Graphs, Nodes, and Edges: Essay Evaluation System

This document presents a modern example of how **LangGraph** can model an intelligent, stateful, and conditional workflow for an **AI Essay Evaluation System**. The process dynamically generates essay topics, evaluates multiple criteria in parallel, and routes based on performance, demonstrating *nodes, edges, parallelism, and looping* in practice.

Conceptual Flow

1. **Generate Topic** The system creates a relevant UPSC-style essay topic for the student.
2. **Collect Essay** The student writes and submits the essay.
3. **Evaluate Essay (Parallel Evaluation Block)** The system evaluates the essay **in parallel** across three dimensions:
   - **EvaluateDepth** – Analyzes depth, critical thinking, and argument strength.
   - **EvaluateLanguage** – Checks grammar, vocabulary, and tone.
   - **EvaluateClarity** – Assesses coherence and logical flow.
4. **Aggregate Results** The three scores are combined into a total score (e.g., out of 15).
5. **Conditional Routing**
   - If score $\geq$ threshold $\Rightarrow$ **ShowSuccess**.
   - If score $<$ threshold $\Rightarrow$ **GiveFeedback**.
6. **Give Feedback** Provides detailed improvement suggestions.
7. **Collect Revision (Optional Loop)** Allows resubmission and loops back to evaluation.
8. **Show Success** Congratulates the student and ends the process.

Key LangGraph Concepts

- **Nodes:** Represent individual tasks or LLM calls.
- **Edges:** Define transitions or logical routing between nodes.
- **Parallel Blocks:** Enable concurrent task execution.
- **Conditional Branching:** Directs workflow paths based on computed values.
- **Loops:** Allow iterative refinement and re-evaluation.

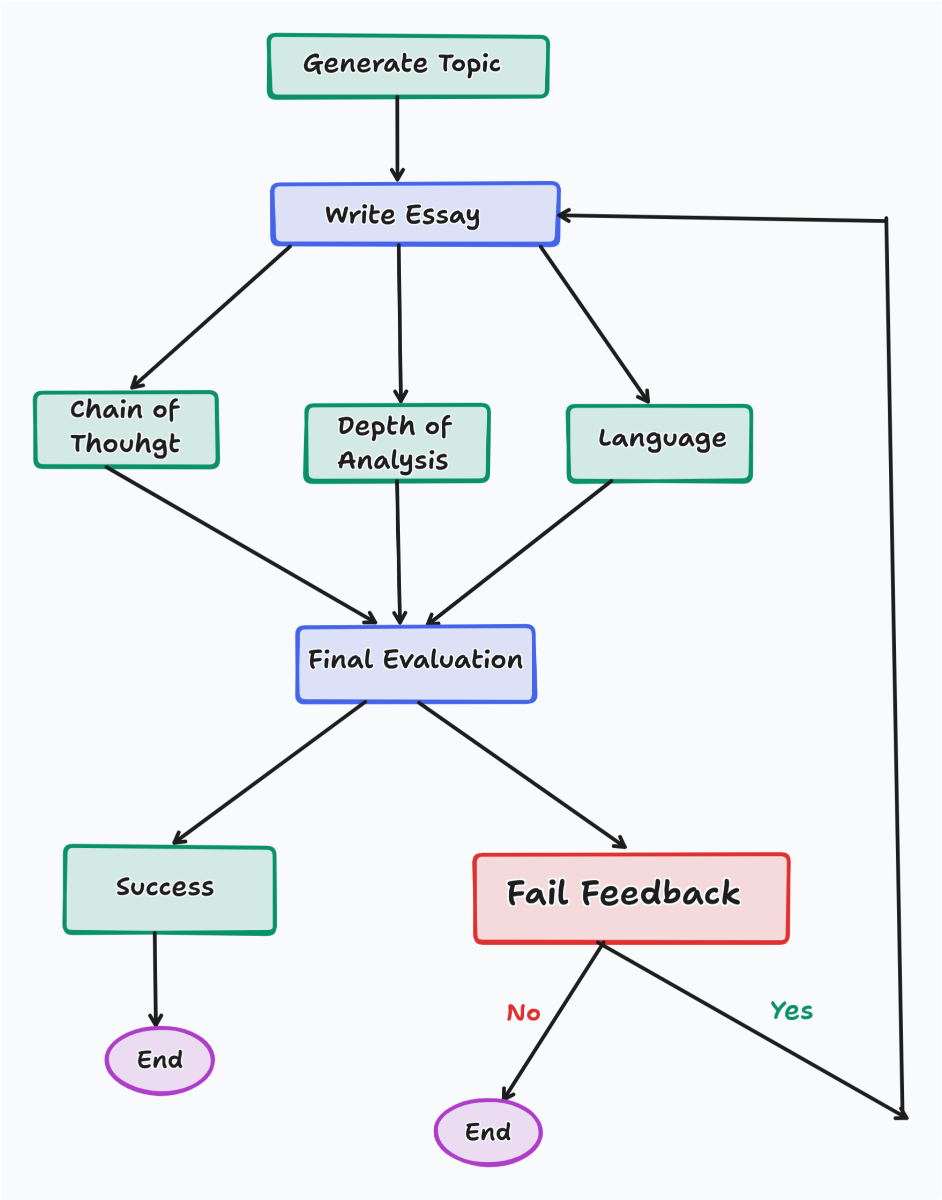

Figure 1: **LangGraph Architecture for Essay Evaluation Workflow**