrow \neg H(x) \Rightarrow \forall x (F(x) \wedge I(x)) \rightarrow H(x)$	false	3	8	2	positive	1
2. $\forall x (F(x) \wedge I(x)) \rightarrow G(x), G(x) \rightarrow H(x) \Rightarrow \forall x (F(x) \wedge I(x)) \rightarrow \neg H(x)$	false	3	8	2	negative	1
3. $\forall x (F(x) \rightarrow G(x)), G(x) \rightarrow H(x) \wedge I(x) \Rightarrow \forall x F(x) \rightarrow (\neg H(x) \wedge I(x))$	false	3	8	2	negative	2

Figure 2: Conditions for a fair analysis of sub-formula factors. In this figure, $H(x)$ is the decisive atom for the false label. The dataset should include variations in sub-formula factors associated with $H(x)$, while the formula-level factors are required to remain comparable.

	generalized modus ponens	generalized contraposition	hypothetical syllogism 1	hypothetical syllogism 2	hypothetical syllogism 3	generalized modus tollens	disjunctive syllogism	generalized dilemma
base schema	$\forall x Fx \rightarrow Gx,$ Fa $\Rightarrow$ Ga	$\forall x Fx \rightarrow \neg Gx$ $\Rightarrow$ $\forall x Gx \rightarrow \neg Fx$	$\forall x Fx \rightarrow Gx,$ $\forall x Gx \rightarrow Hx$ $\Rightarrow$ $\forall x Fx \rightarrow Hx$	$\forall x Fx \rightarrow Gx,$ $\forall x \neg Hx \rightarrow \neg Gx$ $\Rightarrow$ $\forall x Fx \rightarrow Hx$	$\forall x Fx \rightarrow Gx, \exists x$ ($Hx \wedge \neg Gx$) $\Rightarrow$ $\exists x (Hx \wedge \neg Fx)$	$\forall x Fx \rightarrow Gx,$ $\neg Ga$ $\Rightarrow$ $\neg Fa$	$\forall x Fx \rightarrow (Gx \vee Hx),$ $\forall x Fx \rightarrow \neg Gx$ $\Rightarrow$ $\forall x Fx \rightarrow Hx$	$\forall x Fx \rightarrow (Gx \vee Hx),$ $\forall x Gx \rightarrow Jx,$ $\forall x Hx \rightarrow Jx$ $\Rightarrow$ $\forall x Fx \rightarrow Jx$
negation variant	$\forall x \neg Fx \rightarrow Gx,$ $\neg Fa$ $\Rightarrow$ Ga	$\forall x \neg Fx \rightarrow Gx$ $\Rightarrow$ $\forall x Gx \rightarrow Fx$	$\forall x \neg Fx \rightarrow Gx,$ $\forall x Gx \rightarrow Hx$ $\Rightarrow$ $\forall x \neg Fx \rightarrow Hx$	$\forall x Fx \rightarrow Gx,$ $\forall x \neg Hx \rightarrow \neg Gx$ $\Rightarrow$ $\forall x Fx \rightarrow Hx$	$\forall x \neg Fx \rightarrow Gx,$ $\exists x (Hx \wedge \neg Gx)$ $\Rightarrow$ $\exists x (Hx \wedge Fx)$	$\forall x Fx \rightarrow \neg Gx,$ Ga $\Rightarrow$ $\neg Fa$	$\forall x Fx \rightarrow (\neg Gx \vee Hx),$ $\forall x Fx \rightarrow Gx$ $\Rightarrow$ $\forall x Fx \rightarrow Hx$	$\forall x Fx \rightarrow (\neg Gx \vee Hx),$ $\forall x \neg Gx \rightarrow Jx,$ $\forall x \neg Hx \rightarrow Jx$ $\Rightarrow$ $\forall x Fx \rightarrow Jx$
complex predicates	$\forall x (Fx \wedge Hx) \rightarrow Gx,$ $Fa,$ Ha $\Rightarrow$ Ga	$\forall x (Fx \wedge Hx) \rightarrow \neg Gx$ $\Rightarrow$ $\forall x Gx \rightarrow \neg (Fx \wedge Hx)$	$\forall x Fx \rightarrow Gx,$ $\forall x Gx \rightarrow (Hx \vee Ix)$ $\Rightarrow$ $\forall x Fx \rightarrow (Hx \vee Ix)$	$\forall x (Fx \vee Ix) \rightarrow \neg Gx,$ $\forall x \neg Hx \rightarrow \neg Gx$ $\Rightarrow$ $\forall x (Fx \vee Ix) \rightarrow Hx$	$\forall x Fx \rightarrow Gx,$ $\forall x Fx \rightarrow Ix,$ $\exists x Hx \wedge \neg (Gx \wedge Ix)$ $\Rightarrow$ $\exists x (Hx \wedge \neg Fx)$	$\forall x Fx \rightarrow (Gx \wedge Hx),$ $\neg Ga$ $\Rightarrow$ $\neg Fa$	$\forall x (Fx \wedge Ix) \rightarrow (Gx \vee Hx),$ $\forall x Gx \rightarrow \neg (Fx \wedge Ix),$ $\Rightarrow$ $\forall x (Fx \wedge Ix) \rightarrow Hx$	$\forall x (Fx \wedge Ix) \rightarrow (Gx \vee Hx),$ $\forall x Gx \rightarrow Jx,$ $\forall x Hx \rightarrow Jx$ $\Rightarrow$ $\forall x (Fx \wedge Ix) \rightarrow Jx$
de morgan	$\forall x \neg (Fx \vee Hx) \rightarrow Gx,$ $\neg Fa$ $\Rightarrow$ Ga	$\forall x (Fx \wedge Hx) \rightarrow \neg Gx$ $\Rightarrow$ $\forall x Gx \rightarrow \neg (Fx \vee Hx)$	$\forall x \neg Fx \rightarrow Gx,$ $\forall x Gx \rightarrow (\neg Hx \vee \neg Ix)$ $\Rightarrow$ $Gx \rightarrow \neg (Hx \wedge Ix)$	$\forall x (\neg Fx \wedge \neg Ix) \rightarrow \neg Gx,$ $\forall x \neg Hx \rightarrow Gx$ $\Rightarrow$ $\neg (Fx \vee Ix) \rightarrow Hx$	$\forall x Fx \rightarrow Gx,$ $\forall x Fx \rightarrow Ix, \exists x$ $Hx \wedge \neg (Gx \vee Ix)$ $\Rightarrow$ $\exists x Hx \wedge \neg Fx$	$\forall x Fx \rightarrow \neg (Gx \vee Hx),$ Ga $\Rightarrow$ $\neg Fa$	$\forall x (Fx \wedge Ix) \rightarrow (Gx \vee Hx),$ $\forall x Gx \rightarrow \neg (Fx \vee Ix)$ $\Rightarrow$ $\forall x (Fx \wedge Ix) \rightarrow Hx$	$\forall x (\neg Fx \wedge \neg Ix) \rightarrow (Gx \vee Hx),$ $\forall x Gx \rightarrow Jx,$ $\forall x Hx \rightarrow Jx$ $\Rightarrow$ $\forall x \neg Jx \rightarrow (Fx \vee Ix)$
multiple formulas within one group				Gold label	negation of final atom # atoms in conclusion's consequent			
1. $\forall x (Fx \wedge Hx) \rightarrow \neg Gx \Rightarrow \forall x Gx \rightarrow \neg (Fx \wedge Hx)$				True	positive	2	The sub-formula level annotation added in this work	
$\forall x (Fx \wedge Hx) \rightarrow \neg Gx \Rightarrow \forall x Gx \rightarrow \neg (Fx \wedge \neg Hx)$				False	negative	2		
2. $\forall x Fx \rightarrow \neg (Gx \wedge Hx) \Rightarrow \forall x (Gx \wedge Hx) \rightarrow \neg Fx$				True	negative	1		
$\forall x Fx \rightarrow \neg (Gx \wedge Hx) \Rightarrow \forall x (Gx \wedge Hx) \rightarrow Fx$				False	positive	1		

Figure 3: Overview of dataset AAC and our sub-formula level annotation. We annotate the negation status of the final atom and the number of atoms in conclusion's consequent for 1420 samples in AAC. This annotation makes it possible to conduct sub-formula level analysis.

	Overall accuracy	Success rate	Conditional accuracy
Formula	$\frac{\# \text{correct}}{\# \text{all samples}}$	$\frac{\# \text{successfully executed}}{\# \text{all samples}}$	$\frac{\# \text{correct}}{\# \text{successfully executed}}$
Ability (Naive, CoT, CoTwithformula)	Overall	-	Natural language reasoning
Ability (Prover9, Z3)	Overall	Instruction-following	Symbolic logic understanding

Table 1: Definitions of evaluation metrics and the abilities each metric can evaluate.

as $\forall x F(x) \rightarrow G(x)$. Since it is impossible to analyze all possible sub-formula factors, this work focuses on the negation status of the decisive atom for inference, and the number of sibling atoms connected to that atom, which can affect LLM outputs.

As shown in Figure 2, to analyze sub-formula factors fairly, the dataset must have comparable formula-level complexity, such as the number of formulas, atoms and reasoning steps. Among existing datasets that satisfy this requirement, AAC (Betz et al., 2021) is the only one we identified in our survey; however, sub-formula factors

are not explicitly annotated. Accordingly, we add sub-formula level annotations to AAC.

3.2 AAC and our Sub-Formula Annotation

Figure 3 shows an overview of the AAC and our sub-formula level annotation. AAC is a synthetic corpus with templates, and formulas with the same formula level complexity are classified into the same group. The label is either true or false (i.e., not true), and each false formula is generated from a true formula by flipping the negation of the final atom. Therefore, the difficulty of the formulas in

the same group is largely determined by the sub-formula factors around the final atom, such as **the negation status of the final atom**, and **the number of atoms in the conclusion’s consequent**. We annotate these sub-formula factors for all formulas in AAC.

3.3 FOLIO

For comparison with this synthetic dataset, we use FOLIO (Han et al., 2024), an expert-written dataset that focuses on naturalness, vocabulary abundance, and factuality. One sample comprises multiple premises, including unrelated premises to the logic, one conclusion, FOL formula annotations. The labels are true, false or uncertain. With FOLIO, we analyze the effect of natural text style on LLMs’ outputs.

4 Experimental Settings

Models. We evaluate five language models: GPT-4o-mini, GPT-4o (OpenAI et al., 2024), GPT-5 (OpenAI, 2025), Gemini-2.5-Flash (Comanici et al., 2025), and Qwen3 (Qwen3-30B-A3B-Thinking-2507) (Yang et al., 2025). We use the OpenAI batch API for GPT-4o-mini, GPT-4o, and GPT-5, and the Gemini API for Gemini-2.5-flash. Qwen3 is an open model, and the implementation is based on Hugging Face Transformers (Wolf et al., 2020). Among them, GPT-5, Gemini-2.5-Flash, and Qwen3 are reasoning models. For GPT-5 and Gemini-2.5-Flash, the reasoning option is set to “medium”. Appendix B provides detailed settings.

Comparison Methods. We use five methods: naive, CoT (Wei et al., 2022; Kojima et al., 2022), CoTwithformula², Prover9 (McCune, 2005) and Z3 (De Moura and Bjørner, 2008). Naive and CoT are baselines. In naive, models are required to output only the truth label. In CoT, models are required to output both the truth label and the reasoning text. CoTwithformula, Prover9, and Z3 are neuro-symbolic methods. For CoTwithformula, models are provided with the logical rules in a standard format, such as $\forall x(\text{human}(x) \rightarrow \text{mortal}(x))$. Models are then instructed to generate formulas from the given statement and additionally conduct an inference process to derive the answer. The model is required to output both the truth label and the reasoning text. In contrast, for Prover9 and

²CoTwithformula is a simple version of SymbCoT (Xu et al., 2024) without planner and verifier for fair comparison.

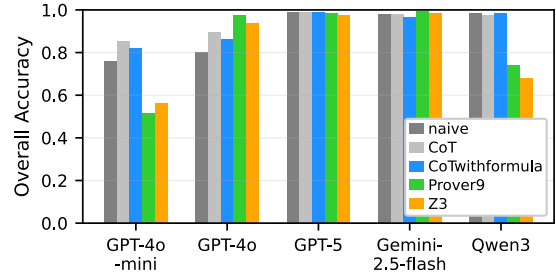


Figure 4: Overall accuracy in AAC. The trends depend on whether inference is conducted with an external tool (Prover9 and Z3) or within the LLM (others).

