Enabling Equitable Access to Trustworthy Financial Reasoning

William Jurayj[1], **Nils Holzenberger**[2], **Benjamin Van Durme**[1]
[1]Johns Hopkins University
[2]Télécom Paris, Institut Polytechnique de Paris
wjurayj1@jhu.edu

Abstract

According to the United States Internal Revenue Service, "the average American spends $270 and 13 hours filing their taxes". Even beyond the U.S., tax filing requires complex reasoning, combining application of overlapping rules with numerical calculations. Because errors can incur costly penalties, any automated system must deliver high accuracy and auditability, making modern large language models (LLMs) poorly suited for this task. We propose an approach that integrates LLMs with a symbolic solver to calculate tax obligations. We evaluate variants of this system on the challenging StAtutory Reasoning Assessment (SARA) dataset, and include a novel method for estimating the cost of deploying such a system based on real-world penalties for tax errors. We further show how combining up-front translation of plain-text rules into formal logic programs, combined with intelligently retrieved exemplars for formal case representations, can dramatically improve performance on this task and reduce costs to well below real-world averages. Our results demonstrate the promise and economic feasibility of neuro-symbolic architectures for increasing equitable access to reliable tax assistance.

Introduction

"GPT is not a certified tax professional, nor am I, so you should always check with your tax advisor."

— Greg Brockman, CTO of OpenAI

Of life's two certainties, taxes should be preferred; yet they may well be the more complicated one. Each year, virtually every adult in the world must calculate and pay a fee to some government, in order to reside and earn a living within the state's guardianship. Even for individuals with relatively simple financial situations, the annual filing process demands meticulous reading and following of dozens of form instructions and the copying of values across schedules, worksheets, and eligibility tests. Completing these tasks without professional assistance can take hours. Alternatively, taxpayers may hire a professional preparer, incurring substantial fees depending on the complexity of their return (Internal Revenue Service 2025).

Accuracy in tax filing is essential. Over-reported income or missed deduction opportunities lead to unnecessary overpayment, while under-reporting may result in penalties, interest, and potential legal consequences. In the United States, the costs of inaccuracies affect lower income communities more significantly, in part because these groups offer the Internal Revenue Service (IRS) high audit success rates (Black et al. 2022; Elzayn et al. 2025). However, these audits deliver a modest return on investment compared to audits of wealthier taxpayers, so there is an opportunity to better align community and institutional interests through improved tax advice to lower income taxpayers (Boning et al. 2024).

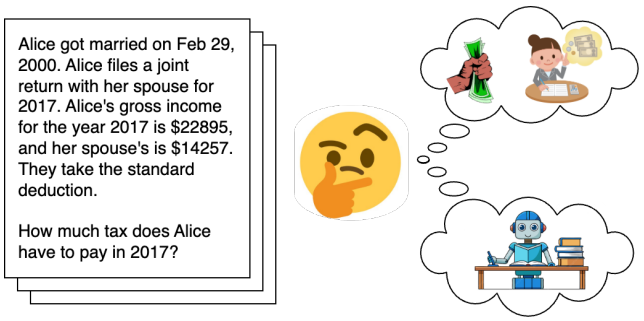

Figure 1: A taxpayer confronted with a tax question might choose between an inexpensive AI preparer and a costlier human professional. The decision considers trade-offs between cost, convenience, and confidence in the result.

However, the concrete costs for errors present a substantial challenge for modern large language models (LLMs). An AI assistant deployed in this domain must meet higher standards than basic accuracy: it should (1) recognize when it lacks sufficient certainty to offer guidance and (2) generate a transparent and faithful trace of logical steps so that taxpayers and auditors can easily verify the derivation of each answer. In this paper we show that symbolic reasoning tools, integrated with LLMs, offer a promising approach to meeting these standards. Our method provides the language model with access to a symbolic solver, enabling it to translate statutory text and taxpayer information into formal logic programs, which are processed by a trusted execution engine. We evaluate the method on the StAtutory Reasoning Assessment (SARA) dataset, a benchmark of synthetic tax scenarios paired with liability calculations carried out through ground-truth representations of rules and facts in formal logic (Holzenberger et al. 2020).

