Automated Reasoning

Guofeng Cui

October 25, 2018

Abstract

In the project, *TT-Entail* and *DPLL-Satisfiable* are implemented to do automated reasoning solving seven problems. Both of the two algorithms are based on truth table. Performance of them are also compared, which shows a situation based result, including size of knowledge base, construction of *conjunctive normal formal (CNF) clauses*, etc..

1 Introduction

Two algorithms are applied to solve automated reasoning problems, including *TT-Entail* and *DPLL-Satisfiable.* The details of such algorithms are shown in section 2.

To analyze the performance of such algorithms, seven automated reasoning problems are used for test, such as *Modus Ponens*, *Simple Wumpus World*, *Horn Clauses*, *Liars and Truth-tellers*, *More Liars and Truth-tellers*, *The Doors of Enlightenment* and *Complete Wumpus World.* The problems is discussed in section 3.

Among the problems above, the first six problems are utilized to analyze the performance of algorithms, which will be shown in section 5. And the purpose of the last problem is to apply algorithms in real world, which is also shown in section 5.

Section 4 shows the construction of code and the final section represents some thoughts about improvement of *TT-Entail* and *DPLL-Satisfiable.*

2 Algorithm

2.1 TT-Entail

One way to prove entailment is applying truth table. Check the statement that whether conclusion will always be truth under the situation that knowledge base is true. If the statement is correct, one can prove that knowledge entails the conclusion, that is $KB \models \alpha$, where KB refers to sentence of knowledge base and α refers to sentence of conclusion. But if the statement is incorrect, the conclusion needs to be negatived to prove $KB \models \neg\alpha$ generally.

Iteration can be utilized to realize such algorithm. In the first step, the algorithm break both sentences of knowledge base and conclusion into symbols. Then apply logical value (truth or false) to each symbol, and check the logical value of knowledge base and conclusion. Notice that whenever $KB = true$,

$\alpha = true$, so that when there appears a situation when $KB = true$, $\alpha = false$, entailment would be false. That is the algorithm can stop and return false. Moreover, to make the code of algorithm readable, both KB and α are translated into clauses at first. Another two advantages of such translation are that, (1) easy to analyze the truth value of clause, that is whenever a symbol contained is truth, such clause can always be true, for the reason that only *disjunction* is used inside a clause; (2) easier to construct the code, as no need to consider the logical condition of logical symbols other than *not*, *disjunction* and *conjunction*, such as *implies*, *if and only if*.

2.2 DPLL-Satisfiable

Although *DPLL-Satisfiable* also make use of truth table and iteration, its idea is total different from that of *TT-Entail*. If *TT-Entail* can be seen as prove of validity, *DPLL-Satisfiable* bases on prove of satisfiability. That is, to prove $KB \models \alpha$, *DPLL-Satisfiable* turns to prove $KB \wedge \neg\alpha$ can never be satisfiable.

The advantage of satisfiable proving is that once a clause is true, the algorithm can terminate for the reason that satisfiable has been proved. Comparing with proving a clause to be false, proving true can be much more easier and terminate in earlier loop, as it does not require the logical value of all symbols in most time. Moreover, *DPLL-satisfiable* further improves the algorithm by exploiting two heuristics: finding pure symbol and finding unit clause. A symbol is pure whenever it appears only positive or negative in the entire clause set. At this point, it can be valued directly to make clauses to be true, as the algorithm only deal with clauses and the purpose is to check whether true value exists. A clause is unit when it only depends on a single symbol, and similarity, it can be set to true.

3 Automated Reasoning Problem

As it is assumed that CNF is directly input into algorithm, CNF of each problem are shown in this section. Also, the section will include discussion of several problems.

3.1 Modus Ponens

Modus Ponens problem is pretty simple, which is to check whether $\{P, P \Rightarrow Q\} \models Q$ is true. The CNF of knowledge base is shown as below:

$$\begin{gathered} P \\ \neg P \vee Q \end{gathered}$$

And the CNF of conclusion is:

$$Q$$

3.2 Simple Wumpus World

Simple Wumpus World is quite more difficult than *Modus Ponens*, the knowledge base (both Propositional Logic sentences and CNF) is shown as table 1,

sentence	**CNF**
$\neg P_{1,1}$	$\neg P_{1,1}$
$B_{1,1} \Leftrightarrow (P_{1,2} \vee P_{2,1})$	$\neg B_{1,1} \vee P_{1,2} \vee P_{2,1}$ $\neg P_{1,2} \vee B_{1,1}$ $\neg P_{2,1} \vee B_{1,1}$
$B_{2,1} \Leftrightarrow (P_{1,1} \vee P_{2,2} \vee P_{3,1})$	$\neg B_{2,1} \vee P_{1,1} \vee P_{2,2} \vee P_{3,1}$ $\neg P_{1,1} \vee B_{2,1}$ $\neg P_{2,2} \vee B_{2,1}$ $\neg P_{3,1} \vee B_{2,1}$
$\neg B_{1,1}$	$\neg B_{1,1}$
$B_{2,1}$	$B_{2,1}$

Table 1: Knowledge Base of Simple Wumpus World.

sentence	**CNF**
$P_{1,2}$	$P_{1,2}$
$\neg P_{1,2}$	$\neg P_{1,2}$

Table 2: Conclusion of Simple Wumpus World.

and the conclusion is shown in table 2. Notice that both $P_{1,2}$ and $\neg P_{1,2}$ are tested, the reason is that both $KB \models P_{1,2}$ and $KB \models \neg P_{1,2}$ needs to be check. If both two statements are incorrect, one can come into conclusion that based on current knowledge base, the conclusion can not be determined.

3.3 Horn Clauses

The first step to sove *Horn Clauses* problem is to translate it into propositional logical sentence, as shown below in table 3 and table 4:

3.4 Liars and Truth-tellers

Similarly, knowledge base and conclusion of the first problem of *Liars and Truth-tellers* are shown in table 5 and table 7. Notice that $\neg Amy \vee Amy$ and $\neg Cal \vee \neg Amy \vee Amy$ in table 5 will always be true, and both *TT-entail* and *DPLL-satisfiable* do not check such situations, they can be deleted directly before input. The knowledge base and conclusion of the second problem of *Liars and Truth-tellers* are shown in table 6 and table 7. Notice that the two problems check the same conclusion.

sentence	**CNF**
$Mythical(U) \Rightarrow Immortal(U)$	$\neg Mythical(U) \vee Immortal(U)$
$\neg Mythical(U) \Rightarrow \neg Immortal(U) \wedge mammal(U)$	$Mythical(U) \vee \neg Immortal(U)$ $Mythical(U) \vee mammal(U)$
$Immortal(U) \vee mammal(U) \Rightarrow Horned(U)$	$\neg Immortal(U) \vee Horned(U)$ $\neg mammal(U) \vee Horned(U)$
$Horned(U) \Rightarrow Magical(U)$	$\neg Horned(U) \vee Magical(U)$

Table 3: Knowledge Base of Horn Clauses Problem

sentence	**CNF**
$Mythical(U)$	$Mythical(U)$
$\neg Mythical(U)$	$\neg Mythical(U)$
$Magical(U)$	$Magical(U)$
$\neg Magical(U)$	$\neg Magical(U)$
$Horned(U)$	$Horned(U)$
$\neg Horned(U)$	$\neg Horned(U)$