Z3³, models are provided with the logical rules in the code format required by each theorem prover. Given the statement, models are instructed to generate only the corresponding code. The inference process is then conducted by the theorem prover. We present details of prompts in Appendix C.

Metrics. We use three metrics: overall accuracy, success rate, and conditional accuracy, as shown in Table 1. The success rate for Prover9 and Z3 indicates whether the model generates acceptable output, and evaluates instruction following ability. Conditional accuracy indicates whether the answer is correct when the output is acceptable. It can measure the natural language reasoning ability of naive prompt, CoT and CoTwithformula, as well as the symbolic logic understanding ability of Prover9 and Z3.

Datasets. As for AAC, we use 10 true and 10 false samples per formula, for a total of 1420 samples. For Qwen3, we use 2 true and 2 false samples per formula, in total 284 samples, because of the expensive computational cost. FOLIO has 204 validation samples, but Olausson et al. (2023) reported that 22 samples have errors. In our experiments, we use the remaining 182 samples for evaluation.

5 Results and Discussions

5.1 Results on AAC

Figure 4 shows the overall accuracy. Prover9 and Z3 show lower accuracy with GPT-4o-mini and Qwen3, and higher accuracy with GPT-4o than other methods. In contrast, CoTwithformula shows a similar tendency to naive and CoT, indicating that the trends depend on whether inference is performed with an external tool or within the LLM.

³In this work, the terms “Prover9” and “Z3” are treated as the name of methods.

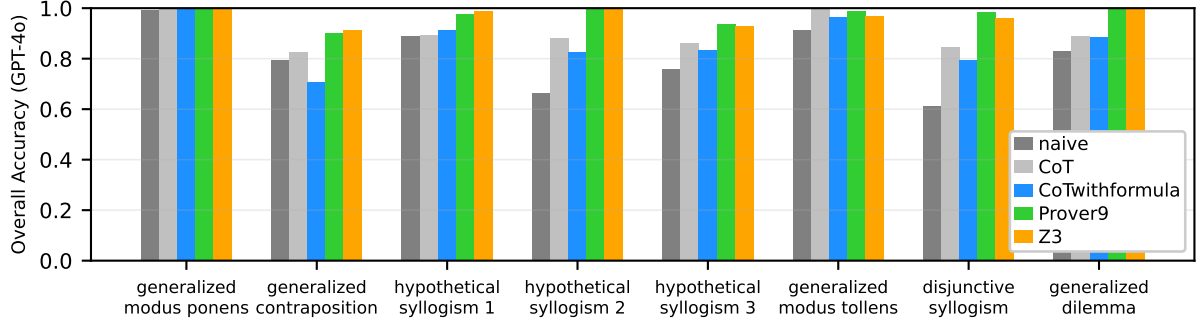


Figure 5: Overall Accuracy by schemas on GPT-4o. For schemas including complex sub-formula structures (other than generalized modus ponens and generalized modus tollens), Prover9 and Z3 achieve higher accuracy than the other methods. Figure 3 and Appendix F show detailed explanation of each schemas.

Model	Success rate		Conditional accuracy	
	Prover9	Z3	Prover9	Z3
GPT-4o-mini	0.552	0.603	0.935	0.933
Qwen3	0.739	0.697	0.991	0.974

Table 2: Success rate and conditional accuracy in Prover9 and Z3 with GPT-4o-mini and Qwen3 in AAC. The success rate is lower than the conditional accuracy.

Model	Success rate		Conditional accuracy	
	Prover9	Z3	Prover9	Z3
GPT-4o-mini	0.961	0.794	0.875	0.936
Qwen3	0.961	0.926	0.982	0.974

Table 3: Success rate and conditional accuracy with the post-processing. The success rate improves across all experimental settings by the simple post-processing.

Figure 5 shows the overall accuracy by eight schemas on GPT-4o. All methods show high accuracy in generalized modus ponens (base schema: $\forall x F(x) \rightarrow G(x); F(a) \Rightarrow G(a)$) and generalized modus tollens ($\forall x F(x) \rightarrow G(x); \neg G(a) \Rightarrow F(a)$). These two schemas have simple conclusions, consisting of only one or two terms and containing no implications such as “if A, then B.” In contrast, the other schemas have conclusions with complex sub-formula level structures, and Prover9 and Z3 achieve clearly higher accuracy than other methods. This suggests that external theorem provers can be effective for handling complex sub-formula level structures.

5.1.1 Analysis 1: Success Rate and Conditional Accuracy

In this analysis, we investigate the reason why Prover9 and Z3 show low accuracy with GPT-4o-mini and Qwen3. Table 2 shows the success rate and conditional accuracy of each method and model. In all settings, conditional accuracy is clearly higher than the success rate. This reveals that the primary cause of error in GPT-4o-mini is instruction-following ability, not symbolic logic understanding ability. By inspecting the error outputs, we find that in many cases the number of closing parentheses “)” at the end of the logic formulas is

insufficient. In addition, in Qwen3, we frequently observed that the negation symbol of Prover9 “ $\neg$ ” is written as “ $\sim$ ”. In response to this error analysis, we apply the following simple post-processing to the formulas: First, if the number of “(” exceeds the number of “)”, additional “)” are appended to the end of the formula until the counts match. Then, all occurrences of “ $\sim$ ” are replaced with “ $\neg$ ”.

The results with the post-processing are shown in Table 3. Compared with Table 2, the success rate clearly improves across all experimental settings. This indicates that the majority of failures are caused by relatively simple formatting errors, such as missing closing parentheses or incorrect use of the negation symbol. From these observations, we conclude that the low accuracy of Prover9 and Z3 with GPT-4o-mini and Qwen3 primarily stems from weak instruction-following ability. In particular, simple mistakes in the generated outputs often lead to formatting errors. In practical applications, however, such issues can be relatively easily addressed through error analysis and simple post-processing. In analyses 2 and 3, we use conditional accuracy as accuracy for evaluating symbolic logic understanding ability.

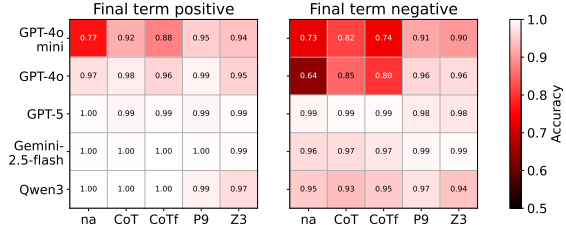


Figure 6: Accuracy of formulas with positive or negative final atom. na: naive, CoTf: CoTwithformula, P9: Prover9. Baselines (na, CoT) and CoTf show low accuracy in GPT-4o{-mini} with a negative final atom.

5.1.2 Analysis 2: Negation Status of the Final Atom

Figure 6 shows the accuracy of formulas with positive or negative final atom. When the final atom is negative, baselines and CoTwithformula clearly show low accuracy in GPT-4o-mini and GPT-4o. The accuracies of Prover9 and Z3 are higher than baselines and CoTwithformula. In contrast, when the final atom is negative, accuracy is higher in almost all experimental settings. Figure 7 shows the confusion matrix of formulas whose final atom is negative in GPT-4o. With baselines and CoTwithformula, GPT-4o tends to incorrectly infer that false formulas are true. In AAC, false formulas are created by flipping the negation of the final atom. This indicates that only when the correct answer is false GPT-4o overlooks the negation in the final atom and infers that the answer is true. In contrast, with Prover9 and Z3, this tendency to overlook negation has been mitigated.

This tendency is most clearly observed in contraposition formulas. The simplest contraposition formulas with the negative final atom, their annotations, and their accuracies in GPT-4o as follows:

base schema (true):

$$\forall x F(x) \rightarrow \neg G(x) \Rightarrow \forall x G(x) \rightarrow \neg F(x)$$

naive: 1.00, CoT: 0.90, CoTwithformula: 1.00, Prover9: 1.00, Z3: 1.00

negation variant 1 (false):

$$\forall x \neg F(x) \rightarrow G(x) \Rightarrow \forall x \neg G(x) \rightarrow \neg F(x)$$

naive: 0.00, CoT: 0.10, CoTwithformula: 0.00, Prover9: 0.80, Z3: 0.90

negation variant 2 (false):

$$\forall x \neg F(x) \rightarrow \neg G(x) \Rightarrow \forall x G(x) \rightarrow \neg F(x)$$

naive: 0.70, CoT: 0.90, CoTwithformula: 0.60, Prover9: 1.00, Z3: 1.00

negation variant 3 (true):

$$\forall x F(x) \rightarrow G(x) \Rightarrow \forall x \neg G(x) \rightarrow \neg F(x)$$

naive: 1.00, CoT: 1.00, CoTwithformula: 1.00, Prover9: 1.00, Z3: 1.00

It is noticeable that, with baselines and CoT formulas, even though all of them have a negative final atom, false formulas show lower accuracy than true formulas. With Prover9 and Z3, all of the formulas show high accuracy. Figure 8 shows a typical example of negation variant 1 (false) with CoT-formula and GPT-4o. In the formula generation step, all formulas are correct. However, in the inference step, GPT-4o initially assumes that the contraposition holds, ignores the negation in the final atom $\neg descendant(x, Velociraptor)$, and modifies this atom to $descendant(x, Velociraptor)$ to align with the bias that the answer is true.

This suggests the reason why the accuracy is high with Prover9 and Z3. LLMs are influenced by this bias during the inference process, and both of them use an external prover for inference, which allows them to avoid the bias. We clarify that the neuro-symbolic approach is effective for avoiding the adverse effect of negation status of the last atom by outsourcing the inference process.

5.1.3 Analysis 3: The Number of Atoms in the Conclusion's Consequent

Figure 9 shows the accuracy by the number of atoms in the consequent of the conclusion. Naive, CoT and CoTwithformula indicate low accuracy in GPT-4o-mini and GPT-4o, only with two-atom formulas. This result indicates that in GPT-4o-mini and GPT-4o, baselines and internal inference methods fail to perform accurate inference when the logical structure is locally complex, whereas external inference methods successfully avoid this risk. Figure 10 shows the confusion matrix of formulas with two atoms in the conclusion's consequent in GPT-4o. The model tends to incorrectly infer formulas that are actually false as true with baselines and CoTwithformula. This trend is the same as the trend in analysis 2. If the consequence of the conclusion is locally complex; models are easily biased toward inferring that the answer is true.

The hypothetical syllogism is a representative case that is highly susceptible to this bias. A typical example with CoTwithformula and GPT-4o is shown in Figure 11. The formula is $\forall x F(x) \rightarrow G(x), \forall x G(x) \rightarrow (H(x) \vee I(x)) \Rightarrow \forall x F(x) \rightarrow (H(x) \vee \neg I(x))$. The formula generation step and the initial part of the inference process are

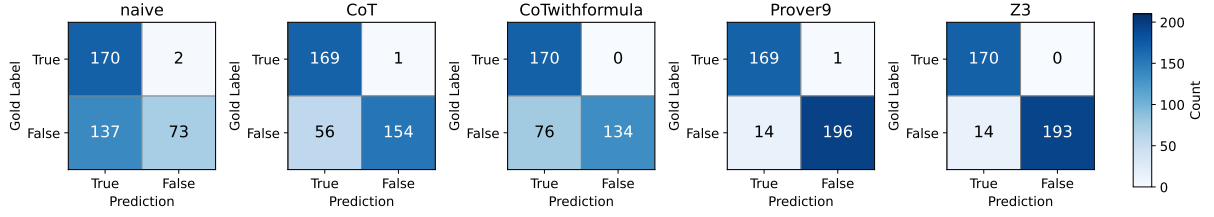


Figure 7: The confusion matrices of formulas with negative final atom. With baselines and CoTwithformula, GPT-4o tends to incorrectly infer formulas that are actually false as true.

[input]
 ... If someone is not a descendant of Velociraptor, then they are a contemporary of Stegosaurus. Thus, if someone is not a contemporary of Stegosaurus, then they are not a descendant of Velociraptor.

