LLM Benchmarking Framework for Description Logic Reasoning

Phanphum Prathumsuwan
Mahidol University
phanphum.pra@student.mahidol.ac.th

August 19, 2025

Abstract

This paper presents a novel benchmarking framework to evaluate the Description Logic (DL) reasoning capabilities of Large Language Models (LLMs). While prior work has tested LLMs on symbolic reasoning and natural language inference[23, 13], few have focused on the structured semantics of DL, particularly the EL and ELH fragments widely used in ontology engineering[5]. The framework extracts EL axioms from the Pizza ontology[21, 14] and extends them with role inclusion to create ELH-compliant cases, thereby increasing DL complexity. It then generates corresponding natural language queries, and compares model responses to formal entailment outcomes using the OWL API[15] and HermiT reasoner[18, 12]. We evaluate seven LLMs, including GPT-4o[19], Gemini 2.5 Pro[10], LLaMA 3[3], and Gemma 2[11], Mistral[4], Qwen[1], and DeepSeek Coder[2] and analyze their accuracy, behavior under increasing DL complexity (moving from EL to ELH), and stability. The results reveal significant differences in model capabilities, with top-tier models like GPT-4o, Gemma 2, and Gemini 2.5 Pro achieved accuracy at or near 100(%) with high consistency. In contrast, other models like LLaMA 3 exhibited significant performance variability and struggled with the increased logical complexity of ELH. This initial framework and its findings lay the groundwork for future investigations into symbolic reasoning capabilities of LLMs, including the application of formal metamorphic testing[20, 8].

1 Introduction

Description Logics (DLs) form the formal foundation of ontologies and semantic web technologies, enabling structured knowledge representation and reasoning[6]. DL-based reasoning is widely used in biomedical informatics[7], intelligent systems, and knowledge graphs[16], where tasks such as concept classification, consistency checking, and entailment are critical. Traditional DL reasoners such as HermiT[12], FaCT++[24], and ELK[17] have provided robust symbolic reasoning under well-defined semantics. However, these systems often struggle with scalability and interpretability when integrated into real-world, language-centric AI applications.

Recent advances in Large Language Models (LLMs) such as GPT-4, Gemini, and LLaMA have demonstrated surprising capabilities in tasks requiring logical inference, commonsense reasoning, and formal language understanding. This has prompted growing interest in evaluating whether LLMs can perform reasoning tasks over ontological structures. Despite the enthusiasm, existing benchmarks fall short in rigorously assessing LLMs' abilities to perform *formal DL reasoning*, particularly in the lightweight but expressive EL and ELH fragments commonly used in biomedical ontologies[5].

To address this gap, we propose a novel **LLM Benchmarking Framework for Description Logic Reasoning**, designed to evaluate the reasoning capability of LLMs across a suite of logically grounded test cases. The framework currently validates reasoning correctness against formal DL

entailment outcomes, specifically focusing on axioms expressed in the OWL 2 EL and ELH profiles. It is designed to be extensible, serving as a foundation for future applications of techniques like metamorphic testing, which can validate reasoning correctness under controlled transformations (e.g., paraphrasing, axiom removal) and minimize reliance on single gold-standard labels.

Our main contributions are:

- Introduce a novel framework that generates and evaluates DL reasoning test cases for LLMs, focusing on formal entailment within EL and ELH profiles.
- Develop a suite of logically grounded test cases derived from a real-world ontology, covering varying levels of DL complexity (EL vs. ELH).
- Compare multiple state-of-the-art LLMs, analyzing their reasoning accuracy, consistency, and robustness to structural changes introduced by increasing DL complexity.
- Release an open-source implementation and dataset to support further research on neural-symbolic reasoning and the future integration of advanced testing strategies like metamorphic relations[22].

