

**COS 30019**  
**Introduction to Artificial Intelligence**

**Assignment 2**  
**Inference Engine**  
**Due 4:30 Friday 31<sup>st</sup> May 2019.**

**COS30019\_A02\_T014**  
**Aseem Malhotra 101557400**  
**Peter Micevski 100651602**

## Table of Contents

Topics	Page number
Introduction	3
Data Structure	3
Truth Table	4
Forward Chaining	4
Backward Chaining	5
Resolution – Research project	5
Features and Missing	6
Test Cases	7
References	8

## Introduction

The purpose of this report is to demonstrate the working of the inference engine called **iengine**. The iengine accepts a Knowledge Base (KB) and query (alpha). On the basis of KB it assesses whether alpha is true or not.

*What form of KB does iengine accept?*

The iengine only accepts **Horn form KB**.

*What methods are used in the iengine?*

There are four methods of evaluating the query.

1. Truth table.
2. Forward chaining.
3. Backward chaining.
4. Resolution. (Research project).

## Data Structure

The KB is stored as “Sentences” a class. The class has 3 fields.

Sentences:

String symbol  
String connective  
Sentences[] children

There are 3 types of KB sentences (clauses) and they are stored as...

### Autonomous Clauses (a; p2;)

These are individual symbols which are stored as string symbol and connective and children are null.

### Implies Statements with one Antecedent (a => b)

In this case the symbol is null, connective is “=>”, first child (children [0]) is antecedent and second child (children [1]) is the consequent. Both children have symbol stored as a and b respectively and connective and children as null.

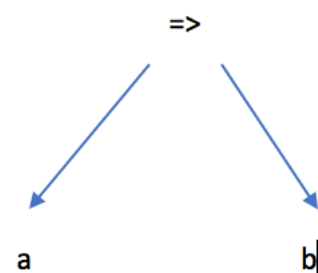
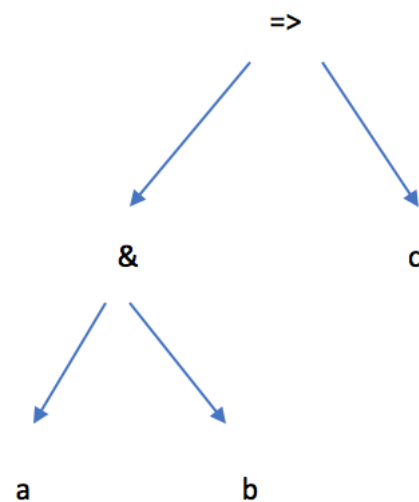


Figure 1: Sentence class with 1 children.

### Implies Statements with two Antecedent ( $a \& b \Rightarrow c$ )

In this case the symbol is null, connective in " $\Rightarrow$ ", first child (children [0]) is antecedent and second child (children [1]) is the consequent. children [1] has symbol stored as c and connective and children as null. children [0] has symbol null, connective "&" and 2 children with symbols a and b.



Query is also stored as a Sentence.

Figure 2: Sentence class with 2 children.

### Truth Table

Truth tables models are built like a binary tree. We start with an empty model. Then for each variable we add two models one where the variable is true and one where variable is false. So, if the number of variables are " $n$ " we end up with  $2^n$  models. For the test case query below we have 11 symbols, p2, p3, p1, c, e, b, f, g, h, a, and b. so we end up with  $2^{11}$  models i.e. 2048 models. for each of the models we use the Boolean value of each symbols to check weather each clause of KB if true. If all clauses of KB are true on each of those occasions we check for query (alpha). If alpha is true on each occasion then **KB entails alpha**  $KB \models \alpha$ . If in any model where KB is true and alpha is false then **KB does not entail alpha**  $KB \not\models \alpha$ .

Test case

$p2 \Rightarrow p3$ ;  $p3 \Rightarrow p1$ ;  $c \Rightarrow e$ ;  $b \& e \Rightarrow f$ ;  $f \& g \Rightarrow h$ ;  $p1 \Rightarrow d$ ;  $p1 \& p3 \Rightarrow c$ ; a; b; p2;

### Forward Chaining

Forward chaining starts with autonomous statements that are true e.g. a, b and p2 in test case. Then we use **Modus Ponens** to prove more autonomous statements to be true.

E.g. if  $p2 \Rightarrow p3$  is true and p2 if true we can infer p3 to be true. If p3 is true and  $p3 \Rightarrow p1$ , p1 is true. If p1 is true and  $p1 \Rightarrow d$ , d is true.

we move forward where we find autonomous statements as antecedent we prove the consequent and add it as an autonomous statement to KB. if in the end  $\alpha$  is inferred true query is true else false. In forward chaining, we may end up inferring statements and that are not needed in inferring query.

## Backward Chaining

Backward chaining starts from the query. If query is not true we look for its antecedent and make it the new query as if antecedent is true and antecedent  $\Rightarrow$  consequent, original query will be true.

E.g.  $d$  is query and  $p1 \Rightarrow d$ . so if  $p1$  is true  $d$  will be true by **Modus Ponens**.  $p3 \Rightarrow p1$  so if  $p3$  is true  $p1$  will be true.  $p2 \Rightarrow p3$  and we know that  $p2$  is autonomous hence  $p2$  is true so  $p3$  is true so  $p1$  is true so  $d$  is true.