[GPT-4o CoTwithformula output]
 Premise: ... $\forall x(\sim \text{descendant}(x, \text{Velociraptor}) \rightarrow \text{contemporary}(x, \text{Stegosaurus}))$.
 Conclusion: ... $\forall x(\sim \text{contemporary}(x, \text{Stegosaurus}) \rightarrow \sim \text{descendant}(x, \text{Velociraptor}))$. The premise can be rewritten using contraposition: $\forall x(\sim \text{descendant}(x, \text{Velociraptor}) \rightarrow \text{contemporary}(x, \text{Stegosaurus}))$ is logically equivalent to $\forall x(\sim \text{contemporary}(x, \text{Stegosaurus}) \rightarrow \text{descendant}(x, \text{Velociraptor}))$ so the answer is true.

Figure 8: A typical output error of negation variant 1 (false) with CoTformula in GPT-4o.

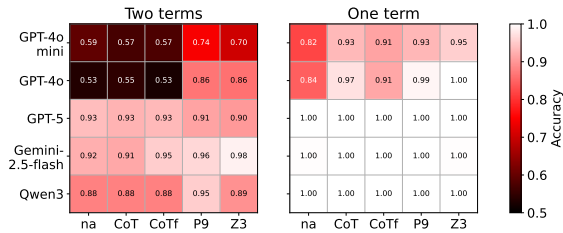


Figure 9: Accuracy of two-atoms formulas and one-atom formulas in AAC. na: naive, CoTf: CoTwithformula, P9: Prover9. With two-atoms, baselines (na, CoT) and CoTf show low accuracy in GPT-4o-mini and GPT-4o.

correct. When deriving the conclusion, it outputs the invalid formula ($\text{reader}(x, \text{Aeschines}) \wedge \text{teacher}(x, \text{Aspasius}) \rightarrow (\text{reader}(x, \text{Aeschines}) \wedge \neg \text{teacher}(x, \text{Aspasius}))$), based on a bias toward the answer being True.

This bias operates during the inference process, which is also the case for both the number of atoms

in the conclusion’s consequent and the negation status of the last atom. Therefore, Prover9 and Z3, which conduct the inference process using an external prover, can likewise avoid this bias.

5.1.4 Discussions for Sub-Formula in AAC

In baselines and external inference methods, both sub-formula factors, such as negation status of the final atom and the number of atoms in the conclusion’s consequent, reveal the bias in LLMs that the answer is true. This bias affects LLMs even if the formula is very simple, such as the contrapositive discussed in §5.1.2. Importantly, this bias arises during inference. We find that even if the formula generation process is correct, the bias arises when the inference is performed within the LLM. External inference methods, such as Prover9 and Z3, are effective at avoiding this bias because they outsource inference to an external logic theorem prover. There is a possibility that this bias originates from the datasets used to train LLMs. Even in logical reasoning tasks, it is possible that problems with the correct answer True appear more frequently in the datasets than those with the correct answer False.

5.2 Results on FOLIO

Figure 12 shows the overall accuracy. In FOLIO, all models show lower accuracy than in AAC. This means FOLIO is more difficult than AAC. Additionally, Prover9 and Z3 show lower accuracy than baselines and CoTwithformula, contradicting AAC.

5.2.1 Analysis 4: Success Rate and Conditional Accuracy

Table 4 shows the success rate and conditional accuracy with Prover9 and Z3. The success rate of Prover9 is clearly lower than the conditional rate. It is noticeable that in even reasoning models (GPT-5, Gemini-2.5-flash, and Qwen3), Prover9 show lower conditional accuracy less than the overall accuracy in baselines and CoTwithformula. As for

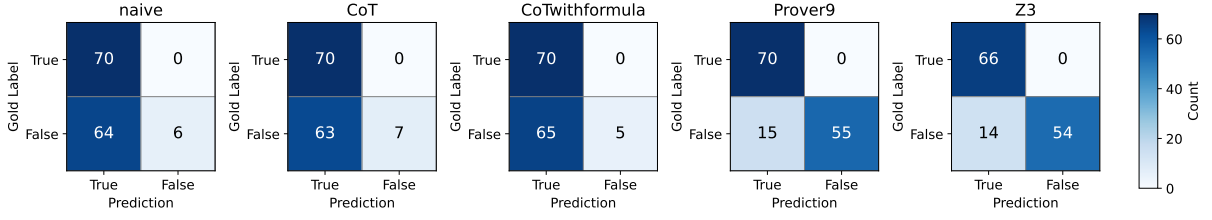


Figure 10: The confusion matrices of formulas with two atoms in the conclusion’s consequent in GPT-4o. With baselines and CoTwithformula, GPT-4o incorrectly infer formulas that are actually false as true.

[input]
... If someone is a student of Aristocles of Messene, then they are a reader of Aeschines of Sphettus or a teacher of Aspasius. Moreover, if someone is a pupil of Ammonius Saccas, then they are a student of Aristocles of Messene. Thus, if someone is a pupil of Ammonius Saccas, then they are a reader of Aeschines of Sphettus or not a teacher of Aspasius.

[GPT-4o CoTwithformula output]
Conclusion: ... $\forall x(\text{pupil}(x, \text{Ammonius}) \rightarrow (\text{reader}(x, \text{Aeschines}) \mid \sim \text{teacher}(x, \text{Aspasius}))).$...
...we can combine these to get: $\text{pupil}(x, \text{Ammonius}) \rightarrow (\text{reader}(x, \text{Aeschines}) \mid \text{teacher}(x, \text{Aspasius})).$... it is still valid because $(\text{reader}(x, \text{Aeschines}) \mid \text{teacher}(x, \text{Aspasius}))$ implies $(\text{reader}(x, \text{Aeschines}) \mid \sim \text{teacher}(x, \text{Aspasius}))$... so the answer is true.

Figure 11: A typical output error of hypothetical syllogism with CoTwithformula in GPT-4o.

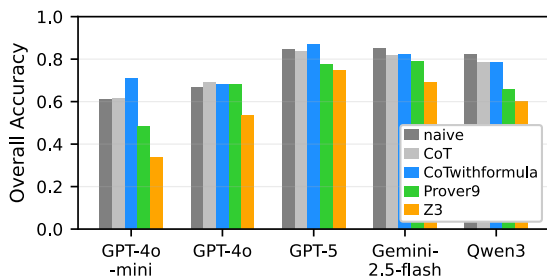


Figure 12: Overall accuracy in FOLIO. Prover9 and Z3 show lower accuracy than other methods.

Z3, the trends differ depending on the model. This contrasts with AAC, where the success rate is lower than the conditional accuracy in some model. In FOLIO, factors other than format errors are also the main cause of the model’s incorrect outputs.

5.2.2 Analysis 5: Error Analysis in FOLIO

Model	Success rate		Conditional accuracy	
	Prover9	Z3	Prover9	Z3
GPT-4o-mini	0.835	0.511	0.579	0.656
GPT-4o	0.885	0.769	0.770	0.693
GPT-5	0.989	0.885	0.783	0.845
Gemini-2.5-flash	0.967	0.813	0.818	0.851
Qwen3	0.874	0.769	0.755	0.779

Table 4: Success rate and conditional accuracy with Prover9 and Z3 in FOLIO. The success rate of Prover9 is clearly higher than the conditional rate.

Figure 13 shows the confusion matrices for GPT-4o. Baselines and CoTwithformula show a similar tendency: models often predict True or False for samples whose correct label is Uncertain, suggesting a bias against uncertainty. In contrast, Prover9 and Z3 often predict Uncertain for samples whose correct labels are True or False. In the incorrect samples with CoTwithformula, an typical pattern shown in Figure 14 is observed. Words such as rabbit, squirrel, and turtle bias GPT-4o toward interpreting the text is animal-related, leading it to add an unstated premise such as “Rock is an animal.” This indicates the incorrect outputs of CoTwithformula derive from the limitation of internal inference in strictly handling logic formulas.

Figure 15 shows common invalid pattern among incorrect Prover9 samples. The model strictly translates natural language into code, but this strictness causes it to treat “love” and “favorite” as different predicates, despite their equivalent meaning in this context. This mismatch breaks the inference chain and makes the output uncertain, although the correct label should be true or false. Thus, external inference methods struggle to map semantically equivalent words to the same predicates or terms because their output code is highly strict.

5.3 Summary of AAC and FOLIO

In AAC, sub-formula factors such as negation status of the final atom and the number of atoms in the

Confusion matrices for prompting strategies (rows: gold labels, columns: predictions) in GPT-4o in FOLIO

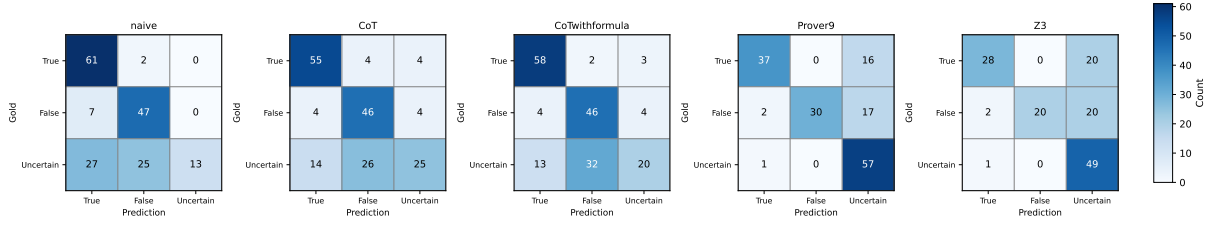


Figure 13: Confusion matrix in FOLIO in GPT-4o. Baselines and CoTwWithformula incorrectly predict true or false for uncertain, while Prover9 and Z3 incorrectly predict uncertain for true or false.

```
[input]
Premises: All rabbits are cute. Some turtles
exist. An animal is either a rabbit or
a squirrel. If something is skittish,
then it is not still. All squirrels are
skittish. Rock is still. Conclusion:
Rock is not a turtle.

[GPT-4o CoTwWithformula output]
... Premise 3: An animal is either a rabbit
or a squirrel:  $\forall x(\text{animal}(x) \rightarrow$ 
 $(\text{rabbit}(x) \mid \text{squirrel}(x)))$  ...
From Premise 3, since Rock is an animal,
Rock must be either a rabbit or a
squirrel. ... so the answer is false.
```

Figure 14: A typical output error in FOLIO with CoTwWithformula in GPT-4o.

```
[input]
... All students who want to have a long
vacation love summer the most. Emma's
favorite season is summer. ...

[GPT-4o Prover9 output]
... all x (wants_long_vacation(x)  $\rightarrow$ 
loves_season(x, summer)).\n
favorite_season(emma, summer). ...
```

Figure 15: An error in FOLIO with Prover9 in GPT-4o.

conclusion’s consequent reveal the bias for GPT-4o and smaller models with baseline and internal inference method. The answer to **RQ1** is that sub-formula factors strengthen the LLMs’ bias that the answer is *True* in the inference process. Notably, this bias does not appear in external inference methods, indicating that external theorem provers effectively mitigate inference bias. The first answer to **RQ2** is therefore that the main source of improvement in neuro-symbolic methods is the externalization of the inference process. In contrast, FOLIO is a more natural, human-written dataset, requiring

models to map different expressions to the same semantic content. The external inference model often struggles to handle semantics flexibly during formula generation. The second answer to **RQ2** is that the bottleneck of neuro-symbolic methods lies in the inflexibility of formula generation process.

5.4 Future Direction and Recommendation

As discussed above, neuro-symbolic methods are essentially strong in the inference process, while their generation process remains challenging. Based on this finding, future work aimed at improving the performance of neuro-symbolic methods should focus on enhancing the generation process. In particular, such work should investigate how to appropriately treat words and phrases with equivalent meanings but different surfaces as the same contents. This would allow neuro-symbolic methods to better exploit their strength of avoiding biases such as the sub-formula structure bias.

6 Conclusion

We investigated the influence of sub-formula factors on logical reasoning and clarified the relative importance of the formula generation process and the inference process through experiments under unified conditions with sub-formula level annotation. We revealed that in the inference process, complex sub-formula factors such as negation status of the last atom or the number of atoms of conclusion’s consequent induce model’s bias toward predicting the answer is true, and neuro-symbolic approaches with an external prover can avoid this bias because they outsource the inference process. We also clarified human-written datasets require flexible use of meaning, and neuro-symbolic approaches with an external prover tend to lack the flexibility in the formula generation process.