2 Related Work

2.1 Description Logic Reasoning

Description Logics (DLs) are a family of formal knowledge representation languages that underpin ontologies used in the Semantic Web, notably OWL 2. Standard DL reasoners such as **HermiT**, **FaCT++**, and **ELK** offer sound and complete reasoning services for expressive DLs, including the EL and ELH fragments. These systems perform classification, entailment checking, and consistency testing with well-defined computational guarantees. However, they are rule-based and rely on explicit formal inputs, making them less suited for integration with natural language systems or handling uncertain/incomplete knowledge.

2.2 Large Language Models and Reasoning

Large Language Models (LLMs) such as **GPT-4**, **Gemini**, and **LLaMA** have shown strong performance in various language understanding tasks, including zero-shot and few-shot reasoning. Studies like *ProofWriter* and *Logic-LLaMA* have evaluated LLMs' ability to perform symbolic and deductive reasoning. While these works demonstrate potential, they often focus on synthetic logical datasets or general commonsense knowledge, and rarely assess formal logic entailment in a structured ontology.

2.3 Neural-Symbolic Integration

Efforts to bridge symbolic AI and neural models have explored hybrid approaches combining rule-based logic with deep learning. Examples include **Neural Theorem Provers** and **NeSy** architectures, which attempt to retain the structure of formal logic while leveraging the flexibility of neural networks. These approaches, however, are not directly applicable to LLMs without task-specific training or architectural modifications.

2.4 Benchmarks and Evaluation of LLMs

LLM evaluation has largely focused on language tasks (e.g., MMLU, BIG-Bench) and general reasoning (e.g., GSM8K, StrategyQA). However, these benchmarks do not provide DL-grounded, formally validated test cases. Recent studies like *Metamorphic LLM Evaluation* have proposed using transformation-based testing (MRs) to evaluate LLM robustness and consistency, offering an attractive model-agnostic strategy that we adopt and extend in our work.

2.5 Our Distinction

While prior work has evaluated reasoning ability in LLMs and explored metamorphic testing, none to our knowledge have benchmarked LLMs against *DL entailment and satisfiability* tasks within *OWL 2 EL/ELH fragments*, nor used formal DL axioms and reasoning tasks to do so. Our framework fills this gap by:

- Grounding reasoning tasks in OWL ontologies.
- Focusing on EL/ELH, which are widely used in biomedical ontologies like SNOMED CT and GO.
- Laying the groundwork for evaluating not just answer correctness but also reasoning robustness through future application of metamorphic relations.

3 Framework Design

Our framework is designed to evaluate the reasoning capabilities of LLMs over OWL ontologies by transforming Description Logic axioms into natural language prompts and validating model responses through metamorphic testing. The framework is composed of four main components: (1) axiom preprocessing and grouping, (2) prompt generation, (3) metamorphic transformation and validation, and (4) LLM interaction and output parsing.

3.1 Axiom Preprocessing and Grouping

We begin by selecting ontologies expressed in OWL 2, focusing on fragments conforming to the EL and ELH profiles. Each ontology is parsed using the OWL API, and axioms are extracted and grouped based on their shared subject or concept. For example, a class like `American` may be associated with a set of axioms such as:

- $\texttt{American} \sqsubseteq \texttt{NamedPizza}$
- $\texttt{American} \sqsubseteq \exists \texttt{hasTopping.MozzarellaTopping}$
- $\texttt{American} \sqsubseteq \exists \texttt{hasTopping.TomatoTopping}$

These are grouped to reflect a localized reasoning context.

3.2 Prompt Generation

Each axiom group is translated into symbolic and natural language formats. Symbolic forms are preserved for clarity and reproducibility, while natural language prompts are used as input to LLMs. A prompt might read:

> *Given that an American pizza has Mozzarella and Tomato as toppings and is a NamedPizza, is it true that an American pizza must have Mozzarella?*

Query axioms (e.g., entailments or satisfiability targets) are automatically identified based on structural heuristics such as the presence of `\exists r.C` patterns or subclass relationships.

