Logical Inference in Wumpus World

Taron Moe-Stull
taronmoe@icloud.com

October 2025

Abstract

This report describes a First-Order Logic (FOL) reasoning system designed to determine the safety of a query cell in Wumpus World using unification and resolution. The system ingests cave text files containing an agent path with percepts (breeze and stench) and translates those observations into logical clauses in Conjunctive Normal Form (CNF). A knowledge base is incrementally expanded as inference proceeds, and the query cell is evaluated using proof by contradiction.

The design keeps domain-specific rules separate from the core inference engine so the same pipeline can be adapted to other rule sets. To evaluate behavior, the system was tested on twelve caves spanning easy, medium, and hard difficulty. Across these tests, accuracy was 83% (10/12). Computation cost increased sharply with grid size and percept ambiguity, with operation counts trending close to exponential growth as environments became larger and less informative.
Keywords: First-Order Logic, Unification, Resolution, Knowledge Base, Wumpus World

1. Problem Statement

The goal is to build a First-Order Logic (FOL) reasoning system centered on unification and resolution. Given a set of rules and observed facts, the system should infer new facts via logical inference. The core inference processes are designed to be independent of any single problem formulation so the system can be extended as new domain rules are introduced.

For each Wumpus World input cave, the system must determine whether the provided query cell is **SAFE**, **UNSAFE**, or **RISKY**. Each puzzle provides:

- a grid size (cave dimensions),
- a hazard constraint (number of Wumpus/arrows),
- an agent path containing percepts at visited cells (Breeze/Stench),
- and a query cell representing the next candidate move.

The system produces a text output artifact that includes the logical clauses used during inference, and ends with a terminal classification line:

```
QUERY: SAFE/UNSAFE/RISKY
```

In addition, the system records metrics (variables unified, clauses resolved, and classification accuracy against the reference resolution supplied with each puzzle).

Hypothesis

As cave difficulty increases (larger grids, fewer visited cells, higher ambiguity in percepts, and greater hazard density), the number of unifications and resolutions required is expected to increase, the frequency of **RISKY** outcomes is expected to rise, and accuracy should remain high on instances where **SAFE** or **UNSAFE** can be proven with sufficient evidence.

Conversely, when the input provides strong negative evidence (e.g., multiple adjacent "no stench" or "no breeze" constraints) and when hazard count constraints reduce ambiguity (e.g., no remaining Wumpus placements are possible), the system should reach **SAFE/UNSAFE** conclusions with fewer inference steps.

2. Problem Description

Each input file describes a Wumpus World cave in terms of a partially observed environment. Visited path cells include percepts: **stench** indicates an adjacent Wumpus and **breeze** indicates an adjacent pit. These percepts form the evidence used to infer the safety of the query cell.

The system converts percept observations into FOL clauses and inserts them into a knowledge base (KB) along with domain rules (e.g., "If a cell has stench, at least one adjacent cell has a Wumpus"; "If a cell has no breeze, no adjacent cell has a pit"). Once the KB is initialized, the system appends a negated query statement (e.g., $\neg SAFE(q)$ or $\neg UNSAFE(q)$) and uses unification and resolution to attempt a proof by contradiction.

To determine safety:

- Append $\neg UNSAFE(q)$ and attempt to derive a contradiction to prove $SAFE(q)$.
- Append $\neg SAFE(q)$ and attempt to derive a contradiction to prove $UNSAFE(q)$.
- If neither can be proven, classify the query cell as **RISKY**.

3. Experimental Approach

To evaluate correctness and performance, the system was run against twelve input cave files across easy, medium, and hard difficulty. For each cave, inference was executed to classify the query cell and to record computational cost as the total number of inference operations.

System Pipeline

1. **File and Data Import.** The input file is parsed to extract cave size, percept data, the visited path, and query coordinates. Percepts are encoded into boolean structures indicating whether each cell has a breeze, a stench, and whether the cell has been visited.
2. **Knowledge Base Initialization.** An internal cave representation is created and initialized with base facts, visited pathing, and domain rules. The system tracks possible and confirmed pits/Wumpus placements using structured arrays. Initial rules include negative evidence constraints such as:

- If a cell has **no stench**, adjacent cells do **not** contain a Wumpus.
- If a cell has **no breeze**, adjacent cells do **not** contain a pit.