Table 4: Conclusion of Horn Clauses Problem

sentence	**CNF**
$Amy \Leftrightarrow Cal \wedge Amy$	$\neg Amy \vee Cal$
	$\neg Amy \vee Amy$
	$\neg Cal \vee \neg Amy \vee Amy$
$Bob \Leftrightarrow \neg Cal$	$\neg Bob \vee \neg Cal$
	$Cal \vee Bob$
$Cal \Leftrightarrow Bob \vee \neg Amy$	$\neg Cal \vee Bob \vee \neg Amy$
	$\neg Bob \vee Cal$
	$Amy \vee Cal$

Table 5: Knowledge Base of First Problem of Liar and Truth-tellers

sentence	**CNF**
$Amy \Leftrightarrow \neg Cal$	$\neg Amy \vee \neg Cal$
	$Cal \vee Amy$
$Bob \Leftrightarrow Amy \wedge Cal$	$\neg Bob \vee Amy$
	$\neg Bob \vee Cal$
	$\neg Amy \vee \neg Cal \vee Bob$
$Cal \Leftrightarrow Bob$	$\neg Cal \vee Bob$
	$\neg Bob \vee Cal$

Table 6: Knowledge Base of Second Problem of second problem Liar and Truth-tellers

sentence	**CNF**
Amy	Amy
$\neg Amy$	$\neg Amy$
Cal	Cal
$\neg Cal$	$\neg Cal$
Bob	Bob
$\neg Bob$	$\neg Bob$

Table 7: Conclusion of Liar and Truth-tellers

3.5 More Liars and Truth-tellers

More Liars and Truth-tellers problem is a more complex version of *Liars and Truth-tellers* problem, the knowledge base and conclusion of which are shown in table 8 and table 9.

3.6 The Doors of Enlightenment

Two problems are included in *The Doors of Enlighenment* problem, the knowledge bases of which are shown in table 10 and table 11. Similar to *Liar and Truth-tellers* problem, clauses such as $\neg C \vee \cdots \vee C \vee \cdots$ exists. At this point, they are deleted and not represents in tables.

Also, the problems share the same conclusion, which is shown in table 12. The philosopher in the problem only want to choose a suitable door, which means only doors exactly to be safe should be chosen. So only $KB \models safeDoor$ needs to be checked.

3.7 Complete Wumpus World

Complete Wumpus World is a complete different problem from the six problems above. Solving such problem, one needs to build up an agent to explore the partially observable world. Whenever the agent moves to a new position, it needs to preserve the world, infer status of each position of the world, and choose its action.

To solve the problem, the *wumpus world* is defined as below:
(1)**States**: State can be defined by $\{arrow, position, sense, status\}$, where $arrow$ is define as $\{1, 0\}$, indicates whether agent has arrow; $position$ is defined as (X, Y), X indicates row of world, Y indicates column of world; $sense$ is defined as $\{stench, breeze, glitter, bump, scream\}$, $stench$ means there may be a wumpus in position $(X \pm 1, Y)$ or $(X, Y \pm 1)$, $breeze$ means there may be a pit in position $(X \pm 1, Y)$ or $(X, Y \pm 1)$, $glitter$ means a gold in position (X, Y), $bump$ means agent remain in previous position (X, Y), $scream$ means a wumpus in position (X, Y); $status$ is defined as $\{gold, wumpus, git, safe\}$.
(2)**Actions**: Actions include $\{feel, grab, shoot, move\}$.
(3)**Translate Model**: $feel$ action return a $sense$, $grad$ get the gold of current position if there has one, $shoot$ shoot the wumpus of current position if there has one, $move$ get agent into certain place, that is position $(X \pm 1, Y)$ or $(X, Y \pm 1)$.
(4)**Goal test**: If agent stands in position of status $gold$ and do the action $grab$, then agent win.
(5)**Cost**: each position moving takes same cost; shoot cost a loss of $arrow$, cost more than position move; falling into a pit or killed by a wumpus cost most, directly to end the game with loss.
(6)**Initial state**: Agent starts at position $(0, 0)$ and senses the environment.

AIMA Figure 7.20 gives an algorithm, *Hybrid-Wumpus-Agent*, to solve such problem. According to that, the agent collects axioms into knowledge base in the very beginning, and inputs percept sentences into knowledge base each time it gets. However, some improvement can be made to it.
(1) During the update of knowledge base, *Hybrid-Wumpus-Agent* stores its action and sensing. Then for each time the agent needs to do an action, it needs to infer based on action and initial knowledge base. This could be a waste of time,

sentence	**CNF**
$Amy \Leftrightarrow Hal \wedge Ida$	$\neg Amy \vee Hal$
	$\neg Amy \vee Ida$
	$\neg Hal \vee \neg Ida \vee Amy$
$Bob \Leftrightarrow Amy \wedge Lee$	$\neg Bob \vee Amy$
	$\neg Bob \vee Lee$
	$\neg Amy \vee \neg Lee \vee Bob$
$Cal \Leftrightarrow Bob \wedge Gil$	$\neg Cal \vee Bob$
	$\neg Cal \vee Gil$
	$\neg Bob \vee \neg Gil \vee Cal$
$Dee \Leftrightarrow Eli \wedge Lee$	$\neg Dee \vee Eli$
	$\neg Dee \vee Lee$
	$\neg Eli \vee \neg Lee \vee Dee$
$Eli \Leftrightarrow Cal \wedge Hal$	$\neg Eli \vee Cal$
	$\neg Eli \vee Hal$
	$\neg Cal \vee \neg Hal \vee Eli$
$Fay \Leftrightarrow Dee \wedge Ida$	$\neg Fay \vee Dee$
	$\neg Fay \vee Ida$
	$\neg Dee \vee \neg Ida \vee Fay$
$Gil \Leftrightarrow \neg Eli \wedge \neg Jay$	$\neg Gil \vee \neg Eli$
	$\neg Gil \vee \neg Jay$
	$Eli \vee Jay \vee Gil$
$Hal \Leftrightarrow \neg Fay \wedge \neg Kay$	$\neg Hal \vee \neg Fay$
	$\neg Hal \vee \neg Kay$
	$Fay \vee Kay \vee Hal$
$Ida \Leftrightarrow \neg Gil \wedge \neg Kay$	$\neg Ida \vee \neg Gil$
	$\neg Ida \vee \neg Kay$
	$Gil \vee Kay \vee Ida$
$Jay \Leftrightarrow \neg Amy \wedge \neg Cal$	$\neg Jay \vee \neg Amy$
	$\neg Jay \vee \neg Cal$
	$Amy \vee Cal \vee Jay$
$Kay \Leftrightarrow \neg Dee \wedge \neg Fay$	$\neg Kay \vee \neg Dee$
	$\neg Kay \vee \neg Fay$
	$Dee \vee Fay \vee Kay$
$Lee \Leftrightarrow \neg Bob \wedge \neg Jay$	$\neg Lee \vee \neg Bob$
	$\neg Lee \vee \neg Jay$
	$Bob \vee Jay \vee Lee$

