Task 7: Logical Inference

LLM+Logic Project

Overview

This document describes the enhanced backward chaining system implemented for Task 7 of the LLM+Logic project. The objective was to build a robust logical inference engine for First Order Logic (FOL) using the backward chaining approach, with a focus on correct variable handling, support for complex rules, and comprehensive testing. The implementation is written from scratch in Python.

System Design

The system represents the knowledge base using facts and rules, both encoded as strings. Each rule consists of a body (premises) and a head (conclusion). The engine includes modules for parsing predicates, performing unification, applying substitutions, and recursively attempting to prove a query based on known facts and applicable rules.

Example Facts and Rules

```
prolog
Facts:
parent(john, mary)
parent(mary, alice)
parent(alice, bob)
parent(bob, charlie)
parent(susan, tom)
parent(tom, jerry)
male(john)
female(mary)
...

Rules:
parent(X, Y) => ancestor(X, Y)
ancestor(X, Y) and parent(Y, Z) => ancestor(X, Z)
parent(X, Y) and male(X) => father(X, Y)
parent(X, Y) and female(X) => mother(X, Y)
ancestor(X, Y) and ancestor(X, Z) and Y != Z => related(Y, Z)
```

Inference Engine

The main function, bc_ask, is goal-driven:

- It first checks whether a query matches any known fact.
- If not, it searches for rules whose head matches the query, then recursively attempts to prove each condition in the rule's body.
- Unification is handled for variable matching, with substitutions applied consistently and cycle prevention for recursion.

Testing and Results

Testing was performed using a comprehensive suite of manually defined test cases, including both direct and chained inferences, as well as negative cases where a query cannot be proven. The system returns accurate results for all tested cases.

Sample Test Results

Query	Expected	Result
ancestor(john, alice)	True	True
ancestor(john, bob)	True	True
ancestor(john, charlie)	True	True
ancestor(alice, john)	False	False
father(john, mary)	True	True
mother(mary, alice)	True	True
father(mary, alice)	False	False
related(alice, bob)	True	True
ancestor(bob, john)	False	False
related(bob, tom)	False	False

The system now correctly returns False for queries such as ancestor(alice, john), reflecting the directionality of ancestry and the improved unification logic.

Conclusion

This task reinforced core concepts of symbolic reasoning, recursion, and rule-based inference. The revised implementation addresses previous issues with variable handling and directionality, and now provides a solid foundation for more advanced logic systems or integration with language models in future work.