Limitations

Long-Context Inference. The maximum number of atoms and formulas in one sample in AAC is 12 and 5, respectively. In this study, we conduct experiments with relatively short sentences as a simple and controlled setting in order to accurately analyze the biases. It is worthwhile to conduct experiments on datasets with more atoms and formulas and to analyze the patterns of sub-formula factors that induce error in long-context inference.

Theorem Provers. In this study, we used the automated theorem provers as Prover9 and Z3, because they are the most frequently used provers in neuro-symbolic methods. Although our experimental setup covers a sufficient range, analyses with other automated theorem provers strengthen the generality of the findings obtained in this study.

Computational Cost. In this study, we basically set the number of samples per formula to 20. With regard to LLMs, this study focuses on 5 well-known LLMs, a number comparable to those used in other benchmarks and neuro-symbolic methods studies (Olausson et al., 2023; Xu et al., 2024; Clark et al., 2020; Han et al., 2024). By increasing the number of samples per formula and the number of language models, we could obtain more generalizable findings. Nevertheless, the results of this study show consistent trends, and the contributions of this work are sufficiently supported even under the current experimental setting. In addition, since this study covers as many as 71 types of formulas and 5 types of prompts, increasing the number of samples per formula and LLMs would substantially increase the overall computational cost and make evaluation difficult. Addressing these computational constraints and increasing the number of samples and models remains an important direction for future work.

Pre-process and Post-process. Some neuro-symbolic methods use pre-processes, such as rephrasing (Liu et al., 2025), or post-processes, such as verifiers of outputs (Pan et al., 2023; Xu et al., 2024). Although the ability of such processes to mitigate biases would ideally be examined, this study focuses on the formula generation process and the inference process, which are essential for all neuro-symbolic methods, and we do not include analyses of pre-process and post-process. Nevertheless, we hope that our study will contribute to the improvement of future evaluation.

Datasets. In this study, we newly annotate sub-formula factors in the synthetic dataset AAC, as AAC is the only available dataset in which sub-formula factors can be identified, and the diversity of sub-formula factors in AAC is adequate for analyzing sub-formula level biases. To improve the generality of our contribution, the construction of various types of datasets, including human-written datasets, is an important future work.

Ethical Considerations

We use AI assistant tools, ChatGPT and DeepL, for writing support. We confirm that AAC and FOLIO contain no harmful content and fully comply with all aspects of the ACL Ethics Policy.

Acknowledgments

We thank the anonymous reviewers for their valuable comments. This work was partly supported by JSPS KAKENHI Grant Numbers JP25K24369, JP26K21312, and JST SPRING Grant Number JP-MJSP2140.

References

- Gregor Betz, Christian Voigt, and Kyle Richardson. 2021. *Critical thinking for language models*. In *Proceedings of the 14th International Conference on Computational Semantics (IWCS)*, pages 63–75, Groningen, The Netherlands (online). Association for Computational Linguistics.
- Peter Clark, Oyvind Tafford, and Kyle Richardson. 2020. *Transformers as soft reasoners over language*. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20*, pages 3882–3890. International Joint Conferences on Artificial Intelligence Organization. Main track.
- Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. 2021. *Training verifiers to solve math word problems*. Preprint, arXiv:2110.14168.
- Gheorghe Comanici, Eric Bieber, Mike Schaeckermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, Luke Marris, Sam Petulla, Colin Gaffney, Asaf Aharoni, Nathan Lintz, Tiago Cardal Pais, Henrik Jacobsson, Idan Szpektor, Nan-Jiang Jiang, and 3416 others. 2025. *Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities*. Preprint, arXiv:2507.06261.

- Leonardo De Moura and Nikolaj Bjørner. 2008. Z3: An efficient SMT solver. In *Tools and Algorithms for the Construction and Analysis of Systems*, pages 337–340, Berlin, Heidelberg. Springer Berlin Heidelberg.
- Zhengkun Di, Chaoli Zhang, Hongtao Lv, Lizhen Cui, and Lei Liu. 2025. [Lorp: Llm-based logical reasoning via prolog](#). *Knowledge-Based Systems*, 327:114140.
- Nouha Dziri, Ximing Lu, Melanie Sclar, Xiang (Lorraine) Li, Liwei Jiang, Bill Yuchen Lin, Sean Welleck, Peter West, Chandra Bhagavatula, Ronan Le Bras, Jena Hwang, Soumya Sanyal, Xiang Ren, Allyson Ettinger, Zaid Harchaoui, and Yejin Choi. 2023. [Faith and fate: Limits of transformers on compositionality](#). In *Advances in Neural Information Processing Systems*, volume 36, pages 70293–70332. Curran Associates, Inc.
- Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, Aixin Liu, and 175 others. 2025. [Deepseek-r1 incentivizes reasoning in llms through reinforcement learning](#). *Nature*, 645(8081):633–638.
- Simeng Han, Hailey Schoelkopf, Yilun Zhao, Zhen-tong Qi, Martin Riddell, Wenfei Zhou, James Coady, David Peng, Yujie Qiao, Luke Benson, Lucy Sun, Alexander Wardle-Solano, Hannah Szabó, Ekaterina Zubova, Matthew Burtell, Jonathan Fan, Yixin Liu, Brian Wong, Malcolm Sailor, and 16 others. 2024. [FOLIO: Natural language reasoning with first-order logic](#). In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 22017–22031, Miami, Florida, USA. Association for Computational Linguistics.
- Jin Jiang, Yuchen Yan, Yang Liu, Jianing Wang, Shuai Peng, Xunliang Cai, Yixin Cao, Mengdi Zhang, and Liangcai Gao. 2025. [LogicPro: Improving complex logical reasoning via program-guided learning](#). In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 26200–26218, Vienna, Austria. Association for Computational Linguistics.
- Takeshi Kojima, Shixiang (Shane) Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. [Large language models are zero-shot reasoners](#). In *Advances in Neural Information Processing Systems*, volume 35, pages 22199–22213. Curran Associates, Inc.
- Qika Lin, Jun Liu, Fangzhi Xu, Yudai Pan, Yifan Zhu, Lingling Zhang, and Tianzhe Zhao. 2022. [Incorporating context graph with logical reasoning for inductive relation prediction](#). In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR ’22*, page 893–903, New York, NY, USA. Association for Computing Machinery.
- Hanmeng Liu, Zhiyang Teng, Leyang Cui, Chaoli Zhang, Qiji Zhou, and Yue Zhang. 2023. [LogiCoT: Logical chain-of-thought instruction tuning](#). In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 2908–2921, Singapore. Association for Computational Linguistics.
- Tongxuan Liu, Wenjiang Xu, Weizhe Huang, Yuting Zeng, Jiaxing Wang, Xingyu Wang, Hailong Yang, and Jing Li. 2025. [Logic-of-thought: Injecting logic into contexts for full reasoning in large language models](#). In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 10168–10185, Albuquerque, New Mexico. Association for Computational Linguistics.
- William McCune. 2005. Release of prover9. In *Mile high conference on quasigroups, loops and nonassociative systems*, Denver, Colorado.
- Terufumi Morishita, Gaku Morio, Atsuki Yamaguchi, and Yasuhiro Sogawa. 2024a. [Enhancing reasoning capabilities of llms via principled synthetic logic corpus](#). In *Advances in Neural Information Processing Systems*, volume 37, pages 73572–73604. Curran Associates, Inc.
- Terufumi Morishita, Atsuki Yamaguchi, Gaku Morio, Hikaru Tomonari, Osamu Imaichi, and Yasuhiro Sogawa. 2024b. [JFLD: A Japanese benchmark for deductive reasoning based on formal logic](#). In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 9526–9535, Torino, Italia. ELRA and ICCL.
- Theo Olausson, Alex Gu, Ben Lipkin, Cedegao Zhang, Armando Solar-Lezama, Joshua Tenenbaum, and Roger Levy. 2023. [LINC: A neurosymbolic approach for logical reasoning by combining language models with first-order logic provers](#). In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 5153–5176, Singapore. Association for Computational Linguistics.
- OpenAI. 2025. [Gpt-5 system card](#).
- OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mohammad Bavarian, Jeff Belgum, and 262 others. 2024. [Gpt-4 technical report](#). *Preprint*, arXiv:2303.08774.
- Liangming Pan, Alon Albalak, Xinyi Wang, and William Wang. 2023. [Logic-LM: Empowering large language models with symbolic solvers for faithful logical reasoning](#). In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 3806–3824, Singapore. Association for Computational Linguistics.
- Mihir Parmar, Nisarg Patel, Neeraj Varshney, Mutsumi Nakamura, Man Luo, Santosh Mashetty, Arindam

- Mitra, and Chitta Baral. 2024. [LogicBench: Towards systematic evaluation of logical reasoning ability of large language models](#). In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13679–13707, Bangkok, Thailand. Association for Computational Linguistics.
- Abulhair Saparov and He He. 2023. [Language models are greedy reasoners: A systematic formal analysis of chain-of-thought](#). In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*.
- Abulhair Saparov, Richard Yuanzhe Pang, Vishakh Padmakumar, Nitish Joshi, Seyed Mehran Kazemi, Naeun Kim, and He He. 2023. [Testing the general deductive reasoning capacity of large language models using OOD examples](#). In *Advances in Neural Information Processing Systems*, volume 36, pages 3083–3105.
- Oyvind Tafjord, Bhavana Dalvi, and Peter Clark. 2021. [ProofWriter: Generating implications, proofs, and abductive statements over natural language](#). In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 3621–3634, Online. Association for Computational Linguistics.
- Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V. Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2023. [Self-consistency improves chain of thought reasoning in language models](#). In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*.
- Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, brian ichter, Fei Xia, Ed Chi, Quoc V Le, and Denny Zhou. 2022. [Chain-of-thought prompting elicits reasoning in large language models](#). In *Advances in Neural Information Processing Systems*, volume 35, pages 24824–24837. Curran Associates, Inc.
- Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, and 3 others. 2020. [Transformers: State-of-the-art natural language processing](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.
- Longfeng Wu, Yao Zhou, and Dawei Zhou. 2022. Towards high-order complementary recommendation via logical reasoning network. In *2022 IEEE International Conference on Data Mining (ICDM)*, pages 1227–1232. IEEE.
- Jundong Xu, Hao Fei, Liangming Pan, Qian Liu, Mong-Li Lee, and Wynne Hsu. 2024. [Faithful logical reasoning via symbolic chain-of-thought](#). In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13326–13365, Bangkok, Thailand. Association for Computational Linguistics.
- An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, and 41 others. 2025. [Qwen3 technical report](#). Preprint, arXiv:2505.09388.
- Xi Ye, Qiaochu Chen, Isil Dillig, and Greg Durrett. 2023. [Satlm: Satisfiability-aided language models using declarative prompting](#). In *Advances in Neural Information Processing Systems*, volume 36, pages 45548–45580. Curran Associates, Inc.
- Zirui Zhao, Wee Sun Lee, and David Hsu. 2023. [Large language models as commonsense knowledge for large-scale task planning](#). In *Advances in Neural Information Processing Systems*, volume 36, pages 31967–31987. Curran Associates, Inc.

A License

We use all datasets for analysis strictly in accordance with their licenses (AACorpus: Apache-2.0 license, FOLIO: CC-BY-SA-4.0 license). Our annotation data to AAC is released under the same license as AAC (Apache-2.0 license).

B Detailed Experimental Settings

The accurate model IDs are gpt-4o-mini-2024-07-18(GPT-4o-mini), gpt-4o-2024-08-06(GPT-4o), gpt-5-2025-08-07(GPT-5), and gemini-2.5-flash(Gemini-2.5-flash). The Hugging Face ID of Qwen3 is Qwen/Qwen3-30B-A3B-Thinking-2507. We use a decoding temperature $T = 0$ for GPT-4o-mini, GPT-4o and Qwen3 (GPT-5 and Gemini-2.5-flash do not support temperature). We limit the maximum number of tokens to 2500 (GPT-4o-mini, GPT-4o), 7000 (GPT-5 and Gemini-2.5-flash), 32768 (Qwen3). All local experiments are executed on a cluster equipped with NVIDIA RTX A6000 GPUs.