3.3 Evaluating Logical Complexity and Future Work

In our current evaluation, we test robustness by increasing logical complexity from the EL to the ELH profile. The framework, however, is designed to be extensible for more sophisticated Metamorphic Relations (MRs) in future work, such as:

- **MR-0**: No change — baseline reasoning test
- **MR-1**: Paraphrased prompt — tests linguistic robustness
- **MR-9**: Removal of key axiom — checks logical dependence

In the current work, we generate a transformed version of the prompt primarily by varying the underlying DL profile (EL vs. ELH) and compare the model's response for consistency with formal entailment.

3.4 LLM Execution and Output Validation

Prompts are submitted to LLMs (e.g., GPT-4, Gemini, LLaMA) using their respective APIs. Model outputs are parsed to extract entailment judgments (e.g., "yes", "no", or equivalent rewordings). For each test case, we track accuracy, consistency across MRs, and failure modes. This allows us to quantify not just correctness, but reasoning stability.

4 Methodology

This work proposes a modular framework for benchmarking LLMs on Description Logic (DL) reasoning tasks using axioms from a real-world OWL ontology. The benchmark focuses on two fragments of DL: EL and ELH (EL with role hierarchies). The pipeline is designed to compare natural language reasoning of LLMs with symbolic entailment results from a formal reasoner.

4.1 Pipeline Overview

Here is a visual representation of our framework pipeline:

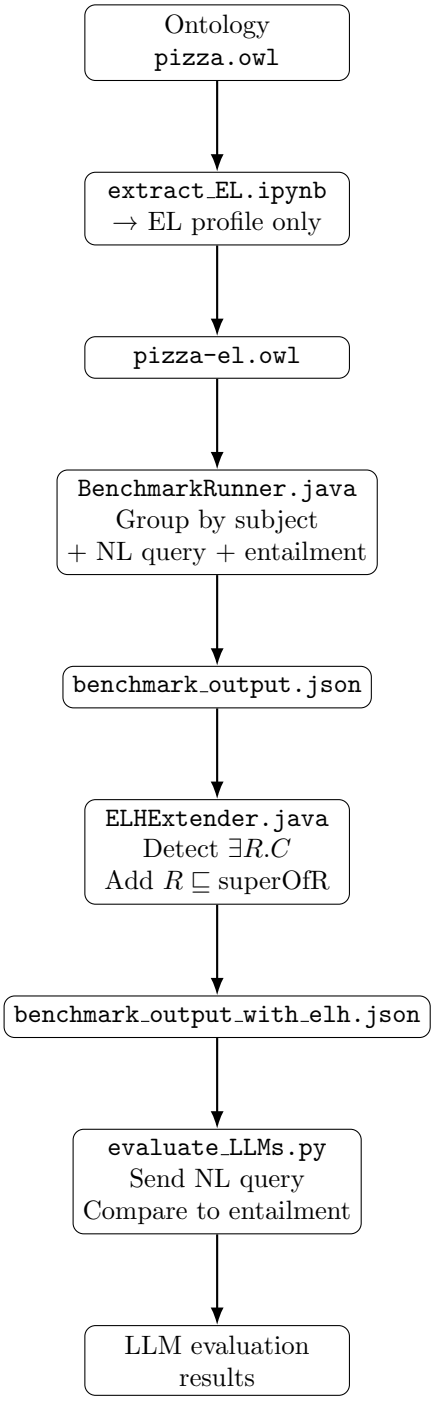

Figure 1: Overview of the DL Reasoning Benchmarking Framework Pipeline

The pipeline consists of four stages:

1. **EL Ontology Extraction:** A subset of axioms conforming to the EL profile is extracted from the Pizza ontology using `extract_EL.ipynb`. The output is an EL-compliant OWL file.
2. **Benchmark Generation:** Java programs parse the EL ontology to group axioms by subject (`AxiomGrouper.java`), generate corresponding natural language queries (`QueryGenerator.java`), and compute ground-truth entailment using the HermiT reasoner (`ReasoningValidator.java`). These are saved in `benchmark_output.json`.
3. **ELH Extension:** The benchmark is enriched with role inclusion axioms to form ELH cases using `ELHExtender.java`, resulting in `benchmark_output_with_elh.json`.
4. **LLM Evaluation:** A Python script sends natural language queries to various LLMs (GPT-4o, Gemini, Gemma, LLaMA, etc.) using their APIs and compares their Yes/No predictions to ground-truth entailment.