## Resolution – Research project

Resolution is proof by contradiction. To prove  $KB \models \alpha$  we prove  $KB \wedge \neg\alpha$  as un-satisfiable. To do this first all clauses in KB are converted into CNF (Conjunctive normal form).

E.g.  $\alpha \Rightarrow \beta$  becomes  $\neg\alpha \vee \beta$

Then contradictions of common variables between clauses are removed by joining them together and forming new clauses.

E.g.  $\neg\alpha \vee \beta \wedge \neg\beta \vee \gamma$  becomes  $\neg\alpha \vee \gamma$

If these new clauses will lead to clause  $\alpha$  that will contradict  $\neg\alpha$  and that will prove  $KB \models \alpha$  by contradiction. For test case, new clauses are formed from existing clauses till an autonomous clause  $d$  is formed which contradicts with our negated alpha  $\neg d$ . Hence, we prove  $d$  to be true.

If all KBs are formed and no new KB can be derived from existing KB and  $KB \wedge \neg\alpha$  is not un-satisfiable then  $KB \not\models \alpha$  and NO will be returned.

The steps for resolution are given below.

**Table 1: Steps in proof by resolution**

Steps	Clauses in KB	Proof
1	$\sim p2 \vee p3$	Given KB
2	$\sim p3 \vee p1$	Given KB
3	$\sim c \vee e$	Given KB
4	$\sim b \vee \sim e \vee f$	Given KB
5	$\sim f \vee \sim g \vee h$	Given KB
6	$\sim p1 \vee d$	Given KB
7	$\sim p1 \vee \sim p3 \vee c$	Given KB
8	A	Given KB
9	B	Given KB
10	P2	Given KB
11	$\sim d$	Negated alpha
12	$\sim p2 \vee p1$	1 & 2
13	p3	1 & 10
14	$\sim p3 \vee d$	2 & 6
15	$\sim p2 \vee d$	1 & 14
16	d	10 & 15
17	.	11 & 16

### Features and Missing

1. The program has an additional method of resolution. On its completion if result is YES it will print to screen a list of all the KBs that were given to it and that it derived to prove KB entails alpha.
2. The program only works for Horn form knowledge base and not for general form KB.

## Test Cases

The following test cases were performed on the program the result achieved was is below. the resolution method will print all the KBs old and new if result is YES.

### Test cases: 1

$p2 \Rightarrow p3$ ;  $p3 \Rightarrow p1$ ;  $c \Rightarrow e$ ;  $b \& e \Rightarrow f$ ;  $f \& g \Rightarrow h$ ;  $p1 \Rightarrow d$ ;  $p1 \& p3 \Rightarrow c$ ;  $a$ ;  $b$ ;  $p2$ ;

**Table 2: results of test case 1 for different alpha**

ASK	TT	Forward Chaining	Backward Chaining	Resolution
d	YES: 3	YES: a b p2 p3 p1 d	Yes: p2 p3 p1 d	YES
f	YES: 3	YES: a b p2 p3 p1 d c e f	Yes: p2 p3 p1 c e b f	YES
m	NO	NO	NO	NO
g	NO	NO	NO	NO
a	YES: 3	YES: a	YES: a	YES
c	YES: 3	YES: a b p2 p3 p1 d c	YES: p2 p3 p1 c	YES
h	NO	NO	NO	NO

### Test cases: 2

$b \Rightarrow a$ ;  $b$

**Table 3: results of test case 2 for different alpha**

ASK	TT	Forward Chaining	Backward Chaining	Resolution
a	YES: 1	YES: b a	YES: b a	YES
b	YES: 1	YES: b	YES: b	YES
c	NO	NO	NO	NO

### Test cases: 3

$b \Rightarrow a$ ;  $c \Rightarrow b$ ;  $b$

**Table 4: results of test case 3 for different alpha**

ASK	TT	Forward Chaining	Backward Chaining	Resolution
a	YES: 2	YES: b a	YES: b a	YES
b	YES: 2	YES: b	YES: b	YES
c	NO	NO	NO	NO

## References

Russell S, Norvig P, Artificial Intelligence A Modern Approach third edition, 2010.

*How this book helped me?*

It's the main book prescribed for this unit. It has explanation of each of the search method implemented in this report.

Stack Overflow, <https://stackoverflow.com>

*How this website helped me?*

Stack overflow is a useful website to ask questions about coding and was very useful when we got stuck.

<https://stackoverflow.com/questions/56286830/truth-table-using-recursion>

MIT university Lecture Slides on Resolution.

<https://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-825-techniques-in-artificial-intelligence-sma-5504-fall-2002/lecture-notes/Lecture7FinalPart1.pdf>

*How this website helped me?*

The following are lecture slides on resolution and were very helpful in understanding resolution method in details.

TT-ENTAILS: Inference by Enumeration in Proposition Logic, by Vassilis Athitsos, University of Texas at Arlington.

[http://vlm1.uta.edu/~athitsos/courses/cse4308\\_fall2016/lectures/03a\\_tt\\_entails.pdf](http://vlm1.uta.edu/~athitsos/courses/cse4308_fall2016/lectures/03a_tt_entails.pdf)

*How this website helped me?*

The following are lecture slides on truth table and were very helpful in understanding the data structure and pseudocode for truth table.

Artificial Intelligence A Modern Approach

<http://aima.cs.berkeley.edu>

*How this website helped me?*

The following website is the website of the prescribed book Artificial Intelligence A Modern Approach and has lots of resources used to complete this assignment.