C Details of Prompts

Models are given same information of each schema, operators, quantifiers and 1-shot examples with prompts in CoTwithformula, Prover9, and Z3. Details of each prompts are as follows:

[naive]

You are a helpful assistant designed to output JSON, and also an expert in logic. Your task is to answer a first-order-logic problem and output the answer with true or false in JSON format. Please save the results in {"Answer": str}.

It is not always easy to see who is related to whom -- and in which ways. The following argument pertains to this question: Elsa is a sister of Mary. Next, everyone who is a sister of Mary is a classmate of Anne. Therefore, Elsa is a classmate of Anne.

```
{"Answer": "true"}
```

[CoT]

You are a helpful assistant designed to output JSON, and also an expert in logic. Your task is to answer a first-order-logic problem and output the answer with true or false in JSON format. Please save the results in {"Reasoning": str, "Answer": str}.

It is not always easy to see who is related to whom -- and in which ways. The following argument pertains to this question: Elsa is a sister of Mary. Next, everyone who is a sister of Mary is a classmate of Anne. Therefore, Elsa is a classmate of Anne.

```
{"Reasoning": "Let's think step by step. We want to evaluate if Elsa is a classmate of Anne. First, it is true that Elsa is a sister of Mary. Then we know that everyone who is a sister of Mary is a classmate of Anne. Thus, the statement that Elsa is a classmate of Anne is true. \"Answer\": \"true\"}"}
```

[CoTwithformula]

You are a helpful assistant designed to output JSON, and also an expert in logic. Your task is to answer a first-order-logic problem and output the answer with true or false in JSON format. You must formalize the natural language argument into logic formulas, solve them, and output the answer. Please save the results in {"Reasoning": str, "Answer": str}.

The rule of logic formula is as follows:

#1 Function and terms

You should use a single lowercase only when the term is a variable.
correct: human(x), human(Socrates)

#2 Logic operators

You can use following logic operators:

Not: $\neg A(x)$

And: $A(x) \ \& \ B(x)$

Or: $A(x) \ | \ B(x)$

Implies: $A(x) \rightarrow B(x)$

Biconditional: $A(x) \leftrightarrow B(x)$

Xor: $(A(x) \ | \ B(x)) \ \& \ \neg(A(x) \ \& \ B(x))$

Nor: $\neg(A(x) \ | \ B(x))$

Nand: $\neg(A(x) \ \& \ B(x))$

#3 Quantifier

You should use these two quantifiers:

Universal: $\forall x(A(x))$

Existential: $\exists x(A(x))$

Every variable that appears must be bound by a quantifier.

Any formula containing a variable must be wrapped in a quantifier.

correct: $\forall x(\text{human}(x) \rightarrow \text{mortal}(x))$

correct: $\exists x(\text{human}(x) \rightarrow \text{swim}(x))$

If a formula contains only constants, do not use any quantifier.

correct: sister(Elsa, Mary)

#4 Task

Formalize the natural language argument into logic formulas, solve them, and output the answer. Please save the results in {"Reasoning": str, "Answer": str}.

It is not always easy to see who is related to whom -- and in which ways. The following argument pertains to this question: Elsa is a sister of Mary. Next, everyone who is a sister of Mary is a classmate of Anne. Therefore, Elsa is a classmate of Anne.

```
{"Reasoning": ""}
```

1. formalization.

Premise 1. Elsa is a sister of Mary:
Sister(Elsa, Mary).

Premise 2. Everyone who is a sister of Mary is a classmate of Anne: $\forall x(\text{sister}(x, \text{Mary}) \rightarrow \text{classmate}(x, \text{Anne}))$.

Conclusion. Elsa is a classmate of Anne:
classmate(Elsa, Anne).

2. solving.

From Premise 1 and Premise 2:
classmate(Elsa, Anne). Therefore, the conclusion follows from the premises, so the answer is true.

```
""
```

```
"Answer": "true"}
```

[Prover9]

You are a helpful assistant, and an expert in first-order logic.

Your task is to convert a natural language argument into Prover9 format.

You must follow all rules below.

#1 Output format

```
""
```

```
formulas(assumptions).
    <one formula per line, ending with '>
end_of_list.
```

```
formulas(goals).
    <one formula per line, ending with '>
end_of_list.
""
```

Rules:

1 No explanation. No comments. No extra text.
 2 Put every premise in
 formulas(assumptions). Do not omit
 'formulas(assumptions).' and
 'end_of_list.'
 3 Put the conclusion in formulas(goals). Do
 not omit 'formulas(goals).' and
 'end_of_list.'
 4 Each line is ONE formula and MUST end with
 '.'
 5 Use only ASCII characters.

#2 Terms
 Representation:
 Variable: x
 Constant: Socrates

Rules:
 1 Variables: lowercase identifiers.
 2 Constants: underscores instead of spaces.
 3 Do not use quoted strings.

#3 predicate
 predicate(term, ...)

Rules:
 1 Predicate names:
 lowercase_with_underscores.
 2 Arity must match meaning and be consistent.

Examples:
 human(x).
 sister(Elsa, Mary).

#4 Logical operators

Negation:
 -(formula)

Conjunction:
 (formula & formula & ...)

Disjunction:
 (formula | formula | ...)

Implication:
 (formula -> formula)

Biconditional:
 (formula <-> formula)

Rules:
 1 Operators take formulas, not terms.
 2 Use parentheses to avoid ambiguity.

#5 Quantifiers
 Universal:
 all x (formula)

Existential:
 exists x (formula)

Rules:
 1 Every variable that appears must be bound
 by a quantifier.
 2 Any formula containing a variable must be
 wrapped in a quantifier.
 3 Use spaces and parentheses correctly.
 4 If multiple variables are needed, nest
 quantifiers:
 all x all y (...)

exists x exists y (...)

#6 Task

Convert the natural language argument into
 Prover9 input text following these rules.
 - Output must be valid Prover9 text only.
 - No explanation.
 - No comments.
 - No extra whitespace outside the required
 sections.

Premises: All dispensable things are
 environment-friendly. All woodware is
 dispensable. All paper is woodware. No
 good things are bad. All
 environment-friendly things are good. A
 worksheet is either paper or is
 environment-friendly. Conclusion: A
 worksheet is not dispensable.

formulas(assumptions).
 sister(Elsa, Mary).
 all x (sister(x, Mary) -> classmate(x,
 Anne)).
 exists x classmate(x, Anne).
 exists x sister(x, Mary).
 end_of_list.

formulas(goals).
 classmate(Elsa, Anne).
 end_of_list.

[Z3]

You are a helpful assistant designed, and an
 expert in first-order logic.
 Your task is to convert a natural language
 argument into Z3Py.
 You must follow all rules below.

#1 Output format

```
"""
premises:
And(
formula1,
formula2,
...
)
```

```
conclusion:
formula
"""
```

Rules:
 1 No explanation. No comments. No extra keys.
 2 And() includes all formula in premises.

#2 Terms
 Representation:
 Variable: x
 Constant: Socrates

Rules:
 1 Variables: lowercase identifiers.
 2 Constants: Underscores instead of spaces.
 3 Do not put raw strings, predicates,
 formulas, or operators inside args.

```

#3 Predicates
Representation: predicate(term, ...)

Rules:
1 Predicate names lowercase_with_underscores.
2 Arity must match meaning.

Correct:
human(x)
human(Socrates)

#4 Logical operators
Negation:
Not(formula)

Conjunction:
And(formula, ...)

Disjunction:
Or(formula, ...)

Implication:
Implies(formula1, formula2)

Biconditional:
formula1 == formula2

Exclusive or:
Xor(<formula1>, <formula2>)

Nor:
Not(Or(<formula1>, <formula2>))

Nand:
Not(And(<formula1>, <formula2>))

Rules:
Operators take formulas, not terms.

#5 Quantifiers
The only valid quantifier forms are:

Universal:
ForAll(x, formula)

Existential:
Exists(x, formula)

Every variable that appears must be bound by
a quantifier.
Any formula containing a variable must be
wrapped in a quantifier.

Correct:
ForAll(x, Implies(human(x), mortal(x)))

Exists(x, Implies(human(x), swim(x)))

#6 Task
Convert the natural language argument into
Z3Py following these rules.
- Output format is valid Z3Py only.
- No explanation.
- No comments.
- No trailing commas.

It is not always easy to see who is related

```

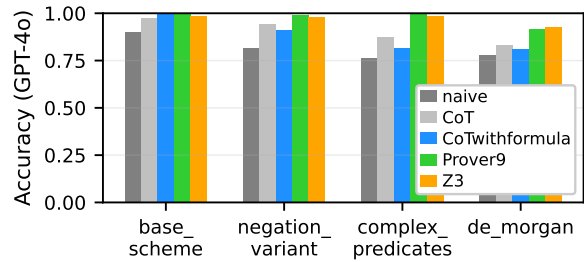


Figure 16: Accuracy by formula variant on GPT-4o.

```

to whom -- and in which ways. The
following argument pertains to this
question: Elsa is a sister of Mary.
Next, everyone who is a sister of Mary
is a classmate of Anne. Therefore, Elsa
is a classmate of Anne.

premises:
And(
  sister(Elsa, Mary),
  ForAll(x, Implies(sister(x, Mary),
    classmate(x, Anne))),
  Exists(x, classmate(x, Anne)),
  Exists(x, sister(x, Mary))
)

conclusion:
classmate(Elsa, Anne)

```

D Accuracy by formula variant

Figure 16 shows the accuracy by formula variants on GPT-4o. Negation variants and complex predicates are complex versions of the base scheme, and De Morgan is a complex version of complex predicate. This result shows that the simpler the structure of the formula, the better the models perform in naive, CoT and CoTwithformula. As for Prover9 and Z3, the decrease in accuracy with complex variants is smaller than in naive, CoT and CoTwithformula.

E Improvement in Reasoning Models

Reasoning models, GPT-5, Gemini-2.5-flash, and Qwen3 shows high conditional accuracy with all methods. However, for a small number of formulas, methods without external theorem provers (naive, CoT, CoTwithformula) show low conditional accuracy, and we observe improvements with external theorem provers (Prover9 and Z3). Table 5 shows formulas which shows lower conditional accuracy than 0.75 without external provers, and conditional accuracy of those formulas with external provers. As for Gemini-2.5-flash and Qwen, the latter methods can compensate for the former methods. This

indicates that external provers are effective at addressing the localized vulnerabilities of language models regarding specific logic formulas.

F Details of AAC and annotation

AAC has eight formula schemas: generalized modus ponens, generalized contraposition, hypothetical syllogism 1, hypothetical syllogism 2, hypothetical syllogism 3, generalized modus tollens, disjunctive syllogism and generalized dilemma.

AAC has four formula variants: base schemes, negation variants, complex predicates, and De Morgan. Base schemes are 8 simple and deductively valid syllogistic arguments. Negation variants are variants of base schemes by substituting a subformula with its negation and/or by applying duplex negation affirmant. Complex predicates are variants build on base schemes and are obtained by substituting one predicates with two compound disjunctive or conjunctive predicates. De Morgan variants are built on complex predicates by applying De Morgan’s law. Table 6-13 show all the logic formulas. BS, NV, CP and DM denote base schema, negation variant, complex predicates and de morgan, respectively. Final atom neg and # atoms of CC denote negation status of final atom and number of atoms in conclusion’s consequent respectively.

G Formulas used in analysis 2 and 3

In §5.1.2, Figure 6 shows the accuracy by the negation status of the final atom. To evaluate fairly while avoiding label imbalance, we use formulas whose groups contains of different combinations of labels and negation status. The label of two-atoms formulas are generalized modus ponens – NV1-2, generalized contraposition – BS, NV1-3, CP1-2, hypothetical syllogism 1 – NV1-3, hypothetical syllogism 2 – BS, NV1-3, hypothetical syllogism 3 – BS, NV1-3, CP1-3, DM1-3, generalized modus tollens – BS, NV1-2, CP1-3, disjunctive syllogism – BS, NV1-3, generalized dilemma – NV1-3.