Table 8: Knowledge Base of More Liars and Truth-tellsers

sentence	**CNF**
Amy	Amy
$\neg Amy$	$\neg$Amy
Hal	Hal
$\neg Hal$	$\neg Hal$
Ida	Ida
$\neg Ida$	$\neg Ida$
Bob	Bob
$\neg Bob$	$\neg Bob$
Lee	Lee
$\neg Lee$	$\neg Lee$
Cal	Cal
$\neg Cal$	$\neg Cal$
Gil	Gil
$\neg Gil$	$\neg Gil$
Dee	Dee
$\neg Dee$	$\neg Dee$
Eli	Eli
$\neg Eli$	$\neg Eli$
Fay	Fay
$\neg Fay$	$\neg Fay$
Jay	Jay
$\neg Jay$	$\neg Jay$
Kay	Kay
$\neg Kay$	$\neg Kay$

Table 9: Conclusion of More Liars and Truth-teller

sentence	**CNF**
$A \Leftrightarrow X$	$\neg A \vee X$
	$\neg X \vee A$
$B \Leftrightarrow Y \vee Z$	$\neg B \vee Y \vee Z$
	$\neg Y \vee B$
	$\neg Z \vee B$
$C \Leftrightarrow A \wedge B$	$\neg A \vee \neg B \vee C$
	$\neg C \vee A$
	$\neg C \vee B$
$D \Leftrightarrow X \wedge Y$	$\neg X \vee \neg Y \vee D$
	$\neg D \vee X$
	$\neg D \vee Y$
$E \Leftrightarrow X \wedge Z$	$\neg X \vee \neg Z \vee E$
	$\neg E \vee X$
	$\neg E \vee Z$
$F \Leftrightarrow (D \wedge \neg E) \vee (\neg D \wedge E)$	$\neg F \vee \neg E \vee \neg D$
	$\neg F \vee E \vee D$
	$D \vee \neg E \vee F$
	$\neg D \vee E \vee F$
$G \Leftrightarrow (C \Rightarrow F)$	$\neg G \vee \neg C \vee F$
	$C \vee G$
	$\neg F \vee G$
$H \Leftrightarrow ((G \wedge H) \Rightarrow A)$	$\neg H \vee \neg G \vee A$
	$G \vee H$
	H
	$\neg A \vee H$

Table 10: Knowledge Base of problem 1 of The Doors of Enlightenment

sentence	**CNF**
$A \Leftrightarrow X$	$\neg A \vee X$
	$\neg X \vee A$
$H \Leftrightarrow ((G \wedge H) \Rightarrow A)$	$\neg H \vee \neg G \vee A$
	$G \vee H$
	H
	$\neg A \vee H$
$C \Leftrightarrow A \wedge (B \vee C \vee D \vee E \vee F \vee G \vee H)$	$\neg C \vee A$
	$\neg A \vee \neg B \vee C$
	$\neg A \vee \neg D \vee C$
	$\neg A \vee \neg E \vee C$
	$\neg A \vee \neg F \vee C$
	$\neg A \vee \neg G \vee C$
	$\neg A \vee \neg H \vee C$
$G \Leftrightarrow C \Rightarrow \cdots$	$C \vee G$

Table 11: Knowledge Base of Problem 2 of The Doors of Enlightenment

sentence	**CNF**
X	X
Y	Y
Z	Z
W	W

Table 12: Conclusion of The Doors of Enlightenment

as the agent throw the status of the world each time it gets it and the same inferences have to be implemented repeatedly. Instead of putting actions and senses into knowledge base, agent can focus on current and stores the status of positions, such as safe positions, positions which have pit, etc.. At this points, inference based on each action and initial knowledge base is implemented for only one time, that is when the agent updates current map. And the time consuming of selecting an action could be quite less for the reason that only status of positions are used and all of them are unit clauses. Moreover, the agent only keeps checking the status of the world and throws the action and sense it took, the space consuming of knowledge base would be at most $axiom\_size + map\_size$, where $axiom\_size$ refers to the size of axioms, map_size refers to the size of map, comparing with $axiom\_size + action * t$ of *Hybrid-Wumpus-Agent.*
(2) For each time of searching, *Hybrid-Wumpus-Agent* makes use of the entire knowledge base. This would be a waste of time and space, as only few clauses are used during the search actually. With the first improvement, agent only utilizes the status of map while selecting actions. Improvement can also be made while updating the map. In reality, when a human being implements inference, it only explores related knowledge. By organizing knowledge bases into several subsets, the agent can only utilizes related knowledge when doing certain inference.

Based on the improvements above, a new algorithm is used. The knowledge are divided into three sets. The first set is related to moving, as shown in table 13. The second set is related to checking status based on sensing, as shown in table 14. The third set is related to special action, including grabbing, and shooting, as shown in table 15. In practice, knowledge base would be larger than that shown in such tables. It could be better if X and Y of clauses could be set by agent. At this point, during each time of inferring or searching, only usable clauses would be selected from certain subset. Moreover, for each time of updating the map, statuses of positions in the map are stored, values of which are select among $\{safe, visited, pit, wumpus, may\_pit, may\_wumpus\}$. Notice that there may exists certain positions with status containing two of values above, for example, there may have either pit or wumpus, but the agent doesn't know which one would be at that position. To handle such situations, it can be assumed that pit is more dangerous than wumpus, as agent may have arrow. Hence when may_pit is marked in a position, may_wumpus will never appear there.

sentence
$L_{X,Y} \Rightarrow (L_{X+1,Y} \wedge U) \vee (L_{X-1,Y} \wedge D) \vee (L_{X,Y+1} \wedge R) \vee (L_{X,Y-1} \wedge L)$
$\neg U \vee \neg D$
$\neg U \vee \neg R$
$\ldots$

Table 13: moving set

sentence
$stench_{X,Y} \Leftrightarrow W_{X-1,Y} \vee W_{X+1,Y} \vee W_{X,Y-1} \vee W_{X,Y+1}$
$breeze_{X,Y} \Leftrightarrow P_{X-1,Y} \vee P_{X+1,Y} \vee P_{X,Y-1} \vee P_{X,Y+1}$
$glitter_{X,Y} \Leftrightarrow G_{X,Y}$
$scream_{X,Y} \Leftrightarrow W_{X,Y}$
$safe_{X,Y} \Leftrightarrow \neg W_{X,Y} \wedge \neg P_{X,Y}$
$\neg W_{X,Y} \vee \neg P_{X,Y}$
$\neg W_{X,Y} \vee \neg G_{X,Y}$
$\neg P_{X,Y} \vee \neg G_{X,y}$

Table 14: check status set

4 Construction of Code

4.1 Basic Problem

The construction of code is shown in UML figure 1. The code is divided into three group, execution group (Exec), algorithm group (Alg) and knowledge group (KBG).
(1)**Execution group** implements $TT-Entail$ and $DPLL-Satisfiable$ algorithms to solve the basic six problems (exclude *Wumpus World*).
(2)**Algorithm group** realizes $TT-Entail$ and $DPLL-Satisfiable$ algorithms, which solve the problems containing CNF. TTE class contains the idea of $TT-Entail$ and $DPLL$ class contains the idea of $DPLL-Satisfiable$.
(3)**Knowledge group** contains the structure of CNF data and realizes the functions to check whether a clause or sentence is true.