3. **Inference via Unification and Resolution.** The system appends the negated query (e.g., $\neg SAFE$ or $\neg UNSAFE$) and performs recursive unification and resolution:
   - `unifyFunction()` matches variables to constants across clauses, producing substitutions and simplifying expressions.
   - `resolveStatements()` combines clauses, applies substitutions, and searches for contradictions.

   Inference terminates when a contradiction is found (proving the target statement) or when no new resolvents can be generated (leading to a **RISKY** classification).
4. **Metrics Collection.** The system tracks unification and resolution activity using an internal counter (e.g., `runCount`) and records whether the final classification matches the reference resolution.
5. **Output Generation.** After inference completes, the system produces a text output that logs the clauses involved in the final determination and ends with the required `QUERY: ...` line. This output serves as both an explanation trace and a debugging artifact.

4. Results

Twelve Wumpus World caves were evaluated. For each input file, the system classified the query cell as SAFE/UNSAFE/RISKY and recorded the total number of inference operations required.

Table 1: Prediction vs. ground truth and operation counts per path (accuracy: 10/12 = 83%).

Path	Ground Truth	Prediction	Operations
path_e1	UNSAFE	UNSAFE	4,151
path_e2	RISKY	RISKY	429,448
path_e3	SAFE	RISKY	269,554
path_e4	RISKY	RISKY	105,272
path_m1	RISKY	RISKY	427,362
path_m2	UNSAFE	RISKY	426,267
path_m3	RISKY	RISKY	144,442
path_m4	RISKY	RISKY	608,291
path_h1	RISKY	RISKY	307,564
path_h2	SAFE	SAFE	118,808
path_h3	RISKY	RISKY	544,352
path_h4	RISKY	RISKY	952,686

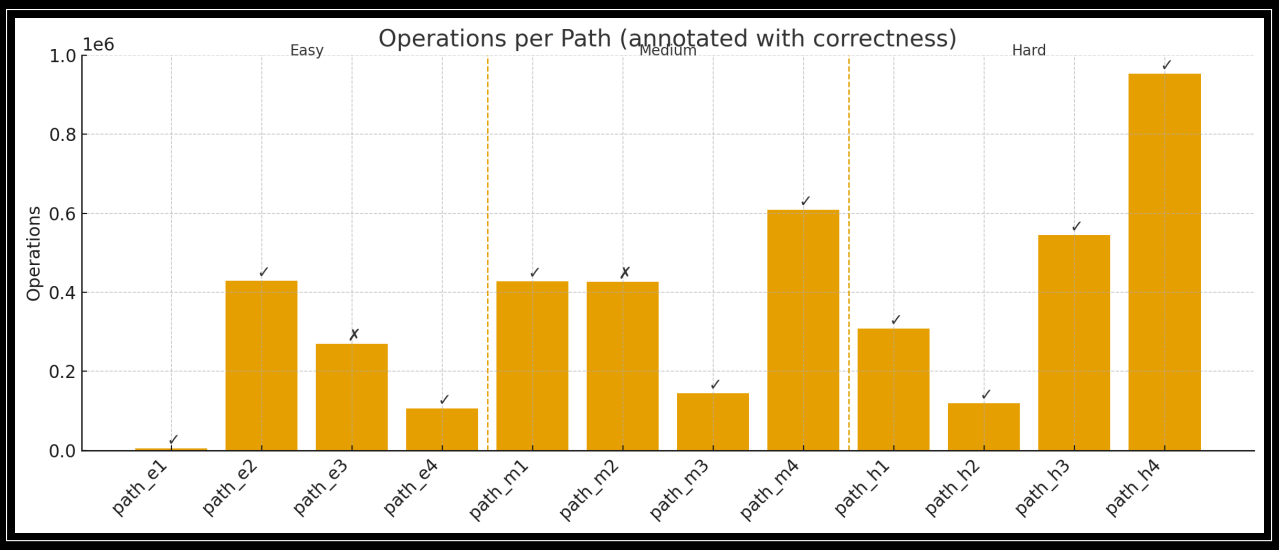

Figure 1: Operations per path (annotated with correctness). Check marks indicate correct predictions; X marks indicate incorrect predictions.

