Project 2

Team Members: Yuxin Liu 31425544, Sining Qu 31428972

Introduction

In this project we are implementing 2 methods - model checking and propositional resolution to solve 4 propositional logic problems. We started with converting clauses into cnf and then apply the two methods respectively.

CNF

We use knowledge base to store the clauses given in the problems after converting then into cnf. To convert clauses into cnf, we wrote the sentences in the problems into a concise form and transformed them obeying the laws given in lectures. First, we eliminated biconditionals, changed them into a conjunction of two implications. Second, we eliminated implications, for example P ⇒ Q was transformed to ¬P∨Q which is represented by (not P) or Q. Then, we moved negation inwards and distributed disjunctions over conjunctions. The clauses in cnf forms are then put into the knowledge base.

Model checking

We define the function tTEntails(kb, alpha) to implement the algorithm of truth table, the inputs are KB and alpha(the query which is a sentence in pl) and includes symbols which is a list of pl symbols in KB and alpha. It returns if KnowledgeBase entailments alpha(the result) True or False using truth table. Firstly, the clauses are converted to CNF with the function disCombine and propSymbols. The function tTCheckAll helps to implement the truth table, make the model likes a dictionary such as {'P': True, "Q": False}, and in this function, we first judge if it is empty then determine if KB and alpha are proposition logically true if not do the function stated next, finally return the conversion completed truth table. Function plTrue relates to the models, it returns True if the clause is true in the model, and False if it is false.Return None if the model does not specify all symbols. Function modelExtend returns the new model with p when value v is added. In each problem, we could easily see the KB transfer to the truth table.

Propositional Resolution

We use the clauses we have plus the negation of the result to try to prove if the result is True or False. We first take all the clauses in the knowledge plus the negation of result, and broke them up in pairs by combination. For each pair, we resolve them to obtain all possible clauses from them and check to see if we can eliminate any items in them, for example, if P and ¬P exist in a pair respectively, we can remove P from the clause and add what was remaining into a list for new information. After resolving each pair, we check if the list is a subset of our existing clauses and if it is then the result we are checking is False. If it is not a subset, then we append the list to our existing clauses to form a new list containing all the information we have and go through the process again. If empty lists are generated from resolving the pairs, it means that the result is True. There is one thing to note that because of the iterations we go through, especially when the result is false, it becomes really slow to solve problems with a large set of knowledge like problem 4. So it may be better to use model checking/truth tables for such problems.

Problem and Performance

The answer for the question 1:

Truth Table: Q is True.

Resolution: Q is True

So we can say that $\{P, P \Rightarrow Q\} \models Q$.

The answer for the question 2:

Truth Table: ['not', 'P12'] is True.

Resolution: ['not', 'P12'] is True.

So we can say that P12 is False, meaning that there is no pit at location [1,2].

The answer for the question 3:

Truth Table: mythical is False; magical is True; horned is True.

Resolution: mythical is False; magical is True; horned is True.

So we cannot prove that unicorn is mythical, but we can prove that it is magical and it is horned.

The answer for the question 4a:

Truth Table: A is True; B is False; C is False; D is False; E is False;F is False;G is True;H is True; X is True; Y is False; Z is False.

Resolution: A is True; G is True; H is True; X is True.

So A,G,H are knights telling the truth and the philospher should choose door X.

Because of the large size of the knowledgebase, propositional resolution for the results with False took way too long to run as it kept generating new lists after resolving pairs. So we only checked the True results and they are consistent with the results from model checking. It has the same issue in 4b so we did the same.

The answer for the question 4b:

Truth Table: A is True; B is False; C is False; D is False; E is False;F is False;G is True;H is True; X is True; Y is False; Z is False.

Resolution: A is True; G is True; H is True; X is True.

It is the same as part a, meaning that he had heard enough information to make a decision.

Member contributions

Yuxin Liu worked on model checking.
Sining Qu worked on cnf and propositional resolution