Task 9 Write-up: LangGraph Integration with Symbolic Reasoning

For Task 9, I extended the symbolic reasoning system from Task 8 by migrating it to LangGraph. The goal was to structure logical inference using LangGraph's stateful graph execution, building on a Prolog-style knowledge base and Retrieval-Augmented Generation (RAG).

Overview

I reused the same knowledge base from Task 8, consisting of:

- **Facts**: `animal(sparrow)`, `flies(eagle)`, etc.
- **Rules**: `bird(X) :- animal(X), flies(X)`

The documents were chunked and embedded using HuggingFace's `sentence-transformers`, and indexed via FAISS. I created a LangGraph `StateGraph` with two nodes:

- **Retriever Node**: Retrieves relevant KB facts/rules.
- **Judgment Node**: Evaluates relevance and generates a simple explanation.

To satisfy the Chain of Thought (CoT) requirement, I added a reasoning step that explicitly prints a rationale from the retrieved context.

Testing

I tested the query:

"Can a sparrow fly?"

LangGraph retrieved:

```
Unset
animal(sparrow). flies(sparrow).
```

And returned:

"→ Reasoning: The fact 'flies(sparrow)' exists, so yes, it can fly."

This demonstrates a basic CoT layer applied after symbolic context retrieval.

Technologies Used

- LangGraph + LangChain
- HuggingFace sentence-transformers
- FAISS
- Google Colab

Repository

All code, notebook, and outputs are available on: GitHub Repository