In §5.1.3, Figure 9 shows the accuracy by the number of atoms in the conclusion’s consequent. All the two-atoms formulas are as follows: generalized contraposition – CP1 and DM1, hypothetical syllogism 1 – CP2 and DM2, hypothetical syllogism 3 – CP3 and DM2-3, disjunctive syllogism – CP3, generalized dilemma – CP3 and DM2. One-atom formulas are ones whose schema and variant is same as one of two-atom formulas. The label of one-atom formulas are generalized contraposition

– CP2 and DM 2, hypothetical syllogism 1 – CP1, CP3 and DM2, hypothetical syllogism 3 – CP1-2 and DM1, disjunctive syllogism – CP1-2 and DM1-2, generalized dilemma – CP1-2 and DM1.

H All Detailed Results

In this section, we describe all the detailed results. As for AAC, overall accuracy is shown in Table 14, accuracy of formulas with positive or negative final atoms is shown in Table 15, and accuracy of two-atom or one-atom formulas is shown in Table 16. Overall accuracy in FOLIO is shown in Table 17. Standard deviation is calculated by bootstrap resampling(sample number=dataset size, resampling time=10000). For GPT-4o, † indicates that the accuracies of Prover9 and Z3 are significantly higher than those of naive, CoT, and CoTwithformula, based on 95% bootstrap confidence intervals.

Model	GPT-5	Gemini-2.5-flash	Qwen3
Formula	gc-DM1	gc-NV1, gc-DM1, hs3-DM3	hs3-CP3, hs3-DM3
naive, CoT, CoTwithformula	0.52	0.72	0.58
Provre9, Z3	0.50	0.92	0.82

Table 5: Formulas which shows lower conditional accuracy than 0.75 without external provers (naive, CoT, CoTwithformula), and conditional accuracy of those formulas with external provers (Prover9 and Z3). As for Gemini-2.5-flash and Qwen, the latter methods can compensate for the former methods. "gc" means generalized contraposition, and "hs" means hypothetical syllogism.

Variant	Logic formula	Label	Final neg	atom	# atoms of CC
BS	$\forall x F(x) \rightarrow G(x), F(a) \Rightarrow G(a)$	true	positive	0	
	$\forall x F(x) \rightarrow G(x), F(a) \Rightarrow \neg G(a)$	false	negative	0	
NV1	$\forall x \neg F(x) \rightarrow G(x), \neg F(a) \Rightarrow G(a)$	true	positive	0	
	$\forall x \neg F(x) \rightarrow G(x), \neg F(a) \Rightarrow \neg G(a)$	false	negative	0	
NV2	$\forall x F(x) \rightarrow \neg G(x), F(a) \Rightarrow \neg G(a)$	true	negative	0	
	$\forall x F(x) \rightarrow \neg G(x), F(a) \Rightarrow G(a)$	false	positive	0	
CP1	$\forall x (F(x) \wedge H(x)) \rightarrow G(x), F(a), H(a) \Rightarrow G(a)$	true	positive	0	
	$\forall x (F(x) \wedge H(x)) \rightarrow G(x), F(a), H(a) \Rightarrow \neg G(a)$	false	negative	0	
CP2	$\forall x (F(x) \vee H(x)) \rightarrow G(x), F(a) \Rightarrow G(a)$	true	positive	0	
	$\forall x (F(x) \vee H(x)) \rightarrow G(x), F(a) \Rightarrow \neg G(a)$	false	negative	0	
CP3	$\forall x (F(x) \vee H(x)) \rightarrow G(x), H(a) \Rightarrow G(a)$	true	positive	0	
	$\forall x (F(x) \vee H(x)) \rightarrow G(x), H(a) \Rightarrow \neg G(a)$	false	negative	0	
DM1	$\forall x \neg (F(x) \wedge H(x)) \rightarrow G(x), \neg F(a) \Rightarrow G(a)$	true	positive	0	
	$\forall x \neg (F(x) \wedge H(x)) \rightarrow G(x), \neg F(a) \Rightarrow \neg G(a)$	false	negative	0	
DM2	$\forall x \neg (F(x) \vee H(x)) \rightarrow G(x), \neg F(a), \neg H(a) \Rightarrow G(a)$	true	positive	0	
	$\forall x \neg (F(x) \vee H(x)) \rightarrow G(x), \neg F(a), \neg H(a) \Rightarrow \neg G(a)$	false	negative	0	

Table 6: Logic formulas of generalized modus ponens

Variant	Logic formula	Label	Final neg	atom	# atoms of CC
BS	$\forall x F(x) \rightarrow \neg G(x) \Rightarrow \forall x G(x) \rightarrow \neg F(x)$	true	negative	1	
	$\forall x F(x) \rightarrow \neg G(x) \Rightarrow \forall x G(x) \rightarrow F(x)$	false	positive	1	
NV1	$\forall x \neg F(x) \rightarrow G(x) \Rightarrow \forall x \neg G(x) \rightarrow F(x)$	true	positive	1	
	$\forall x \neg F(x) \rightarrow G(x) \Rightarrow \forall x \neg G(x) \rightarrow \neg F(x)$	false	negative	1	
NV2	$\forall x \neg F(x) \rightarrow \neg G(x) \Rightarrow \forall x G(x) \rightarrow F(x)$	true	positive	1	
	$\forall x \neg F(x) \rightarrow \neg G(x) \Rightarrow \forall x G(x) \rightarrow \neg F(x)$	false	negative	1	
NV3	$\forall x F(x) \rightarrow G(x) \Rightarrow \forall x \neg G(x) \rightarrow \neg F(x)$	true	negative	1	
	$\forall x F(x) \rightarrow G(x) \Rightarrow \forall x \neg G(x) \rightarrow F(x)$	false	positive	1	
CP1	$\forall x (F(x) \wedge H(x)) \rightarrow \neg G(x) \Rightarrow \forall x G(x) \rightarrow \neg (F(x) \wedge H(x))$	true	positive	2	
	$\forall x (F(x) \wedge H(x)) \rightarrow \neg G(x) \Rightarrow \forall x G(x) \rightarrow \neg (F(x) \wedge \neg H(x))$	false	negative	2	
CP2	$\forall x F(x) \rightarrow \neg (G(x) \wedge H(x)) \Rightarrow \forall x (G(x) \wedge H(x)) \rightarrow \neg F(x)$	true	negative	1	
	$\forall x F(x) \rightarrow \neg (G(x) \wedge H(x)) \Rightarrow \forall x (G(x) \wedge H(x)) \rightarrow F(x)$	false	positive	1	
DM1	$\forall x (F(x) \wedge H(x)) \rightarrow \neg G(x) \Rightarrow \forall x G(x) \rightarrow (\neg F(x) \vee \neg H(x))$	true	negative	2	
	$\forall x (F(x) \wedge H(x)) \rightarrow \neg G(x) \Rightarrow \forall x G(x) \rightarrow (\neg F(x) \vee H(x))$	false	positive	2	
DM2	$\forall x F(x) \rightarrow (\neg G(x) \vee \neg H(x)) \Rightarrow \forall x (G(x) \wedge H(x)) \rightarrow \neg F(x)$	true	negative	1	
	$\forall x F(x) \rightarrow (\neg G(x) \vee \neg H(x)) \Rightarrow \forall x (G(x) \wedge H(x)) \rightarrow F(x)$	false	positive	1	

Table 7: Logic formulas of generalized contraposition

Variant	Logic formula	Label	Final neg	atom	# atoms of CC
BS	$\forall x F(x) \rightarrow G(x), \forall x G(x) \rightarrow H(x) \Rightarrow \forall x F(x) \rightarrow H(x)$	true	positive	1	
	$\forall x F(x) \rightarrow G(x), \forall x G(x) \rightarrow H(x) \Rightarrow \forall x F(x) \rightarrow \neg H(x)$	false	negative	1	
NV1	$\forall x \neg F(x) \rightarrow G(x), \forall x G(x) \rightarrow H(x) \Rightarrow \forall x \neg F(x) \rightarrow H(x)$	true	positive	1	
	$\forall x \neg F(x) \rightarrow G(x), \forall x G(x) \rightarrow H(x) \Rightarrow \forall x \neg F(x) \rightarrow \neg H(x)$	false	negative	1	
NV2	$\forall x F(x) \rightarrow \neg G(x), \forall x \neg G(x) \rightarrow H(x) \Rightarrow \forall x F(x) \rightarrow H(x)$	true	positive	1	
	$\forall x F(x) \rightarrow \neg G(x), \forall x \neg G(x) \rightarrow H(x) \Rightarrow \forall x F(x) \rightarrow \neg H(x)$	false	negative	1	
NV3	$\forall x F(x) \rightarrow G(x), \forall x G(x) \rightarrow \neg H(x) \Rightarrow \forall x F(x) \rightarrow \neg H(x)$	true	negative	1	
	$\forall x F(x) \rightarrow G(x), \forall x G(x) \rightarrow \neg H(x) \Rightarrow \forall x F(x) \rightarrow H(x)$	false	positive	1	
CP1	$\forall x (F(x) \wedge G(x)) \rightarrow I(x), \forall x G(x) \rightarrow H(x) \Rightarrow \forall x (F(x) \wedge H(x)) \rightarrow I(x)$	true	positive	1	
	$\forall x (F(x) \wedge G(x)) \rightarrow I(x), \forall x G(x) \rightarrow H(x) \Rightarrow \forall x (F(x) \wedge H(x)) \rightarrow \neg I(x)$	false	negative	1	
CP2	$\forall x F(x) \rightarrow G(x), \forall x G(x) \rightarrow (H(x) \vee I(x)) \Rightarrow \forall x F(x) \rightarrow (H(x) \vee I(x))$	true	positive	2	
	$\forall x F(x) \rightarrow G(x), \forall x G(x) \rightarrow (H(x) \vee I(x)) \Rightarrow \forall x F(x) \rightarrow (H(x) \vee \neg I(x))$	false	negative	2	
CP3	$\forall x F(x) \rightarrow G(x), \forall x F(x) \rightarrow I(x), \forall x (G(x) \wedge I(x)) \rightarrow H(x) \Rightarrow \forall x F(x) \rightarrow H(x)$	true	positive	1	
	$\forall x F(x) \rightarrow G(x), \forall x F(x) \rightarrow I(x), \forall x (G(x) \wedge I(x)) \rightarrow H(x) \Rightarrow \forall x F(x) \rightarrow \neg H(x)$	false	negative	1	
DM1	$\forall x (\neg F(x) \wedge \neg I(x)) \rightarrow G(x), \forall x G(x) \rightarrow H(x) \Rightarrow \forall x \neg (F(x) \vee I(x)) \rightarrow H(x)$	true	positive	1	
	$\forall x (\neg F(x) \wedge \neg I(x)) \rightarrow G(x), \forall x G(x) \rightarrow H(x) \Rightarrow \forall x \neg (F(x) \vee \neg I(x)) \rightarrow \neg H(x)$	false	negative	1	
DM2	$\forall x \neg F(x) \rightarrow G(x), \forall x G(x) \rightarrow (\neg H(x) \vee \neg I(x)) \Rightarrow G(x) \rightarrow \neg (H(x) \wedge I(x))$	true	positive	2	
	$\forall x \neg F(x) \rightarrow G(x), \forall x G(x) \rightarrow (\neg H(x) \vee \neg I(x)) \Rightarrow G(x) \rightarrow (H(x) \wedge I(x))$	false	negative	2	