Our experiments demonstrate two key findings. First,

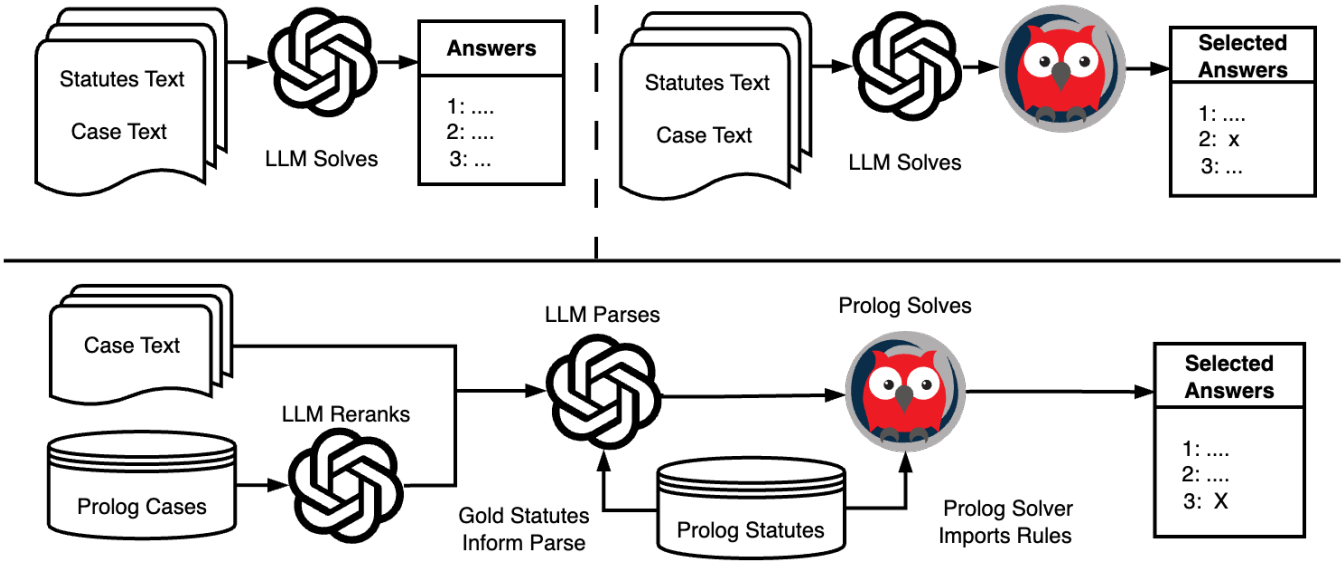

Figure 2: **Methods for solving. Top Left:** Plain-text for statutes and a case is fed into a language model, along with the instruction to calculate a person's tax obligation. **Top Right:** Statutes and a case are fed into the model as before, but it is instructed to convert these into a logic program which calculates a person's tax obligation. If the SWI-Prolog engine fails to execute the program, the case is considered unanswered. **Bottom:** A language model parses a case's facts into Prolog, conditioned on gold parses of the most relevant cases and of the rules contained in the statutes. The symbolic solver imports the gold parses of the statutes before attempting to execute the generated parse of the case. Note that unlike the approaches above it, this requires gold symbolic representations of both the statutes and a representative selection of correctly-decided cases.

whereas frontier reasoning models outperform non-reasoning models at both directly solving and at parsing case and statute text into the symbolic solver, non-reasoning models consistently outperform their reasoning counterparts when given gold symbolic representations of statutes and of their application to similar cases. Second, we show that by adding additional refusal criteria through a symbolic solver and self-checking, the expected costs of deploying such a system in the real world could be brought down to less than 20% of the average cost for an American to file their taxes. Our results indicate the promise of neuro-symbolic architectures for expanding access to trustworthy and reliable tax expertise.