4.2 Wumpus World

The construction of code for *Wumpus World* is shown in UML figure 2. Three types of classes work for three kinds of functions.
(1)Class *WumpusWorld* acts as environment, which gives the status of a certain position, such as stench, breeze, etc.. And it contains the status of entire world, including player, wumpus, and so on.
(2)Class *WumpusUI* deals with interface between player and computer, which

sentence
$arrow \wedge \neg shoot \Leftrightarrow newArrow$
$G_{X,Y} \wedge Grab \Rightarrow \neg newGold_{X,Y}$

Table 15: special action set

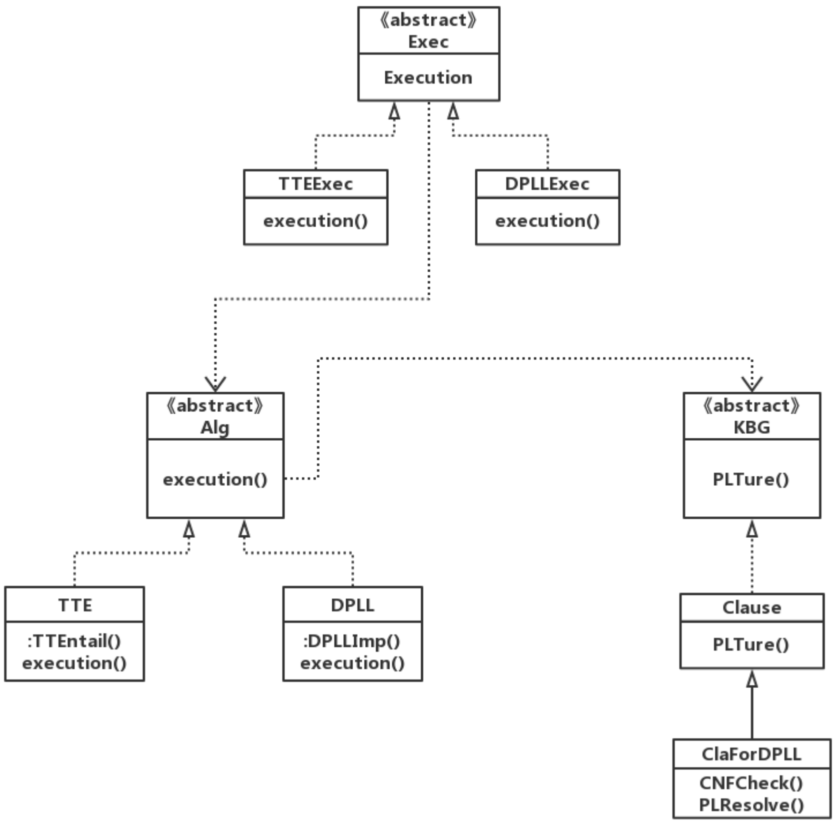


Figure 1: UML Figure for Basic Problem

is used mainly when user in real world is playing the game. It includes functions showing the board, asking user to select an action, and telling user the state of the game.
(3)Both classes *WumpusImp* and *WumpusAgent* can be used to play the game. The former is called when user want to play the game by himself, while the latter is called when user want computer to play the game, which implement the algorithm shown in section 3.7.

5 Experiment and Analysis

5.1 Results and Comparison

The experiment can be separated into two steps. The first step is implementing two algorithms in six basic problems to prove the statement. The second step is implementing entire program for 5 times, can compare the performances of two algorithms.

The result of *Modus Ponens* problem is shown in figure 3, which represents that Q is always true when the knowledge base is true.

The result of *simple Wumpus World* problem is shown in figure 4. Based on the figure, one can see that $P_{1,2}$ is prove to be false, which means that $KB \models \neg P_{1,2}$ can be proved.

The result of *Horn Clauses* problem is shown in figure 5. It appears that unicorn can not be determined whether it is "Mythical". But it does can be determined to be "Magical" and "Horned".

The result of *Liars and Truth-tellers* problem is shown in figure 6. For the first problem, it comes into conclusion that Amy and Bob are liars, and Cla is a truth-teller. For the second problem, the result is that Amy is the only truth-teller, and Bob as well as Cal are both liars.

The result of *More Liars and Truth-tellers* problem is shown in figure 7. We can see that only Jay and Kay are truth-tellers, the rest ones are liars.

The result of *The Doors of Enlightenment* problem is shown in figure 8. For both the first and second problems, X door can be determined to be the safe one, however the state of rest of three can not be determined.

After proving the six problems, each algorithm are implemented five times for each problem, the result is shown in figure 9. The result in grey is the best average result of each problem. It shows that for *Modus Ponens* problem and *More Liars and Truth-tellers* problem, *DPLL-Satisfiable* performs better than *TT-Entail*. *TT-Entail* performs better in other problems.

Based only on the six problems, the comparison may not be powerful, but some trends or potential conditions related to performance can still be analyzed. The reason that *DPLL-Satisfiable* performs better in the first problem is that it contains only unit clause (or can be translated into). *DPLL-Satisfiable* takes at most two loops to finish the calculation by applying heuristic, compared with four loops of *TT-Entail*. However, this could also be the shortcoming of *DPLL-Satisdfiable*, as the heuristic and terminate determination take time. Especially when clauses are badly constructed. However, when the amount of clauses get larger, the time consuming of *TT-Entail* increases exponentially. But it has more chance for pure symbols and unit clauses to appear, and the early terminate determination of *DPLL-Satisfiable* can also be satisfied. That