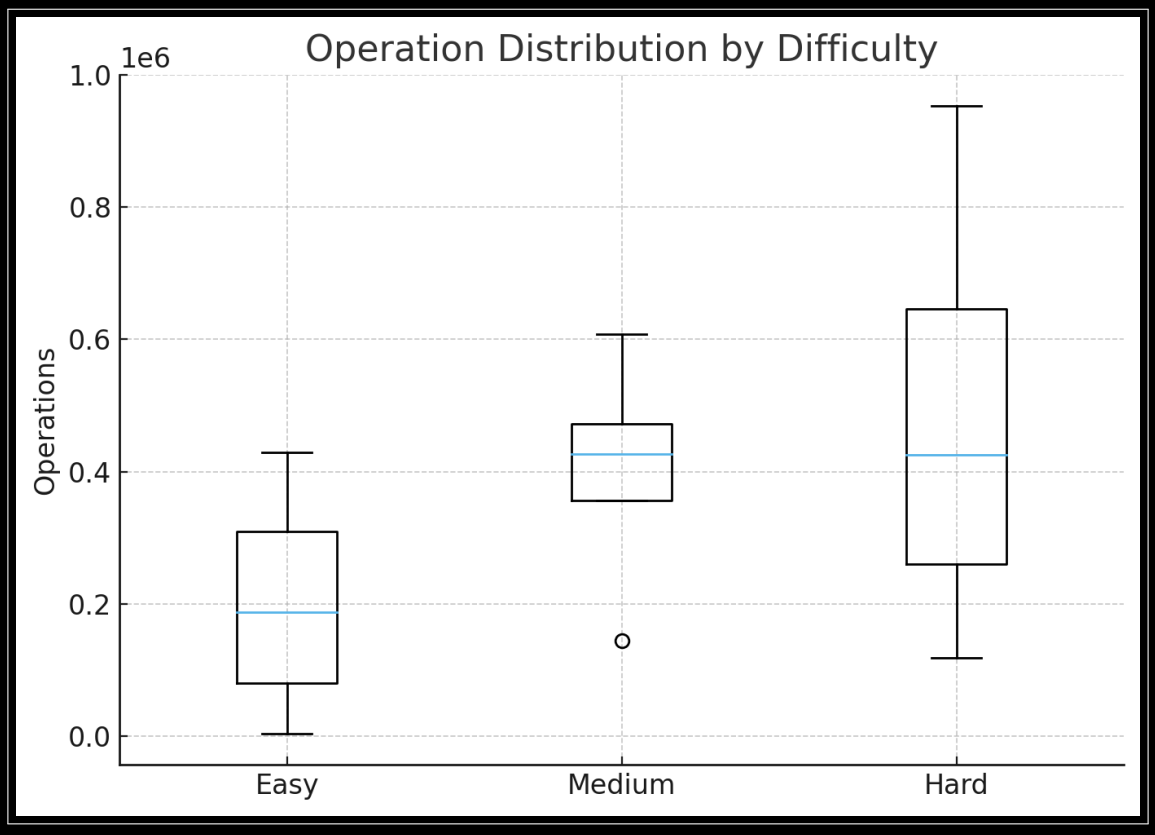

Figure 2: Operation distribution by difficulty (Easy, Medium, Hard).

5. Discussion

Table 1 and Figures 1 and 3 show a clear trend in how the reasoning system behaves as cave difficulty increases. Overall accuracy reached 83%, with 10 of 12 query cells correctly classified. The two