Background

Logic Programming for Legal Reasoning

Several programming languages have been designed to represent and facilitate logical reasoning. Prolog is a declarative programming language for representing and reasoning over knowledge, with roots in first order logic. A programmer defines rules using Horn clauses (Horn 1951) and facts by declaring which rules apply to entities, thus populating a knowledge base. Subsequently, this knowledge base can be queried by defining a 'goal', which launches computation in the form of a backward-chaining search attempting to prove that the goal holds a certain value (Wielemaker et al. 2010). Prolog has been used since the early days of legal AI, where it has formed the backbone of legal expert systems because its declarative syntax keeps knowledge base entries for rules human-readable while powerfully representing the reasoning around which legal questions revolve (Sherman 1989). Efforts in countries like the United Kingdom (Sergot et al. 1986), Canada (Sherman 1987), and the United States (Kant et al. 2025) have leveraged this capacity to encode legal rules in executable formal logic. Related languages have also been used in the legal domain, such as Answer-set Programming (Gelfond and Lifschitz 1988; Morris 2020), Datalog (Ceri et al. 1989; Huang et al. 2021), and Catala (Merigoux et al. 2021a; Merigoux 2023). More broadly, hierarchical templates are a popular tool for evaluation of legal reasoning (Hou et al. 2024). We focus on the SARA dataset (Holzenberger et al. 2020), which encodes statutes and cases into Prolog logic programs to show how a symbolic expert system can perfectly solve a task which large language models struggle to complete.

Statutory Tax Reasoning and the SARA Dataset

We focus on the task of statutory reasoning for tax law. Some elements of this task bear similarity to popular mathematical reasoning tasks such as GSM-8k (Cobbe et al. 2021) or MATH-500 (Hendrycks et al. 2021), such as chaining together mathematical operations to solve a real-world problem described in words. However, unlike these math datasets which require application of a small set of universal arithmetic rules which models learn during training, statutory reasoning considers a set of contingent rules contained within documents provided to a model in-context at inference time, in addition to these basic arithmetic principles.

We evaluate our methods on the SARA dataset, which

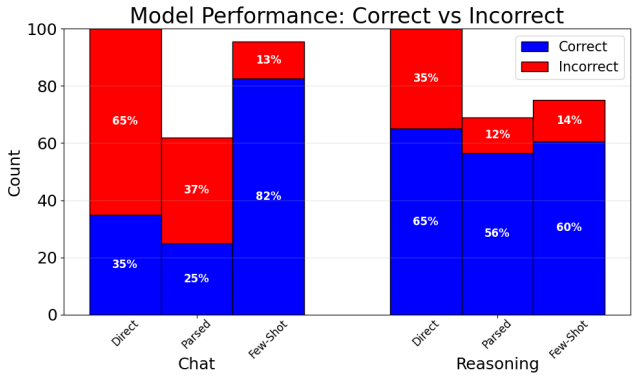

Figure 3: Number of correct and incorrect solutions produced by each solution method, for large chat- and reasoning-optimized models (served by DeepSeek and OpenAI).

tests the ability of language models to do statutory reasoning about the United States Tax Code (Holzenberger et al. 2020). This dataset is included in the popular aggregate benchmark LegalBench (Guha et al. 2023), and was used in the GPT-4 product launch to highlight the model's superior reasoning capacity (Blair-Stanek et al. 2024). The SARA dataset consists of 9 sections from the US federal tax code which have been moderately edited to make them self-contained and unambiguous. These manipulations allow the dataset to serve as a self-contained and solvable task for language models that lack live internet access and human-like abilities to process ambiguity (Jurayj et al. 2022; Stengel-Eskin et al. 2024). These statute sections are accompanied by 376 hand-crafted cases to test understanding of these statutes, each containing a question about a person's tax obligation. Each statute and case has been manually translated into Prolog, which allows them to be trivially solved using the language's powerful execution engine to resolve queries about cases. This Prolog is defined using neo-Davidsonian event semantics (Davidson 1966), thus categorizing each event as one of 61 possible predicates onto which various arguments are attached. Of the 376 cases and corresponding questions, 276 require binary judgments about whether a statute applies to a given case, and 100 require numerical judgments about how much tax a person owes in a given year. We focus on these 100 tax cases because of their increased difficulty, and because a trivial baseline of always guessing a single answer delivers poor performance at predicting the numerical output.