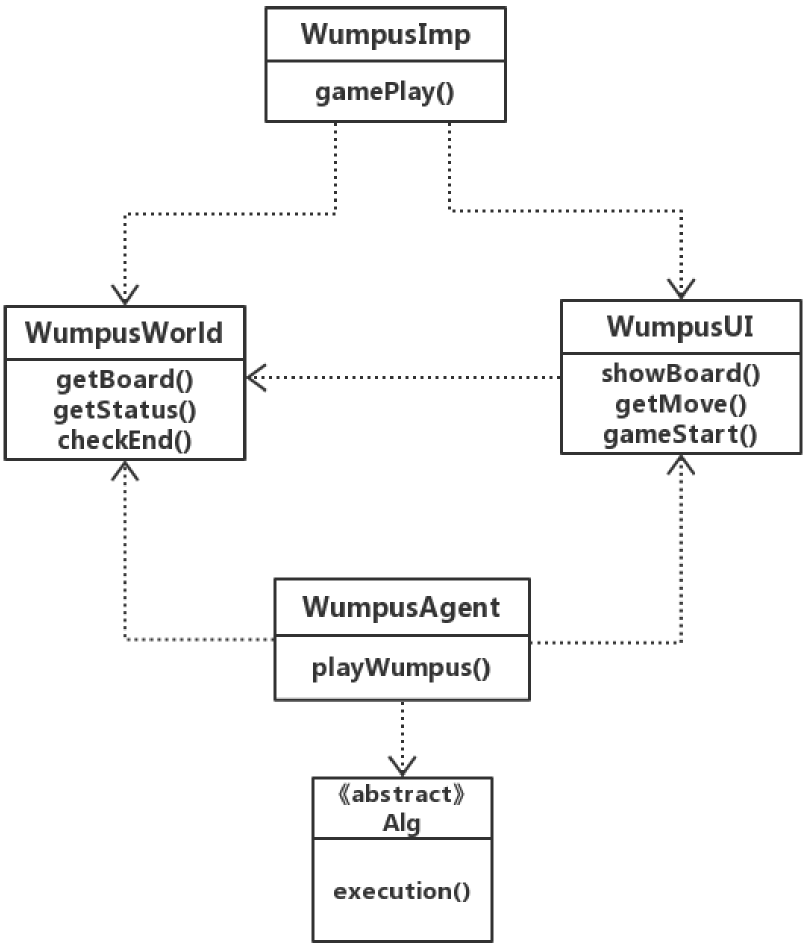
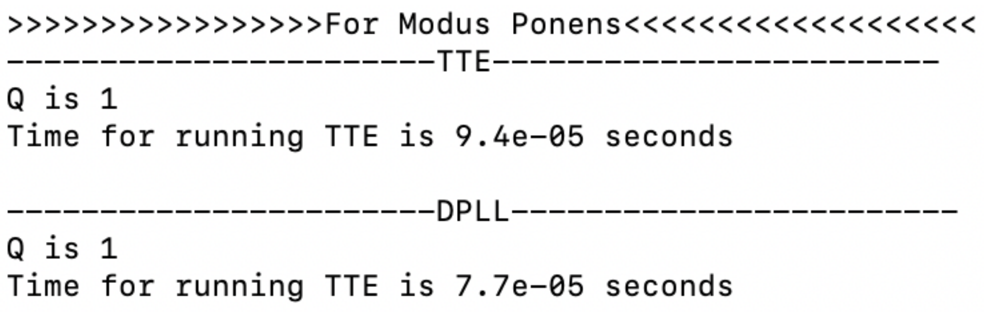
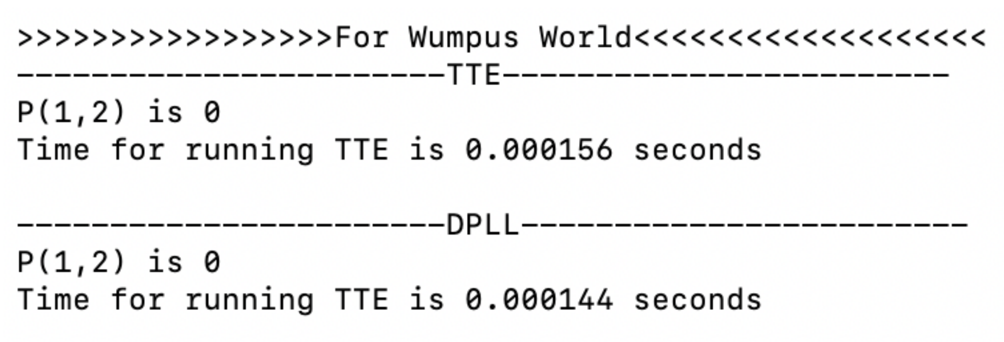


Figure 2: UML Figure for Wumpus World



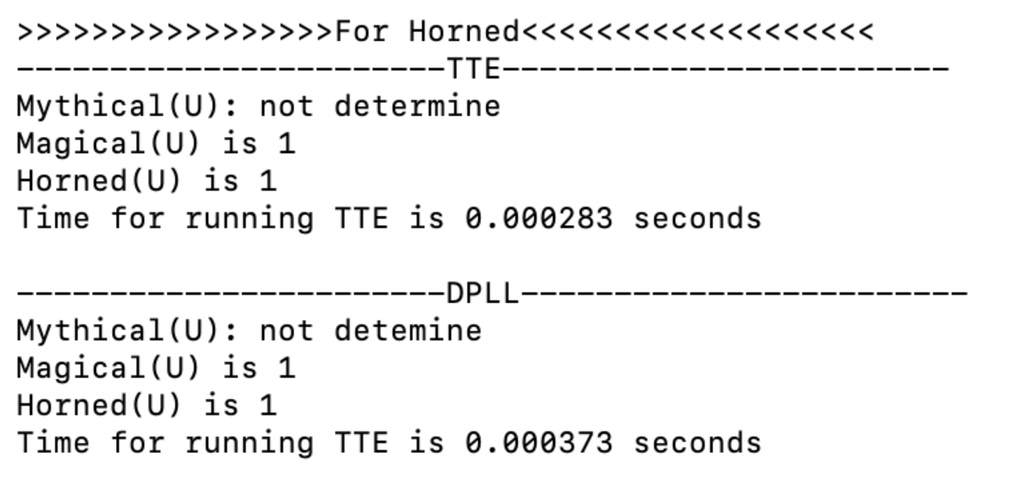
```
>>>>>>>>>>>>>>>>>For Modus Ponens<<<<<<<<<<<<<<<<<<<
-----------------------TTE------------------------
Q is 1
Time for running TTE is 9.4e-05 seconds

-----------------------DPLL------------------------
Q is 1
Time for running TTE is 7.7e-05 seconds
```

Figure 3: Result of Modus Ponens problem



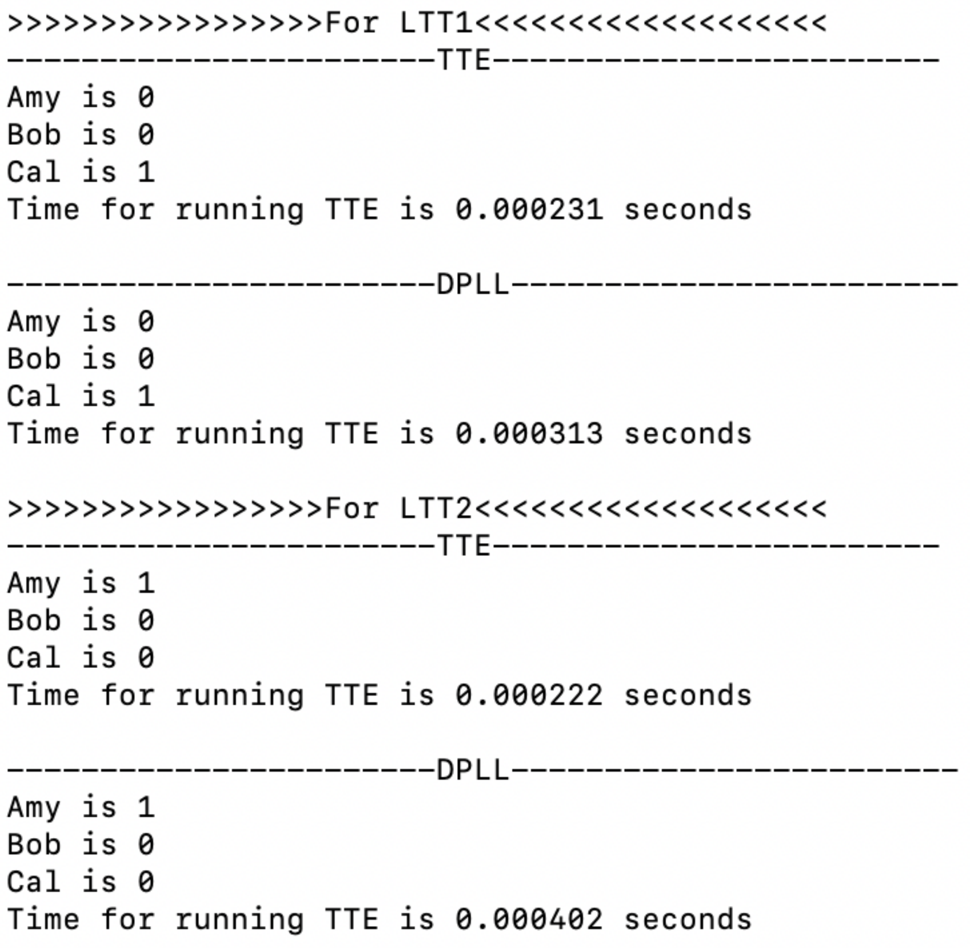
```
>>>>>>>>>>>>>>>>>For Wumpus World<<<<<<<<<<<<<<<<<<<
-----------------------TTE------------------------
P(1,2) is 0
Time for running TTE is 0.000156 seconds

-----------------------DPLL------------------------
P(1,2) is 0
Time for running TTE is 0.000144 seconds
```