Table 8: Logic formulas of hypothetical syllogism 1

Variant	Logic formula	Label	Final neg	atom	# atoms of CC
BS	$\forall x F(x) \rightarrow G(x), \forall x \neg H(x) \rightarrow \neg G(x) \Rightarrow \forall x F(x) \rightarrow H(x)$	true	positive	1	
	$\forall x F(x) \rightarrow G(x), \forall x \neg H(x) \rightarrow \neg G(x) \Rightarrow \forall x F(x) \rightarrow \neg H(x)$	false	negative	1	
NV1	$\forall x F(x) \rightarrow G(x), \forall x H(x) \rightarrow \neg G(x) \Rightarrow \forall x F(x) \rightarrow \neg H(x)$	true	negative	1	
	$\forall x F(x) \rightarrow G(x), \forall x H(x) \rightarrow \neg G(x) \Rightarrow \forall x F(x) \rightarrow H(x)$	false	positive	1	
NV2	$\forall x \neg F(x) \rightarrow G(x), \forall x \neg H(x) \rightarrow \neg G(x) \Rightarrow \forall x \neg F(x) \rightarrow H(x)$	true	positive	1	
	$\forall x \neg F(x) \rightarrow G(x), \forall x \neg H(x) \rightarrow \neg G(x) \Rightarrow \forall x \neg F(x) \rightarrow \neg H(x)$	false	negative	1	
NV3	$\forall x F(x) \rightarrow G(x), \forall x \neg H(x) \rightarrow \neg G(x) \Rightarrow \forall x F(x) \rightarrow H(x)$	true	positive	1	
	$\forall x F(x) \rightarrow G(x), \forall x \neg H(x) \rightarrow \neg G(x) \Rightarrow \forall x F(x) \rightarrow \neg H(x)$	false	negative	1	
CP1	$\forall x (F(x) \vee I(x)) \rightarrow \neg G(x), \forall x \neg H(x) \rightarrow \neg G(x) \Rightarrow \forall x (F(x) \vee I(x)) \rightarrow H(x)$	true	positive	1	
	$\forall x (F(x) \vee I(x)) \rightarrow \neg G(x), \forall x \neg H(x) \rightarrow \neg G(x) \Rightarrow \forall x (F(x) \vee I(x)) \rightarrow \neg H(x)$	false	negative	1	
CP2	$\forall x F(x) \rightarrow \neg(G(x) \vee I(x)), \forall x H(x) \rightarrow \neg(G(x) \vee I(x)) \Rightarrow \forall x F(x) \rightarrow H(x)$	true	positive	1	
	$\forall x F(x) \rightarrow \neg(G(x) \vee I(x)), \forall x H(x) \rightarrow \neg(G(x) \vee I(x)) \Rightarrow \forall x F(x) \rightarrow \neg H(x)$	false	negative	1	
CP3	$\forall x \neg F(x) \rightarrow G(x), \forall x \neg F(x) \rightarrow I(x), \neg H(x) \rightarrow \neg(G(x) \wedge I(x)) \Rightarrow \forall x \neg F(x) \rightarrow \neg H(x)$	true	positive	1	
	$\forall x \neg F(x) \rightarrow G(x), \forall x \neg F(x) \rightarrow I(x), \neg H(x) \rightarrow \neg(G(x) \wedge I(x)) \Rightarrow \forall x \neg F(x) \rightarrow \neg \neg H(x)$	false	negative	1	
DM1	$\forall x (\neg F(x) \wedge \neg I(x)) \rightarrow \neg G(x), \forall x \neg H(x) \rightarrow G(x) \Rightarrow \forall x \neg (F(x) \vee I(x)) \rightarrow H(x)$	true	positive	1	
	$\forall x (\neg F(x) \wedge \neg I(x)) \rightarrow \neg G(x), \forall x \neg H(x) \rightarrow G(x) \Rightarrow \forall x \neg (F(x) \vee I(x)) \rightarrow \neg H(x)$	false	negative	1	
DM2	$\forall x F(x) \rightarrow \neg(G(x) \vee I(x)), \forall x H(x) \rightarrow (\neg G(x) \wedge \neg I(x)) \Rightarrow \forall x F(x) \rightarrow H(x)$	true	positive	1	
	$\forall x F(x) \rightarrow \neg(G(x) \vee I(x)), \forall x H(x) \rightarrow (\neg G(x) \wedge \neg I(x)) \Rightarrow \forall x F(x) \rightarrow \neg H(x)$	false	negative	1	
DM3	$\forall x \neg F(x) \rightarrow G(x), \forall x \neg F(x) \rightarrow I(x), \forall x \neg H(x) \rightarrow (\neg G(x) \vee \neg I(x)) \Rightarrow \forall x \neg F(x) \rightarrow H(x)$	true	positive	1	
	$\forall x \neg F(x) \rightarrow G(x), \forall x \neg F(x) \rightarrow I(x), \forall x \neg H(x) \rightarrow (\neg G(x) \vee \neg I(x)) \Rightarrow \forall x \neg F(x) \rightarrow \neg H(x)$	false	negative	1	

Table 9: Logic formulas of hypothetical syllogism 2

Variant	Logic formula	Label	Final neg	atom	# atoms of CC
BS	$\forall x F(x) \rightarrow G(x), \exists x(H(x) \wedge \neg G(x)) \Rightarrow \exists x(H(x) \wedge \neg F(x))$	true	negative	0	
	$\forall x F(x) \rightarrow G(x), \exists x(H(x) \wedge \neg G(x)) \Rightarrow \exists x(H(x) \wedge F(x))$	false	positive	0	
NV1	$\forall x \neg F(x) \rightarrow G(x), \exists x(H(x) \wedge \neg G(x)) \Rightarrow \exists x(H(x) \wedge F(x))$	true	positive	0	
	$\forall x \neg F(x) \rightarrow G(x), \exists x(H(x) \wedge \neg G(x)) \Rightarrow \exists x(H(x) \wedge \neg F(x))$	false	negative	0	
NV2	$\forall x F(x) \rightarrow \neg G(x), \exists x(H(x) \wedge G(x)) \Rightarrow \exists x(H(x) \wedge \neg F(x))$	true	negative	0	
	$\forall x F(x) \rightarrow \neg G(x), \exists x(H(x) \wedge G(x)) \Rightarrow \exists x(H(x) \wedge F(x))$	false	positive	0	
NV3	$\forall x G(x) \rightarrow F(x), \exists x(H(x) \wedge G(x)) \Rightarrow \exists x(H(x) \wedge F(x))$	true	positive	0	
	$\forall x G(x) \rightarrow F(x), \exists x(H(x) \wedge G(x)) \Rightarrow \exists x(H(x) \wedge \neg F(x))$	false	negative	0	
CP1	$\forall x F(x) \rightarrow G(x), \forall x F(x) \rightarrow I(x), \exists x H(x) \wedge \neg(G(x) \wedge I(x)) \Rightarrow \exists x(H(x) \wedge \neg F(x))$	true	negative	0	
	$\forall x F(x) \rightarrow G(x), \forall x F(x) \rightarrow I(x), \exists x H(x) \wedge \neg(G(x) \wedge I(x)) \Rightarrow \exists x(H(x) \wedge F(x))$	false	positive	0	
CP2	$\forall x \neg F(x) \rightarrow G(x), \exists x(H(x) \wedge I(x)) \wedge \neg G(x) \Rightarrow \exists x(H(x) \wedge I(x)) \wedge F(x)$	true	positive	0	
	$\forall x \neg F(x) \rightarrow G(x), \exists x(H(x) \wedge I(x)) \wedge \neg G(x) \Rightarrow \exists x(H(x) \wedge I(x)) \wedge \neg F(x)$	false	negative	0	
CP3	$\forall x(F(x) \wedge I(x)) \rightarrow \neg G(x), \exists x H(x) \wedge G(x) \Rightarrow \exists x H(x) \wedge \neg(F(x) \wedge I(x))$	true	positive	0	
	$\forall x(F(x) \wedge I(x)) \rightarrow \neg G(x), \exists x H(x) \wedge G(x) \Rightarrow \exists x H(x) \wedge \neg(F(x) \wedge \neg I(x))$	false	negative	0	
DM1	$\forall x F(x) \rightarrow G(x), \forall x F(x) \rightarrow I(x), \exists x H(x) \wedge (\neg G(x) \vee \neg I(x)) \Rightarrow \exists x H(x) \wedge \neg F(x)$	true	negative	0	
	$\forall x F(x) \rightarrow G(x), \forall x F(x) \rightarrow I(x), \exists x H(x) \wedge (\neg G(x) \vee \neg I(x)) \Rightarrow \exists x H(x) \wedge F(x)$	false	positive	0	
DM2	$\forall x \neg F(x) \rightarrow G(x), \exists x \neg G(x) \wedge (\neg H(x) \vee \neg I(x)) \Rightarrow \exists x F(x) \wedge \neg(H(x) \wedge I(x))$	true	positive	0	
	$\forall x \neg F(x) \rightarrow G(x), \exists x \neg G(x) \wedge (\neg H(x) \vee \neg I(x)) \Rightarrow \exists x F(x) \wedge \neg(H(x) \wedge \neg I(x))$	false	negative	0	
DM3	$\forall x(F(x) \wedge I(x)) \rightarrow \neg G(x), \exists x H(x) \wedge G(x) \Rightarrow \exists x H(x) \wedge (\neg F(x) \vee \neg I(x))$	true	negative	0	
	$\forall x(F(x) \wedge I(x)) \rightarrow \neg G(x), \exists x H(x) \wedge G(x) \Rightarrow \exists x H(x) \wedge (\neg F(x) \vee I(x))$	false	positive	0	

Table 10: Logic formulas of hypothetical syllogism 3

Variant	Logic formula	Label	Final neg	atom	# atoms of CC
BS	$\forall x F(x) \rightarrow G(x), \neg G(a) \Rightarrow \neg F(a)$	true	negative	0	
	$\forall x F(x) \rightarrow G(x), \neg G(a) \Rightarrow F(a)$	false	positive	0	
NV1	$\forall x F(x) \rightarrow \neg G(x), G(a) \Rightarrow \neg F(a)$	true	negative	0	
	$\forall x F(x) \rightarrow \neg G(x), G(a) \Rightarrow F(a)$	false	positive	0	
NV2	$\forall x \neg F(x) \rightarrow G(x), \neg G(a) \Rightarrow F(a)$	true	positive	0	
	$\forall x \neg F(x) \rightarrow G(x), \neg G(a) \Rightarrow \neg F(a)$	false	negative	0	
CP1	$\forall x F(x) \rightarrow (G(x) \wedge H(x)), \neg G(a) \Rightarrow \neg F(a)$	true	negative	0	
	$\forall x F(x) \rightarrow (G(x) \wedge H(x)), \neg G(a) \Rightarrow F(a)$	false	positive	0	
CP2	$\forall x(F(x) \vee G(x)) \rightarrow H(x), \neg H(a) \Rightarrow \neg(F(a) \vee G(a))$	true	positive	0	
	$\forall x(F(x) \vee G(x)) \rightarrow H(x), \neg H(a) \Rightarrow \neg(F(a) \vee \neg G(a))$	false	negative	0	
CP3	$\forall x F(x) \rightarrow (G(x) \vee H(x)), \neg G(a), \neg H(a) \Rightarrow \neg F(a)$	true	negative	0	
	$\forall x F(x) \rightarrow (G(x) \vee H(x)), \neg G(a), \neg H(a) \Rightarrow F(a)$	false	positive	0	
DM1	$\forall x F(x) \rightarrow (G(x) \wedge H(x)), (\neg G(a) \vee \neg H(a)) \Rightarrow \neg F(a)$	true	negative	0	
	$\forall x F(x) \rightarrow (G(x) \wedge H(x)), (\neg G(a) \vee \neg H(a)) \Rightarrow F(a)$	false	positive	0	
DM2	$\forall x F(x) \rightarrow \neg(G(x) \vee H(x)), G(a) \Rightarrow \neg F(a)$	true	negative	0	
	$\forall x F(x) \rightarrow \neg(G(x) \vee H(x)), G(a) \Rightarrow F(a)$	false	positive	0	