Zero-Shot Solving from Statutory Text

Direct Calculation

We evaluate our methods against the baseline method of direct solving, mirroring the approach used in OpenAI's GPT-4 demonstration (Blair-Stanek et al. 2024). This approach treats the tax calculation as a mathematical question answering task with the additional demand that the model must apply entries from a large corpus of rules contained in the statutory text, in addition to the generic arithmetic rules that govern all calculation. For each case in this setting, a model's context is filled with all sections of the statutes concatenated together, the description of the case's facts, and the question about a person in that case's tax liability, all in plain text. It is instructed to calculate the person in question's tax obligation based on the rules outlined in the statutes.

Chat Model	Reasoning Model	Size
Qwen2.5-Coder	R1-Distill Qwen2.5	32 billion
Llama 3.3	R1-Distill Llama 3.3	70 billion
DeepSeek-V3	DeepSeek-R1	671 billion
GPT-4.1	OpenAI o3	$8/m tokens

Table 1: **Chat and Reasoning model pairs.** Each open-weight model pair is fine-tuned from the same base model. Although the exact dimensions and provenance of OpenAI's models are unknown, the two models have identical token pricing structures, suggesting that they incur similar costs for OpenAI to serve.

Calculation by Parsing for a Symbolic Solver

To extend the direct solution approach, we augment the language model with a symbolic solver. Here, a model is given the plain text of the statutes as in the direct calculation case. It is instructed to generate a Prolog program which encodes the relevant rules and facts necessary to compute the person in question's tax obligation. The symbolic solver ingests a set of rules and facts in Prolog, and is invoked to execute a query. The execution of this Prolog program offers a straightforward mechanism for refusal: if the program fails to execute into the proper format or hangs beyond a preallocated time limit (10 seconds), the system is considered to have refused to answer.

Experimental Setup

We run experiments across four model families of different sizes, three of which are open-weight models. The bases for these models are: Qwen2.5 32B (Qwen et al. 2025), Llama 3.3 70B (Grattafiori et al. 2024), DeepSeek-V3 671B (DeepSeek-AI et al. 2025b), and OpenAI's GPT-4.1 (OpenAI 2025a,b), each of which has an instruction-tuned version designed for common chat applications, and a reasoning version optimized to expend additional inference-time compute to solve harder problems. The full list of models is included in table 1. We run auxiliary experiments using GPT-5, but we do not conduct the same chat vs. reasoning comparison because this product appears to be an integrated system containing several models, and therefore is not as analogous to these other comparisons (Zhang et al. 2025). The reasoning model for three of these pairings stem from the DeepSeek R1 project (DeepSeek-AI et al. 2025a), where strong base models were fine-tuned to generate long chains-of-thought that help them solve harder quantitative reasoning problems. Although there is no formal documentation stating that OpenAI's GPT-4.1 and o3 models are derived from the same base model, the proximity of the two model's launch dates and their identical per-token pricing schemes