Figure 4: Result of Simple Wumpus World Problem



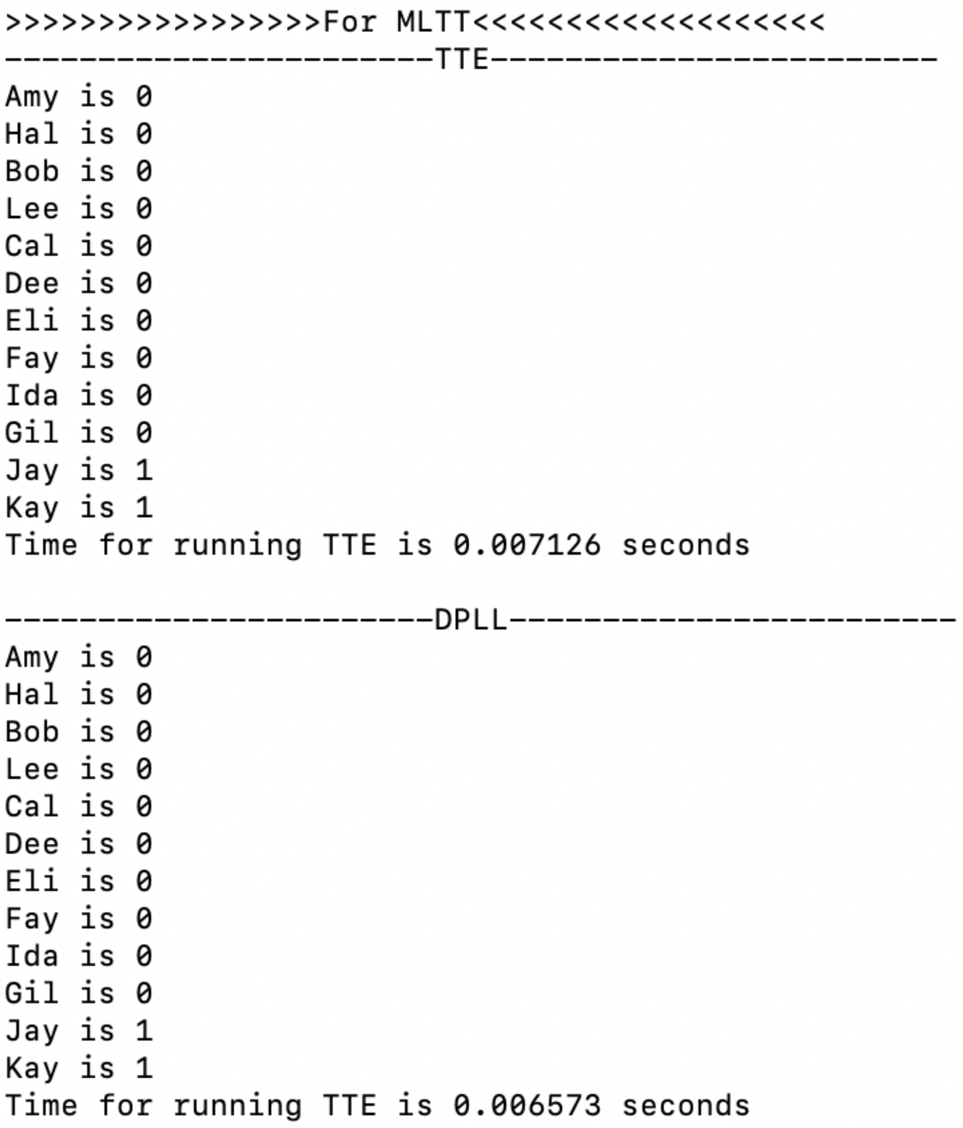
```
>>>>>>>>>>>>>>>>>For Horned<<<<<<<<<<<<<<<<<<<
-----------------------TTE------------------------
Mythical(U): not determine
Magical(U) is 1
Horned(U) is 1
Time for running TTE is 0.000283 seconds

-----------------------DPLL------------------------
Mythical(U): not detemine
Magical(U) is 1
Horned(U) is 1
Time for running TTE is 0.000373 seconds
```