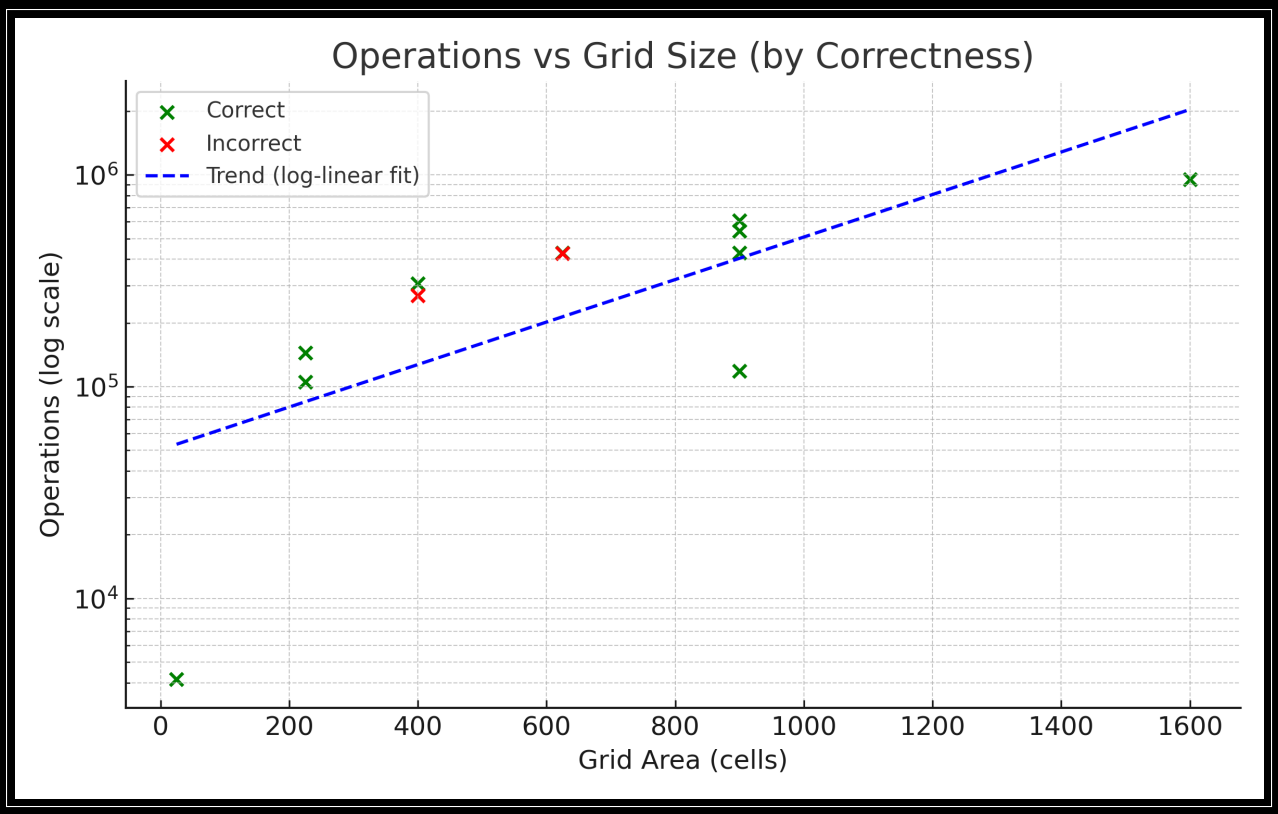

Figure 3: Operations versus grid area (cells), annotated by correctness, with a log-linear trend line.

incorrect predictions occurred in scenarios where the ground truth was SAFE or UNSAFE but percept evidence was highly ambiguous; in those cases, the visited path did not provide sufficient information to support a definitive proof by contradiction.

Across difficulty levels, computation cost correlates strongly with environment size. As shown in Figure 3, operation counts rise sharply as grid area increases. This supports the hypothesis that larger and more ambiguous environments require substantially more unifications and resolutions. As the knowledge base grows, clause interactions increase and the combinatorial space of plausible hazard configurations expands.

Easy caves generally resolved with fewer operations and more frequently produced definitive SAFE/UNSAFE classifications when strong negative evidence was present (e.g., multiple adjacent cells with no breeze or no stench). Medium and hard caves commonly required hundreds of thousands of operations, reflecting both larger grid sizes and reduced percept coverage. Harder caves also more frequently resulted in RISKY classifications, not necessarily due to system error, but because available percept data was insufficient to prove safety or unsafety conclusively.

Figure 2 further confirms that higher difficulty corresponds to higher computational cost and greater variance in operation counts. This variance suggests performance may depend not only on grid size, but also on the ordering and distribution of percept evidence along the visited path.

6. Summary and Next Steps

This project implements a First-Order Logic reasoning system that applies unification, resolution, and proof by contradiction to classify the safety of a query cell in Wumpus World caves. Across twelve input caves of increasing difficulty, the system achieved 83% accuracy.

The results show that computation cost increases significantly with cave size and percept ambiguity. Additionally, the system frequently outputs RISKY when available evidence cannot support a proof of SAFE or UNSAFE, which is expected behavior under partial observability.

The main limitation observed is scalability: as the environment grows and uncertainty increases, knowledge-based inference can become computationally expensive even when the logic is correct. If revisiting this system, the most impactful improvements would likely come from optimizing clause resolution and introducing heuristics to reduce the search space of resolvents.

References

1. Stuart J. Russell and Peter Norvig. *Artificial Intelligence: A Modern Approach.* Pearson, 4th edition, 2020.
2. Lakshmi Prabha. *The Wumpus World Description.* GeeksforGeeks, 2025. Accessed October 2025.
3. Overleaf. *Overleaf, online LaTeX editor.* Accessed October 2025.