Table 11: Logic formulas of generalized modus tollens

Variant	Logic formula	Label	Final neg	atom	# atoms CC	of
BS	$\forall x F(x) \rightarrow (G(x) \vee H(x)), \forall x F(x) \rightarrow \neg G(x) \Rightarrow \forall x F(x) \rightarrow H(x)$	true	positive		1	
	$\forall x F(x) \rightarrow (G(x) \vee H(x)), \forall x F(x) \rightarrow \neg G(x) \Rightarrow \forall x F(x) \rightarrow \neg H(x)$	false	negative		1	
NV1	$\forall x F(x) \rightarrow (\neg G(x) \vee H(x)), \forall x F(x) \rightarrow G(x) \Rightarrow \forall x F(x) \rightarrow H(x)$	true	positive		1	
	$\forall x F(x) \rightarrow (\neg G(x) \vee H(x)), \forall x F(x) \rightarrow G(x) \Rightarrow \forall x F(x) \rightarrow \neg H(x)$	false	negative		1	
NV2	$\forall x F(x) \rightarrow (G(x) \vee H(x)), \forall x G(x) \rightarrow \neg F(x) \Rightarrow \forall x F(x) \rightarrow H(x)$	true	positive		1	
	$\forall x F(x) \rightarrow (G(x) \vee H(x)), \forall x G(x) \rightarrow \neg F(x) \Rightarrow \forall x F(x) \rightarrow \neg H(x)$	false	negative		1	
NV3	$\forall x F(x) \rightarrow (G(x) \vee \neg H(x)), \forall x F(x) \rightarrow \neg G(x) \Rightarrow \forall x F(x) \rightarrow \neg H(x)$	true	negative		1	
	$\forall x F(x) \rightarrow (G(x) \vee \neg H(x)), \forall x F(x) \rightarrow \neg G(x) \Rightarrow \forall x F(x) \rightarrow H(x)$	false	positive		1	
CP1	$\forall x F(x) \rightarrow (G(x) \vee H(x) \vee I(x)), \forall x F(x) \rightarrow \neg G(x), \forall x F(x) \rightarrow \neg I(x) \Rightarrow \forall x F(x) \rightarrow H(x)$	true	positive		1	
	$\forall x F(x) \rightarrow (G(x) \vee H(x) \vee I(x)), \forall x F(x) \rightarrow \neg G(x), \forall x F(x) \rightarrow \neg I(x) \Rightarrow \forall x F(x) \rightarrow \neg H(x)$	false	negative		1	
CP2	$\forall x (F(x) \wedge I(x)) \rightarrow (G(x) \vee H(x)), \forall x G(x) \rightarrow \neg (F(x) \wedge I(x)), \Rightarrow \forall x (F(x) \wedge I(x)) \rightarrow H(x)$	true	positive		1	
	$\forall x (F(x) \wedge I(x)) \rightarrow (G(x) \vee H(x)), \forall x G(x) \rightarrow \neg (F(x) \wedge I(x)), \Rightarrow \forall x (F(x) \wedge I(x)) \rightarrow \neg H(x)$	false	negative		1	
CP3	$\forall x F(x) \rightarrow (G(x) \vee H(x) \vee I(x)), \forall x F(x) \rightarrow \neg G(x) \Rightarrow \forall x F(x) \rightarrow (H(x) \vee I(x))$	true	positive		2	
	$\forall x F(x) \rightarrow (G(x) \vee H(x) \vee I(x)), \forall x F(x) \rightarrow \neg G(x) \Rightarrow \forall x F(x) \rightarrow (H(x) \vee \neg I(x))$	false	negative		2	
DM1	$\forall x (F(x) \wedge I(x)) \rightarrow (G(x) \vee H(x)), \forall x G(x) \rightarrow (\neg F(x) \vee \neg I(x)) \Rightarrow \forall x (F(x) \wedge I(x)) \rightarrow H(x)$	true	positive		1	
	$\forall x (F(x) \wedge I(x)) \rightarrow (G(x) \vee H(x)), \forall x G(x) \rightarrow (\neg F(x) \vee \neg I(x)) \Rightarrow \forall x (F(x) \wedge I(x)) \rightarrow \neg H(x)$	false	negative		1	
DM2	$\forall x (\neg F(x) \wedge \neg I(x)) \rightarrow (G(x) \vee H(x)), \forall x G(x) \rightarrow (F(x) \vee I(x)) \Rightarrow \forall x \neg (F(x) \vee I(x)) \rightarrow H(x)$	true	positive		1	
	$\forall x (\neg F(x) \wedge \neg I(x)) \rightarrow (G(x) \vee H(x)), \forall x G(x) \rightarrow (F(x) \vee I(x)) \Rightarrow \forall x \neg (F(x) \vee I(x)) \rightarrow \neg H(x)$	false	negative		1	

Table 12: Logic formulas of disjunctive syllogism

Variant	Logic formula	Label	Final neg	atom	# atoms of CC
BS	$\forall x F(x) \rightarrow (G(x) \vee H(x)), \forall x G(x) \rightarrow J(x), \forall x H(x) \rightarrow J(x) \Rightarrow$	true	positive	1	
	$\forall x F(x) \rightarrow J(x)$	false	negative	1	
NV1	$\forall x F(x) \rightarrow (\neg G(x) \vee \neg H(x)), \forall x \neg G(x) \rightarrow J(x), \forall x \neg H(x) \rightarrow$	true	positive	1	
	$J(x) \Rightarrow \forall x F(x) \rightarrow J(x)$	false	negative	1	
NV2	$\forall x F(x) \rightarrow (G(x) \vee H(x)), \forall x G(x) \rightarrow \neg J(x), \forall x H(x) \rightarrow \neg J(x) \Rightarrow$	true	negative	1	
	$\forall x F(x) \rightarrow \neg J(x)$	false	positive	1	
NV3	$\forall x F(x) \rightarrow (G(x) \vee H(x)), \forall x J(x) \rightarrow \neg G(x), \forall x J(x) \rightarrow \neg H(x) \Rightarrow$	true	negative	1	
	$\forall x F(x) \rightarrow \neg J(x)$	false	positive	1	
CP1	$\forall x F(x) \rightarrow (G(x) \vee H(x) \vee I(x)), \forall x G(x) \rightarrow J(x), \forall x H(x) \rightarrow$	true	positive	1	
	$J(x), \forall x I(x) \rightarrow J(x) \Rightarrow \forall x F(x) \rightarrow J(x)$	false	negative	1	
CP2	$\forall x (F(x) \wedge I(x)) \rightarrow (G(x) \vee H(x)), \forall x G(x) \rightarrow J(x), \forall x H(x) \rightarrow$	true	positive	1	
	$J(x) \Rightarrow \forall x (F(x) \wedge I(x)) \rightarrow J(x)$	false	negative	1	
CP3	$\forall x F(x) \rightarrow (G(x) \vee H(x) \vee I(x)), \forall x G(x) \rightarrow J(x), \forall x H(x) \rightarrow$	true	positive	2	
	$J(x) \Rightarrow \forall x F(x) \rightarrow (J(x) \vee I(x))$	false	negative	2	
DM1	$\forall x F(x) \rightarrow \neg(G(x) \wedge H(x)), \forall x \neg G(x) \rightarrow J(x), \forall x \neg H(x) \rightarrow J(x) \Rightarrow$	true	positive	1	
	$\forall x F(x) \rightarrow J(x)$	false	negative	1	
DM2	$\forall x (\neg F(x) \wedge \neg I(x)) \rightarrow (G(x) \vee H(x)), \forall x G(x) \rightarrow J(x), \forall x H(x) \rightarrow$	true	positive	2	
	$J(x) \Rightarrow \forall x \neg J(x) \rightarrow (F(x) \vee I(x))$	false	negative	2	

Table 13: Logic formulas of generalized dilemma

Model	Naive	CoT	CoTwithformula	Prover9	Z3
GPT-4o-mini	0.759 ± 0.011	0.854 ± 0.009	0.819 ± 0.010	0.516 ± 0.013	0.563 ± 0.013
GPT-4o	0.800 ± 0.011	0.896 ± 0.008	0.863 ± 0.009	$0.973 \pm 0.004^\dagger$	$0.937 \pm 0.006^\dagger$
GPT-5	0.990 ± 0.003	0.989 ± 0.003	0.990 ± 0.003	0.982 ± 0.004	0.975 ± 0.004
Gemini-25-flash	0.980 ± 0.004	0.977 ± 0.004	0.966 ± 0.005	0.992 ± 0.002	0.985 ± 0.003
Qwen3	0.982 ± 0.008	0.975 ± 0.009	0.982 ± 0.008	0.739 ± 0.026	0.680 ± 0.028

Table 14: Overall accuracy in AAC.

Positive final atom					
Model	Naive	CoT	CoTwithformula	Prover9	Z3
GPT-4o-mini	0.766 ± 0.012	0.921 ± 0.010	0.876 ± 0.013	0.952 ± 0.009	0.939 ± 0.009
GPT-4o	0.968 ± 0.005	0.982 ± 0.006	0.958 ± 0.009	0.987 ± 0.005	0.955 ± 0.008
GPT-5	0.997 ± 0.002	0.992 ± 0.004	0.995 ± 0.003	0.992 ± 0.004	0.995 ± 0.003
Gemini-25-flash	0.997 ± 0.003	0.997 ± 0.002	1.000 ± 0.000	0.997 ± 0.002	0.995 ± 0.004
Qwen3	1.000 ± 0.000	1.000 ± 0.000	1.000 ± 0.000	0.986 ± 0.010	0.967 ± 0.000
Negative final atom					
GPT-4o-mini	0.729 ± 0.016	0.816 ± 0.011	0.739 ± 0.013	0.914 ± 0.013	0.899 ± 0.015
GPT-4o	0.639 ± 0.012	0.850 ± 0.010	0.800 ± 0.010	$0.961 \pm 0.006^\dagger$	$0.963 \pm 0.007^\dagger$
GPT-5	0.989 ± 0.005	0.995 ± 0.003	0.995 ± 0.003	0.982 ± 0.005	0.976 ± 0.006
Gemini-25-flash	0.963 ± 0.007	0.966 ± 0.006	0.975 ± 0.007	0.992 ± 0.004	0.995 ± 0.003
Qwen3	0.947 ± 0.013	0.934 ± 0.009	0.947 ± 0.013	0.971 ± 0.000	0.941 ± 0.015

Table 15: Accuracy of formulas with positive or negative final atoms.

Two-atom formulas					
Model	Naive	CoT	CoTwithformula	Prover9	Z3
GPT-4o-mini	0.595 ± 0.034	0.570 ± 0.034	0.575 ± 0.034	0.743 ± 0.044	0.703 ± 0.039
GPT-4o	0.535 ± 0.035	0.545 ± 0.035	0.525 ± 0.035	$0.865 \pm 0.020^\dagger$	$0.860 \pm 0.021^\dagger$
GPT-5	0.935 ± 0.015	0.925 ± 0.015	0.930 ± 0.015	0.910 ± 0.017	0.900 ± 0.018
Gemini-25-flash	0.920 ± 0.016	0.914 ± 0.018	0.949 ± 0.014	0.960 ± 0.013	0.984 ± 0.009
Qwen3	0.875 ± 0.045	0.875 ± 0.041	0.875 ± 0.045	0.950 ± 0.035	0.892 ± 0.049
One-atom formulas					
GPT-4o-mini	0.817 ± 0.018	0.930 ± 0.013	0.910 ± 0.016	0.929 ± 0.020	0.952 ± 0.013
GPT-4o	0.843 ± 0.018	0.970 ± 0.009	0.913 ± 0.015	0.990 ± 0.005	1.000 ± 0.000
GPT-5	1.000 ± 0.000	1.000 ± 0.000	1.000 ± 0.000	1.000 ± 0.000	1.000 ± 0.000
Gemini-25-flash	0.996 ± 0.004	1.000 ± 0.000	1.000 ± 0.000	1.000 ± 0.000	0.997 ± 0.003
Qwen3	1.000 ± 0.000	1.000 ± 0.000	1.000 ± 0.000	1.000 ± 0.000	1.000 ± 0.000

Table 16: Accuracy of two-atom or one-atom formulas in AAC.

Model	Naive	CoT	CoT_Logic	Prover9	Z3
GPT-4o-mini	0.610 ± 0.036	0.615 ± 0.036	0.709 ± 0.034	0.484 ± 0.037	0.335 ± 0.035
GPT-4o	0.665 ± 0.035	0.692 ± 0.035	0.681 ± 0.035	0.681 ± 0.035	0.533 ± 0.037
GPT-5	0.846 ± 0.026	0.835 ± 0.027	0.868 ± 0.025	0.775 ± 0.031	0.747 ± 0.032
Gemini-25-flash	0.852 ± 0.026	0.819 ± 0.029	0.824 ± 0.028	0.791 ± 0.030	0.692 ± 0.034
Qwen3	0.824 ± 0.029	0.786 ± 0.031	0.786 ± 0.030	0.659 ± 0.035	0.599 ± 0.036

Table 17: Overall accuracy in FOLIO.