Model Family	Model	Method	Correct	Incorrect	Abstentions	Break-Even Price
Baseline	N/A	Always Abstain	0	0	100	$270 ± 0
	N/A	Always Predict $0	5	95	0	$16227.11 ± 7805.94
Qwen-2.5	Qwen-32b	Direct	13	87	0	$3,051.64 ± 1,828.31
	Qwen-32b	Parsed	2	17	81	$490.34 ± 230.75
	R1-32b	Direct	38	62	0	$505.25 ± 287.67
	R1-32b	Parsed	1	2	97	$278.70 ± 24.33
Llama-3.3	Llama-70b	Direct	9	91	0	$1,065.90 ± 675.07
	Llama-70b	Parsed	1	43	56	$252,027.73 ± 414,049.97
	R1-70b	Direct	43	57	0	$1,257.03 ± 1,620.47
	R1-70b	Parsed	2	1	97	$266.10 ± 6.81
DeepSeek	DeepSeek-V3	Direct	22	78	0	$739.45 ± 474.59
	DeepSeek-V3	Parsed	11	43	46	$2,099.13 ± 1,253.57
	DeepSeek-V3	Direct + Direct	16	15	69	$265.46 ± 63.53
	DeepSeek-V3	Direct + Parsed	7	4	89	$285.53 ± 55.57
	DeepSeek-V3	Parsed + Parsed	5	8	87	$310.47 ± 67.95
	DeepSeek-R1	Direct	74	26	0	$304.29 ± 225.57
	DeepSeek-R1	Parsed	38	10	52	$249.64 ± 84.77
	DeepSeek-R1	Direct + Direct	66	12	22	$94.20 ± 59.76
	DeepSeek-R1	Direct + Parsed	34	3	63	$170.10 ± 21.75
	DeepSeek-R1	Parsed + Parsed	17	4	79	$241.80 ± 29.45
OpenAI GPT-4.1	GPT-4.1	Direct	48	52	0	$532.84 ± 492.99
	GPT-4.1	Parsed	39	31	30	$228.89 ± 151.69
	GPT-4.1	Direct + Direct	42	13	45	$196.92 ± 88.43
	GPT-4.1	Direct + Parsed	27	6	67	$185.10 ± 21.33
	GPT-4.1	Parsed + Parsed	26	5	69	$186.30 ± 20.84
	o3	Direct	56	44	0	$6,431.84 ± 2,637.94
	o3	Parsed	75	15	10	$47.43 ± 22.16
	o3	Direct + Direct	41	17	42	$3,472.29 ± 1,859.32
	o3	Direct + Parsed	52	10	38	$115.90 ± 24.63
	o3	Parsed + Parsed	65	9	26	$77.51 ± 22.41
OpenAI GPT-5	GPT-5	Direct	76	24	0	$299.11 ± 288.41
	GPT-5	Parsed	53	13	34	$122.72 ± 29.21
	GPT-5	Direct + Direct	73	9	18	$218.64 ± 270.19
	GPT-5	Direct + Parsed	46	6	48	$138.30 ± 25.53
	GPT-5	Parsed + Parsed	31	5	64	$180.23 ± 23.42

Table 2: **Results of different methods without gold statutes.** Models in the same family have the same base model (or seem most likely to, in the case of closed-weights models). Note that "break-even price" measures only the costs of failures and abstentions, and does not include inference costs. For each model, the approach that delivered the lowest break-even price is shown in bold. The top two rows show the break-even price of trivial systems, which always defer to an expert or which always tell a person not to pay any taxes. Errors represent a 90% confidence interval. The lowest break-even price method for each model family is in bold.

suggest that these two models have a similar relationship to the other model pairs we explore.

We consider 'correct' attempts to be those where the output calculated by our system is exactly the same as the actual tax obligation, when rounded to the nearest dollar. All Prolog code is executed using the SWI-Prolog (Wielemaker et al. 2010) implementation of Prolog, and externally halted after 10 seconds of reasoning. We run experiments on the v2 release of the SARA dataset, because its programmatic representations most closely match the natural language surface forms (Holzenberger and Van Durme 2021).

Self-Consistency Tests

We further ask how effectively these methods can serve to improve each other's selectivity, by using comparisons between different solution methods to expend additional compute to help determine whether an answer should be trusted (Wang et al. 2023; Stengel-Eskin and Van Durme 2023; Jurayj et al. 2025). In these settings, an answer is only accepted if it is reached via two independent reasoning processes: two chains-of-thought and answers (either directly calculated obligations or Prolog programs) are sampled from the same model. When self-checking using the same method (for instance, "Parsed + Parsed"), these answers are conditioned on the same prompt and context. In the parsing-based