Research report

ASSIGNMENT 2

CHRISHAN CHRISTESIOUS ALOYSIOUS

101268207

VANESSA YING XHIEN WONG

101951824

COS30019\_A02\_T037

Table of Content

Contents

[Table of Content 1](#_Toc10201035)

[Implementing a Logical Expression for General Knowledge Base 2](#_Toc10201036)

[Resolution Algorithm 4](#_Toc10201037)

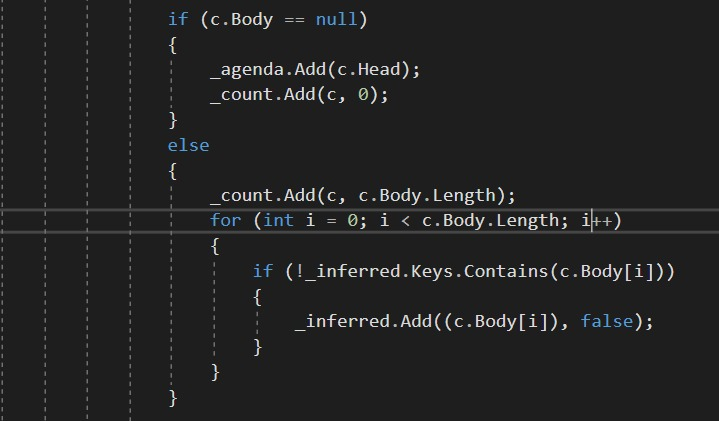
[References 6](#_Toc10201038)

Implementing a Logical Expression for General Knowledge Base

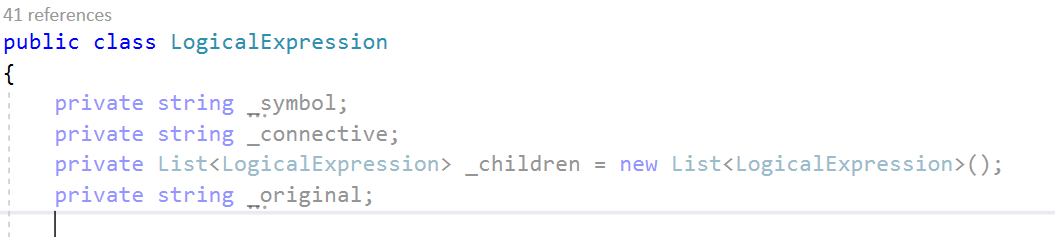
The main challenge of implementing a general knowledge base compared to a horn knowledge base is the correct selection of data structure for a sentence in proposition logic. In horn clauses, all sentences can be viewed and categorized as three types depending on the number of bodies they have.

|  |  |  |  |
| --- | --- | --- | --- |
| Type | Sentence | Body | Head |
| 1 | A | Null | A |
| 2 | A => B | A | B |
| 3 | A&B&C => D | A, B, C | D |

This categorization was used throughout the InferenceEngine program coding to identify the sentences.



However, this approach is not the best strategy when it comes to dealing with a general knowledge base. We follow the concept of trees which is recursive data structure.



Here symbols are the leafs and connective is a tree node and the sentences it connects are the children of the node. For example, the statement B12 <=> (P11 OR P22 OR P13) becomes:

A picture containing object

Description automatically generated

The symbol and connective member variables in the content of the tree node. In the case, when the sentence is a symbol, then

\_symbol = symbol;

\_connective = NULL;

\_children = NULL;

If the sentence is not a simple symbol,

\_symbol = NULL;

\_connective = the top-level connective;

\_children = the sentences that the connectives connect;

It is clearly shown that symbol and sentence cannot both exists in a node.

Resolution Algorithm

Resolution is using proof by Contradiction. Proof by Contradiction means, to show that KB entails query, we need to show that KB entails not-query is unsatisfiable. To perform resolution algorithm, the first step is to convert the clauses into Conjunctive Normal Form (CNF) with the negated conclusion. The CNF is a collection of disjunctions inside of clauses and all the clauses are conjunct together. By eliminating the implication connective to negation and disjunction connectives, we can easily make CNF.

For example, a=>b in CNF is ~(a||b).

Also, using De-Morgan’s Rule can be used to eliminate conjunction to disjunction and negation.

For example, ~ (a & b) = (~a || ~b).

An example of converting clauses into CNF:

|  |  |  |
| --- | --- | --- |
| A < = > (B || C) | Biconditional Elimination | (A => (B||C))  &  ((B||C)=>A) |
| (A => (B||C))  &  ((B||C)=>A) | Implication Elimination | (~A||B||C)  &  (~(B||C)||A) |
| (~A||B||C)  &  (~(B||C)||A) | De Morgan | (~A||B||C)  &  ((~B & ~C)||A) |
| (~A||B||C)  &  ((~B & ~C)||A) | Distribution | (~A||B||C)  &  (~B||A) &  (~C||A) |

After the clauses have been converted, the equation consists of complimentary pairs. Complimentary literal means in 2 different clauses, they contain the same symbol, but the values are opposite.

For example, R1: (A||B), (~ A|| C), the complimentary literals in the both clauses are (a, ~a). after resolving the pairs, the clause is (B || C).

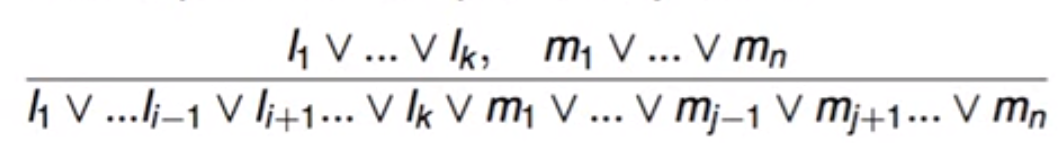


Figure 1 Resolution pairs.

The next step would be resolving the complimentary literals.

|  |  |  |
| --- | --- | --- |
| (~A||B||C)  &  (~B||A)  &  (~C||A) | Resolve | (B||C ||~B) &  (~C||A) |
| (B||C ||~B) &  (~C||A) | Resolve | (C) &  (~C||A) |

If after resolving all the complimentary literals results in an empty clause, then we can conclude that KB entails query.

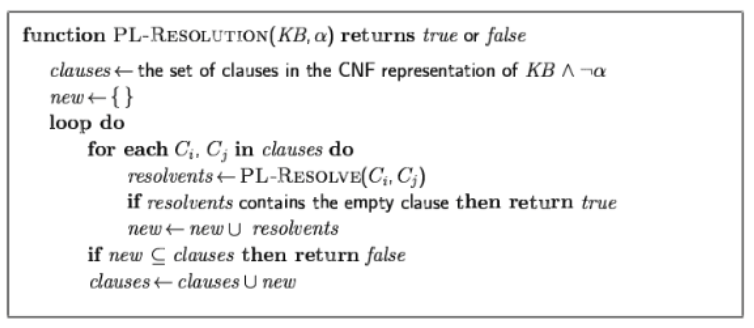


Figure 2 Pseudocode for Resolution Algorithm.

References

Vassilis Athitsos, TT-ENTAILS: Inference by Enumeration in Propositional Logic

* This pdf gives us a in depth information to structure the truth table algorithm for generic knowledge base.

Introduction to AI Lecture 3 and 4 slides

Ankit Shah*,* 2007. *RESOLUTION.*

* Detailes explanation and example on resolution algorithm

Francisco Iacobelli. <https://www.youtube.com/user/fiacobelli/videos>

* A series of AI algorithms explanation videos which helps me to understand resolution.