4.2 DL Profiles and Scope

The benchmark supports:

- **EL:** A subset of DL supporting intersection and existential quantification
- **ELH:** EL extended with role hierarchy axioms (e.g., $r \sqsubseteq s$)

At this stage, no additional MR (metamorphic relation) types have been applied beyond the original EL/ELH logical transformations.

4.3 Evaluation Metrics

To ensure robust evaluation, each model was tested on the entire benchmark three times. Model performance is measured using the following metrics based on the aggregated results of these runs:

- **Mean Accuracy:** The primary metric is the average accuracy across the three runs, representing the model's typical reasoning performance.
- **Consistency (Standard Deviation):** We report the standard deviation (σ) of the accuracy across the three runs. A low standard deviation indicates high consistency and reliability, while a high value suggests erratic performance.
- **Output Format Stability:** We track the number of correct responses that were not in the expected "Yes" or "No" format and required manual validation (e.g., answers embedded in a full sentence). This serves as a qualitative measure of the model's ability to follow instructions.
- **Execution Time and Errors:** We measure the total execution time and log any API failures or server errors encountered during the evaluation.

5 Results and Analysis

We evaluated seven large language models on a benchmark dataset derived from the Pizza ontology. To ensure reliability, each model was evaluated three times. The aggregated results, covering accuracy, consistency, error patterns, and runtime performance, are presented in Table 1.

5.1 Overall Model Performance

Table 1: Model performance on EL and ELH. Accuracy values are reported as Mean ± Standard Deviation over three runs. Error counts are averages per run (ELH).

Model	EL (%)	ELH (%)	FN	FP	Time	API
GPT-4o	**100.0 ± 0.0**	**100.0 ± 0.0**	0	0	45s	0
Gemma 2 (4.3B Instruct)	**100.0 ± 0.0**	99.43 ± 0.99	0.33	0	22s	0
Gemini 2.5 Pro	98.28 ± 0.0	**100.0 ± 0.0**	0.33	0	10+ min	2
Mistral (7B Instruct)	92.53 ± 2.65	96.55 ± 1.73	2	0	35s	0
LLaMA 3 (8B Instruct)	83.33 ± 3.56	79.31 ± 4.49	12	0	30s	0
Qwen 7B	41.95 ± 2.00	30.46 ± 5.45	39	0	30s	0
DeepSeek Coder (6.7B Instruct)	14.94 ± 7.15	17.82 ± 9.81	45	0	4.5 min	0

EL = EL Accuracy; ELH = ELH Accuracy; FN = Avg. False Negatives; FP = Avg. False Positives; Time = wall-clock per run; API = API errors per run. Results averaged over three runs.

5.2 Analysis of Results

The results in Table 1 reveal a clear hierarchy of reasoning capabilities.

Accuracy and Consistency: Top-tier models like **GPT-4o**, **Gemma 2**, and **Gemini 2.5 Pro** not only achieved the highest accuracy but also demonstrated exceptional consistency, with standard deviations at or near zero. In contrast, models like **LLaMA 3** and particularly **DeepSeek Coder** showed significant performance variability between runs, indicating a lack of stable reasoning processes.

Error Analysis: A crucial finding is the complete absence of **false positives** across all models. This suggests that the models, when they fail, do so by being overly cautious and failing to identify a valid entailment (a false negative), rather than inventing incorrect ones. The number of false negatives correlates directly with lower accuracy, with **LLaMA 3**, **Qwen 7B**, and **DeepSeek Coder** exhibiting a strong tendency to miss valid inferences.