Figure 5: Result of Horn Clauses Problem



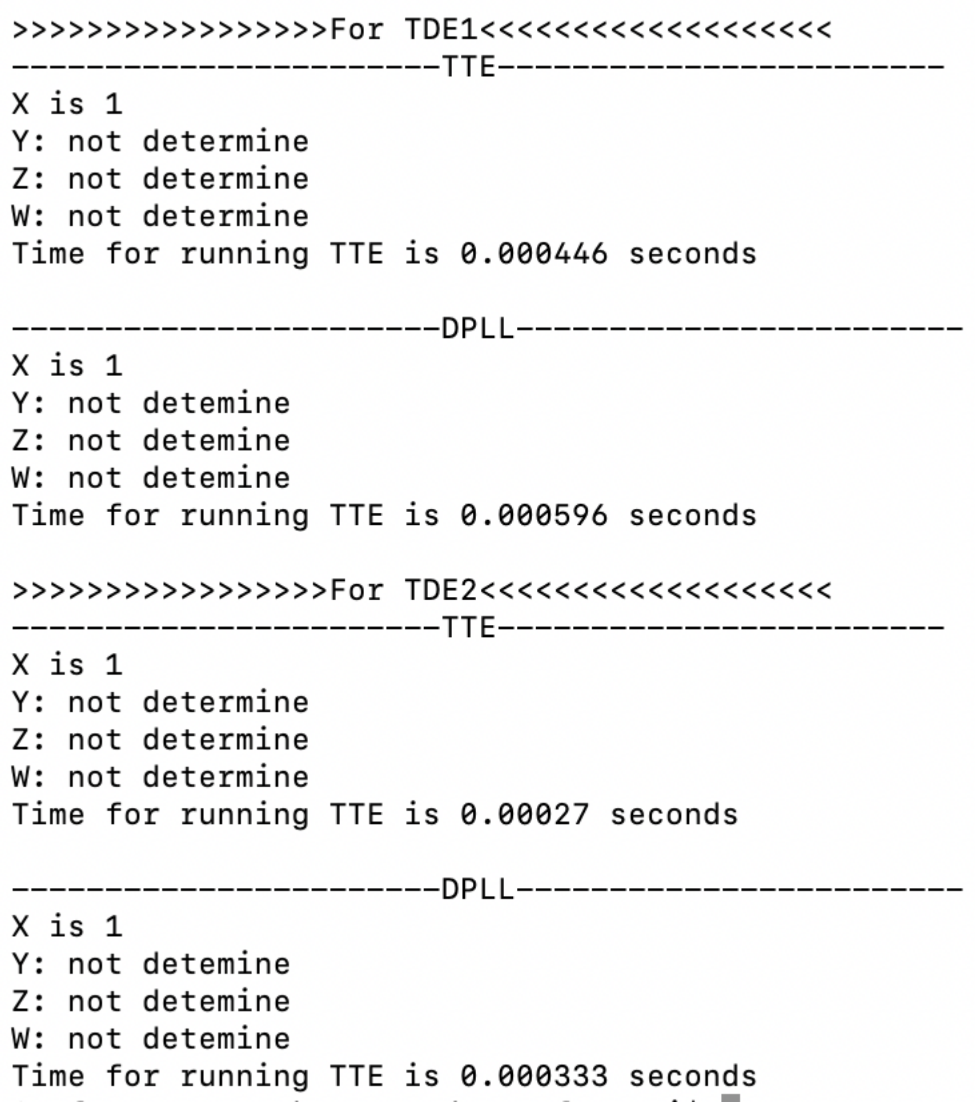
```
>>>>>>>>>>>>>>>>>For LTT1<<<<<<<<<<<<<<<<<<<
-----------------------TTE------------------------
Amy is 0
Bob is 0
Cal is 1
Time for running TTE is 0.000231 seconds

-----------------------DPLL------------------------
Amy is 0
Bob is 0
Cal is 1
Time for running TTE is 0.000313 seconds

>>>>>>>>>>>>>>>>>For LTT2<<<<<<<<<<<<<<<<<<<
-----------------------TTE------------------------
Amy is 1
Bob is 0
Cal is 0
Time for running TTE is 0.000222 seconds

-----------------------DPLL------------------------
Amy is 1
Bob is 0
Cal is 0
Time for running TTE is 0.000402 seconds
```

Figure 6: Result of Liars and Truth-tellers problem



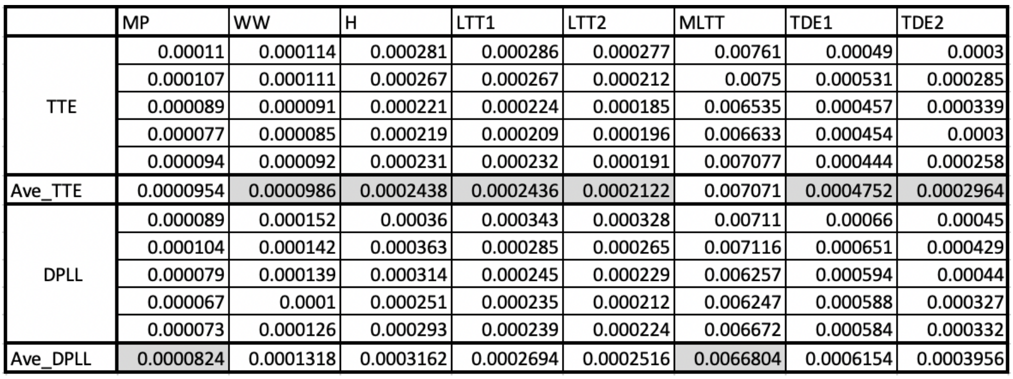
```
>>>>>>>>>>>>>>>>>For MLTT<<<<<<<<<<<<<<<<<<<<
-----------------------TTE------------------------
Amy is 0
Hal is 0
Bob is 0
Lee is 0
Cal is 0
Dee is 0
Eli is 0
Fay is 0
Ida is 0
Gil is 0
Jay is 1
Kay is 1
Time for running TTE is 0.007126 seconds

-----------------------DPLL-----------------------
Amy is 0
Hal is 0
Bob is 0
Lee is 0
Cal is 0
Dee is 0
Eli is 0
Fay is 0
Ida is 0
Gil is 0
Jay is 1
Kay is 1
Time for running TTE is 0.006573 seconds
```

Figure 7: Result of More Liars and Truth-tellers



```
>>>>>>>>>>>>>>>>>For TDE1<<<<<<<<<<<<<<<<<<<
-----------------------TTE------------------------
X is 1
Y: not determine
Z: not determine
W: not determine
Time for running TTE is 0.000446 seconds

-----------------------DPLL------------------------
X is 1
Y: not detemine
Z: not detemine
W: not detemine
Time for running TTE is 0.000596 seconds

>>>>>>>>>>>>>>>>>For TDE2<<<<<<<<<<<<<<<<<<<
-----------------------TTE------------------------
X is 1
Y: not determine
Z: not determine
W: not determine
Time for running TTE is 0.00027 seconds

-----------------------DPLL------------------------
X is 1
Y: not detemine
Z: not detemine
W: not detemine
Time for running TTE is 0.000333 seconds
```

Figure 8: Result of The Doors of Enlightenment



	MP	WW	H	LTT1	LTT2	MLTT	TDE1	TDE2
TTE	0.00011	0.000114	0.000281	0.000286	0.000277	0.00761	0.00049	0.0003
	0.000107	0.000111	0.000267	0.000267	0.000212	0.0075	0.000531	0.000285
	0.000089	0.000091	0.000221	0.000224	0.000185	0.006535	0.000457	0.000339
	0.000077	0.000085	0.000219	0.000209	0.000196	0.006633	0.000454	0.0003
	0.000094	0.000092	0.000231	0.000232	0.000191	0.007077	0.000444	0.000258
Ave_TTE	0.0000954	0.0000986	0.0002438	0.0002436	0.0002122	0.007071	0.0004752	0.0002964
DPLL	0.000089	0.000152	0.00036	0.000343	0.000328	0.00711	0.00066	0.00045
	0.000104	0.000142	0.000363	0.000285	0.000265	0.007116	0.000651	0.000429
	0.000079	0.000139	0.000314	0.000245	0.000229	0.006257	0.000594	0.00044
	0.000067	0.0001	0.000251	0.000235	0.000212	0.006247	0.000588	0.000327
	0.000073	0.000126	0.000293	0.000239	0.000224	0.006672	0.000584	0.000332
Ave_DPLL	0.0000824	0.0001318	0.0003162	0.0002694	0.0002516	0.0066804	0.0006154	0.0003956

Figure 9: Result of running five times for each algorithms

maybe the reason why *DPLL-Satisfiable* performs better in *More Liars and Truth-tellers* problem, which contained the largest amount of clauses.

5.2 Wumpus World

During the experiment, it is assumed that player does not know how many pits and wumpus are there in the world, but player would always begin in position $(0, 0)$ and knows that there is exactly one gold. Moreover, the algorithm is simplified to that player can go to the selected place directly, which does not effect the result for the reason that player would only choose adjacent places of visited position and can always find a safe path to reach there.

