Datalog: Logic Reasoning for LLMs

Arsh Advani
Abhishek Chavan
Suryakiran Valavala

What is Datalog?

Datalog is a declarative logic programming language used for querying structured data.

It is a syntactically restricted version of Prolog that ensures termination.

Commonly used in database systems, program analysis, and knowledge representation.

Programs are composed of facts, rules, and queries.

Datalog Syntax

```
% Facts
parent(john, mary).
parent(mary, alice).

% Rule
grandparent(X, Y) :- parent(X, Z), parent(Z, Y).

% Query
?- grandparent(john, Y).
```

Datalog Semantics

Uses **bottom-up evaluation** (also called fixpoint semantics).

Applies rules repeatedly to known facts until no new facts can be derived.

Guarantees termination due to no function symbols or complex terms.

Result is the minimal model that satisfies all facts and rules.

How It Runs

Tool Options:

- **Soufflé** – Command-line high-performance engine for `.dl` files.
- **PyDatalog** – Integrates Datalog syntax into Python programs.

```
from pyDatalog import pyDatalog

pyDatalog.create_terms('parent, grandparent, X, Y, Z')

+parent('john', 'mary')

+parent('mary', 'alice')

grandparent(X,Y) <= parent(X,Z) & parent(Z,Y)

print(grandparent(X, 'alice'))
```

Why It’s Useful for LLMs

Datalog provides symbolic structure that LLMs can be guided to generate.

LLMs can translate natural language into Datalog rules and queries.

Symbolic logic engines handle inference reliably and explainably.

Combines strengths: LLMs for language understanding, Datalog for precise reasoning.