Practical Performance: Practical usability diverges significantly. While highly accurate, **Gemini 2.5 Pro** suffered from extremely long execution times and was the only model to experience API stability issues (500 errors). Most other models offered rapid responses under a minute, with **Gemma 2** providing an excellent balance of top-tier accuracy, high consistency, and the fastest execution time.

5.3 Structural vs. HermiT Reasoner Runtime Comparison

To establish a ground truth for entailment, we primarily used the well-established **HermiT** reasoner. For validation, we also compared its outputs against the OWL API's built-in **StructuralReasoner**. For all test cases in our EL benchmark, both reasoners produced identical entailment results, confirming the correctness of our ground truth. However, as shown in Table 2, their runtime performance differed substantially. This validates the use of either reasoner for the current scope while justifying HermiT's role in future evaluations involving more complex logics.

Table 2: Average Runtime per Entailment Check (in nanoseconds)

Reasoner	**Min Time (ns)**	**Max Time (ns)**
StructuralReasoner	5,584	84,733
HermiT Reasoner	118,958	659,709

The **StructuralReasoner**, which uses syntactic approximation and supports tractable fragments like EL, consistently achieved inference times under 20 microseconds. In contrast, **HermiT**, a complete tableau-based reasoner, required up to 660 microseconds due to its support for more expressive DLs and complete reasoning.

Despite the significant runtime gap, both reasoners agreed on all entailments in the EL test set. This agreement stems from the simplicity of the benchmark cases, which strictly conform to the OWL 2 EL profile—a fragment designed for efficient reasoning and fully supported by both systems.

This validates the use of StructuralReasoner for efficient EL benchmarking while justifying HermiT's role in future evaluations involving more complex logics (e.g., ELH, ALC) or structural transformations under metamorphic testing.

6 Conclusion and Future Work

This study presents a comprehensive framework for benchmarking the reasoning capabilities of large language models (LLMs) using Description Logic (DL), specifically focusing on the EL and ELH fragments. By translating logical axioms into natural language queries and comparing model predictions against formal entailment results, we revealed substantial differences in model behavior, logical generalization, and sensitivity to DL complexity.

Our findings indicate GPT-4o, Gemma 2, and Gemini 2.5 Pro achieved consistently high accuracy and reliability in DL-style tasks. In contrast, LLaMA 3 and DeepSeek Coder exhibit significant deficiencies and performance instability, particularly in ELH settings.

Future work includes:

- Implementing and evaluating more metamorphic reasoning types (e.g., paraphrasing, axiom removal)
- Expanding to additional DL fragments such as ALC or SHOIN
- Applying fine-tuning or prompt engineering to guide model reasoning more effectively
- Incorporating formal proof explanations alongside binary entailment responses

References

[1] Alibaba DAMO Academy. Qwen: Large language models by alibaba damo. https://github.com/QwenLM/Qwen, 2024. Accessed July 2025.

[2] DeepSeek AI. Deepseek coder: Code llm by deepseek ai. https://github.com/deepseek-ai/DeepSeek-Coder, 2024. Accessed July 2025.

[3] Meta AI. Llama 3: Open foundation models. https://ai.meta.com/llama/, 2024. Accessed July 2025.

[4] Mistral AI. Mistral ai model overview. https://mistral.ai/news/, 2024. Accessed July 2025.

[5] Franz Baader, Sebastian Brandt, and Carsten Lutz. Pushing the el envelope. In *Proceedings of the 19th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 364–369, 2005.

[6] Franz Baader, Diego Calvanese, Deborah L McGuinness, Daniele Nardi, and Peter F Patel-Schneider. *The Description Logic Handbook: Theory, Implementation, and Applications*. Cambridge University Press, 2007.

