Andrew Nyaisonga 03/06/2018

Automated Reasoning

1. Program

This program uses a lot of design choices by Professor Ferguson. The blueprint of his code is all over the program. It has 4 main packages, ‘cnf’,’Computation’,’core’ and ‘examples’.

- **Core:** deals with the representation of propositional logic and it has
  - KB(Knowledge Base) which holdes premises of the specified word.
  - Connective: This are connectors to the sentences AND, OR and urinary like Negation.
  - Symbols: propositions that are true or false
  - Sentence: Make of Symbols
  - Compound Sentences: Make of small sentences with connectives-Urinary for negations and Binary for other connectives

  Model: Assign the truth value of the sentences
- **cnf:** This has methods like
  - CNF Converter which converts any sentence to conjunctive normal form. This is extra important on using resolution
  - Closure: Used in Resolution to iterate through literals when resolving two clauses
  - Literals: Negation of the symbol or the symbol itself
- **Computation**: This is the most important package of the whole project. It contains:
  - Model Checking: Which is the method that uses truth table to enumerate all possibilities
  - Resolution: That has the method that uses Resolution Theorem as in AIMA Figure 7.12
- example: This contains all the implemented examples: It has Wumpus World, Modus Ponens and Horn Clauses

2. Basic Model Checking

For this part I implemented a very straightforward Model checking algorithm from AIMA figure 7.10. I implemented FIRST/REST with arraylist which just remove the first element (FIRST) and the remaining is REST. This algorithm is sound as explain on the bok(Sound as if a follows b then b can be derived from a). The algorithm is also complete as it will terminate for any model. It is really slow however but that is just as good as we can get with Model checking. It 's O(2^n) in time and O(n) for space.

Here is a snippet of the book description:

function TT-ENTAILS?(KB, α) **returns** true or false
inputs: KB , the knowledge base, a sentence in propositional logic

α, the query, a sentence in propositional logic symbols ← a list of the proposition symbols in KB and α
return TT-CHECK-ALL(KB,α,symbols,{})

function TT-CHECK-ALL(KB,α,symbols,model) **returns** true or false **if** EMPTY?(symbols) **then**
if PL-TRUE?(KB,model) **then return** PL-TRUE?(α,model)
else return true // *when KB is false, always return true* **else do**

P ← FIRST(symbols)
rest ← REST(symbols)
return
(TT-CHECK-ALL(KB,α,rest,model ∪ {P = true})**and**TT-CHECK-ALL(KB,α,rest,model ∪ {P = false }))

3. Advanced Propositional Inference

Used resolution as described in AIMA Figure 7.12.

As the Ferguson implementation of CNF converter for proposition with ~a and KB it; remove implications and biconditionals and move negation inwards, finally it distribute OR over AND. The center part of resolution algorithm is the resolver which check for

pairs of literals that have same symbol but different polarity, remove them those from the pair of clauses and add it to the new clauses that will be used for another iteration.

This algorithm is sound and complete. Sound as it can be shown with truth table. Complete as stated in the theorem on the book. There is also a cool proof for this that I found online.

function PL-RESOLUTION(KB, α) **returns** true or false
inputs: KB , the knowledge base, a sentence in propositional logic

α, the query, a sentence in propositional logic

clauses ← the set of clauses in the CNF representation of KB ∧ ¬α new ← { }
loop do

for each pair of clauses Ci, Cj **in** clauses **do**
resolvents ← PL-RESOLVE(Ci, Cj)
if resolvents contains the empty clause **then return** true new ← new ∪ resolvents

if new ⊆ clauses **then return** false clauses ← clauses ∪ new

4. Worlds implemented

There are three world implemented in the program. There is

Wumpus World that has this fact(KB)

¬P1,1
B1,1 ⇐⇒ (P1,2 ∨ P2,1)
B2,1 ⇐⇒ (P1,1 ∨ P2,2 ∨ P3,1)
¬B1,1
B2,1

We are trying to show find whether P1,2 (that is, whether there is a pit at location [1,2]) is true or not.

Modus Ponens

That claims that {P, P ⇒ Q} |= Q and we are trying to prove this with both Resolution and Model checking.

Horn Clauses (Russell & Norvig)

Using the KB:If the unicorn is mythical, then it is immortal, but if it is not mythical, then it is a mortal mammal. If the unicorn is either immortal or a mammal, then it is horned. The unicorn is magical if it is horned.
We are tying to answer:

(a) Can we prove that the unicorn is mythical?
(b) Can we prove that the unicorn is magical?
(c) Can we prove that the unicorn is horned?

All implementation and output are on their corresponding Files

5. Running

You can run the program via command line (terminal)

-cd to the src directory of the
Andrew_Nyaisonga_Folder
-Compile using normal java compilation Example:
javac WumpusWorldKB.java
-Then run the program with:
java WumpusWorldKB

6. Finally

Doing this project has allowed me to implement some of the important algorithms in reasoning on AI. It also helped me appreciate the difficulties of answering this kind of questions. I implemented Liar Truth tellers but failed to include with this version of project. Looking to add it in the near future