AI ASSIGNMENT-3

Name: Rishav Aich
Roll No.: B21AI029
Experiment code: GitHub Repo

Experimental Details

Purpose

The purpose of this experiment is to evaluate the performance of the resolution refutation system implemented in the given C++ code. Specifically, we aim to determine the satisfiability of a knowledge base through both uninformed search and informed (greedy) search.

Resolution Rule

The core of the resolution refutation approach is the resolution rule, which states that if there are two clauses that contain complementary literals, then a new clause can be formed by resolving these clauses. This involves removing the complementary literals and merging the remaining literals.

Implementations	Approach
Conversion of all clause in the KB to CNF	• **CNF Conversion:** Convert the logical expression into Conjunctive Normal Form (CNF), a conjunction of disjunctions. • **Implication Handling:** Modify bidirectional implications and convert implications to simplify the CNF structure. • **Negation Simplification:** Apply De Morgan's laws and push negations inside parentheses to streamline the expression. • **Conjunction-Distribution:** Address cases where conjunctions distribute over disjunctions, further simplifying the CNF. • **Resolution Refutation:** Utilize the CNF expression as a knowledge base for resolution refutation to determine logical satisfiability.

Resolution Refutation Procedure	● Initialize the Knowledge Base (KB) with the CNF form of the logical expression. ● Repeat the following steps until a contradiction is found or no further resolutions can be made: ○ For each pair of clauses in the KB, attempt to resolve them using the resolution rule. ○ If a new resolvent is obtained, add it to the KB. ○ If an empty clause is obtained, indicating a contradiction, terminate and conclude that the knowledge base is unsatisfiable.
Searches	● **Uninformed Search:** ○ Use uninformed search to explore resolution paths systematically by BFS. ○ Evaluate pairs of clauses for resolution, adding new resolvents to the KB. ○ Terminate the process if a contradiction is found. ● **Informed Search:** ○ Applied informed greedy search strategies with a heuristic function. ○ **Heuristic:** Heuristic prioritize resolving clauses that result in the smallest resolvents. This is based on the assumption that smaller resolvents are more likely to lead to a contradiction (an empty clause). ○ Prioritize clauses that lead to smaller resolvents, assuming they are more likely to contradict.

Observation

- Successful implementation of the resolution rule, identifying contradictions based on complementary literals during resolution.
- Efficient conversion of logical expressions to CNF with proper handling of bidirectional implications, negations, and conjunction-distribution cases.
- Systematic resolution refutation procedure using CNF as the Knowledge Base (KB), with effective termination upon contradiction detection.
- Both uninformed and informed (greedy) search strategies demonstrated efficacy, with informed search showing potential for optimization by prioritizing smaller resolvents.

- The output of the code shows that uninformed search and informed search have almost the same number of steps.

Result

- The resolution refutation system successfully identified contradictions based on complementary literals during the resolution process.
- The conversion of logical expressions to Conjunctive Normal Form (CNF) demonstrated efficiency, addressing bidirectional implications, negations, and conjunction-distribution cases effectively.
- The systematic resolution refutation procedure, utilizing CNF as the Knowledge Base (KB), proved effective in detecting contradictions, with timely termination upon contradiction detection.
- Both uninformed and informed (greedy) search strategies showed efficacy in exploring resolution paths, with the informed search demonstrating potential for optimization by prioritizing smaller resolvents.