[7] Olivier Bodenreider. Biomedical ontologies in action: Role in knowledge management, data integration and decision support. *Yearbook of Medical Informatics*, 17:67–79, 2008. https://pubmed.ncbi.nlm.nih.gov/18660879/.

[8] Tsong Yueh Chen, Feng-Jian Kuo, Douglas Towey, and Tsun S. Tse. Metamorphic testing: A review of challenges and opportunities. *ACM Computing Surveys (CSUR)*, 51(1):1–27, 2018.

[9] Peter Clark, Oyvind Tafjord, and Kyle Richardson. Transformers as soft reasoners over language. *arXiv preprint arXiv:2002.05867*, 2020.

[10] Google DeepMind. Gemini 1.5 technical overview. https://deepmind.google/technologies/gemini/, 2024. Accessed July 2025.

[11] Google DeepMind. Gemma: Lightweight, open models by google. https://ai.google.dev/gemma, 2024. Accessed July 2025.

[12] Birte Glimm, Ian Horrocks, Boris Motik, and Rob Shearer. Hermit: An owl 2 reasoner. *Journal of Automated Reasoning*, 53(3):245–269, 2014.

[13] Simeng Han, Hailey Schoelkopf, Yilun Zhao, Zhenting Qi, Martin Riddell, Wenfei Zhou, James Coady, David Peng, Yujie Qiao, Luke Benson, et al. Folio: Natural language reasoning with first-order logic. *arXiv preprint arXiv:2209.00840*, 2022.

[14] Matthew Horridge. The pizza owl ontology, 2004. https://owl.cs.manchester.ac.uk/publications/ontologies/pizza/.

[15] Matthew Horridge and Sean Bechhofer. Owl api: A java api for owl ontologies, 2011. Available at http://owlcs.github.io/owlapi/.

[16] Shaoxiong Ji, Shirui Pan, Erik Cambria, Pasi Marttinen, and Philip S. Yu. A survey on knowledge graphs: Representation, acquisition and applications. *IEEE Transactions on Neural Networks and Learning Systems*, 33(2):494–514, 2022.

[17] Yevgeny Kazakov, Markus Krötzsch, and Filip Simančík. The incredible elk: From polynomial procedures to efficient reasoning with el ontologies. In *Journal of Automated Reasoning*, volume 53, pages 1–61, 2014.

[18] Boris Motik, Rob Shearer, and Ian Horrocks. The hermit owl reasoner. In *Proc. of the 5th International Workshop on OWL: Experiences and Directions (OWLED)*, 2007.

[19] OpenAI. Gpt-4o technical report. `https://openai.com/index/gpt-4o/`, 2024. Accessed July 2025.

[20] Teeradaj Racharak, Chaiyong Ragkhitwetsagul, Chommakorn Sontesadisai, and Thanwadee Sunetnanta. Test it before you trust it: Applying software testing for trustworthy in-context learning. 2025.

[21] Stanford Center for Biomedical Informatics Research. The pizza owl ontology (protege). `https://protege.stanford.edu/ontologies/pizza/pizza.owl`, 2025. Accessed July 2025.

[22] Meng Sun, Yuxuan Zhao, Xin Liu, Nan Meng, Ming Yang, Xiaoting Lu, Tian Chen, Tian Zhang, Xin Li, Yuening Hu, Chao Li, and Xiaofei Wang. Metamorphic testing of large language models. In *Proceedings of the 46th International Conference on Software Engineering (ICSE)*, pages 2270–2282, 2024.

[23] Alon Talmor, Oyvind Tafjord, Peter Clark, Yoav Goldberg, and Jonathan Berant. Leap-of-thought: Teaching pre-trained models to systematically reason over implicit knowledge. *arXiv preprint arXiv:2006.06609*, 2020.

[24] Dmitry Tsarkov and Ian Horrocks. Fact++ description logic reasoner: System description. In *International Joint Conference on Automated Reasoning*, pages 292–297. Springer, 2006.