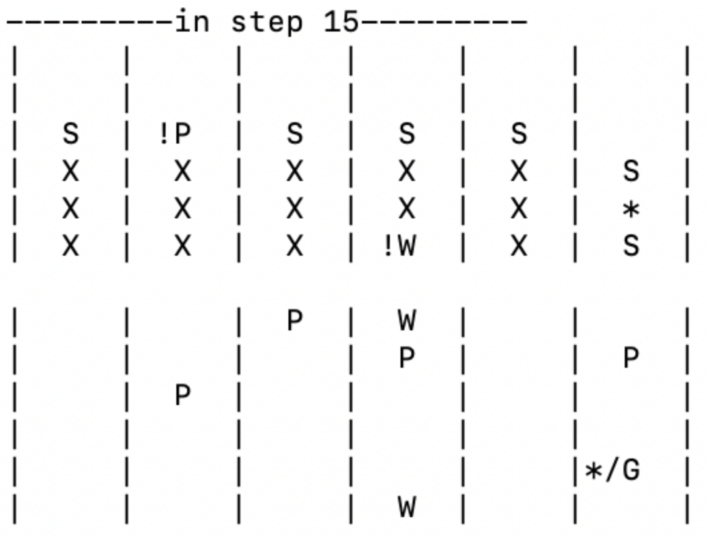
Implementing algorithm in section 3.7, the result is shown in figure 10. The X mark refers to the position the player has passed, S mark refers to the position proved to be safe, $!W$ mark refers to the position may have a wumpus, $!P$ mark refers to the position may have a pit. The player passed through wumpus and pits to get the gold.

Another result is shown in figure 11. The player has killed a wumpus when no safe positions can be selected. But such situation happens again, so player has to take a risk. But unfortunately, he losses.

Although the agent can win in most time during the test, it still did something stupid. For example, in figure 10, the mark $!M$ and $!P$ can be changed into M and P. But single mark without relations between may pit or may wumpus places can not realize such function. And in figure 11, the agent can actually go back to position $(1, 0)$ to prove that position $(2, 0)$ is safe. All of these problems can further be improved, which are introduced in section 6. (But I don't have time to handle this, sorry).

6 Future Work

Based on the result of experiments, several improvement can be made to current algorithm, which are listed below:
(1) For wumpus world algorithm, it can be improved by add relation mark for positions that may have wumpus or pits. The positions cause the same place to stench or bleeze are mark with a relation mark, which can update the



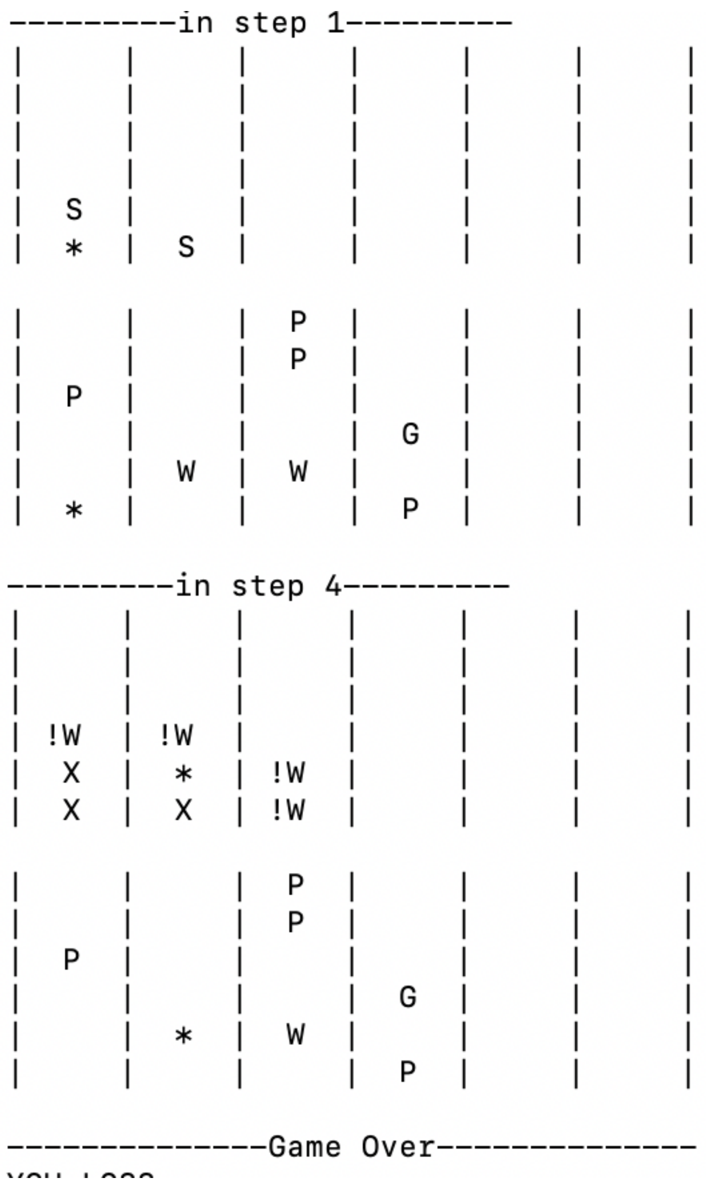
```
---------in step 15---------
|     |     |     |     |     |     |
|     |     |     |     |     |     |
|  S  | !P  |  S  |  S  |  S  |     |
|  X  |  X  |  X  |  X  |  X  |  S  |
|  X  |  X  |  X  |  X  |  X  |  *  |
|  X  |  X  |  X  | !W  |  X  |  S  |

|     |     |  P  |  W  |     |     |
|     |     |     |  P  |     |  P  |
|     |  P  |     |     |     |     |
|     |     |     |     |     |     |
|     |     |     |     |     |*/G  |
|     |     |     |  W  |     |     |
```

Figure 10: Winning the Wumpus World

"may pit" or "may wumpus" mark. Or the relation can be directly stored into knowledge base. Moreover, after the agent killed a wumpus, it should come back to the last position to detect whether there are more wumpus in adjacent positions, which can improve the probability for the agent to be survival. (2) For $DPLL - Satisfiable$ algorithm, if there exists several pure symbols or unit clause, they can be chosen in group. And parallel can be applied.

7 Conclusion

In the conclusion, the report shows two algorithm to prove entailment, which are *TT-Entail* and *DPLL-Satisfiable*. They are applied to six basic problems and their performance are compared. One can see that the comparison depends on the construction and the amount of clauses. Moreover, they are also utilized to play wumpus world, although more improvement can be made further.



```
---------in step 1---------
|     |     |     |     |     |     |
|     |     |     |     |     |     |
|     |     |     |     |     |     |
|     |     |     |     |     |     |
|  S  |     |     |     |     |     |
|  *  |  S  |     |     |     |     |

|     |     |  P  |     |     |     |
|     |     |  P  |     |     |     |
|  P  |     |     |     |     |     |
|     |     |     |  G  |     |     |
|     |  W  |  W  |     |     |     |
|  *  |     |     |  P  |     |     |

---------in step 4---------
|     |     |     |     |     |     |
|     |     |     |     |     |     |
|     |     |     |     |     |     |
| !W  | !W  |     |     |     |     |
|  X  |  *  | !W  |     |     |     |
|  X  |  X  | !W  |     |     |     |

|     |     |  P  |     |     |     |
|     |     |  P  |     |     |     |
|  P  |     |     |     |     |     |
|     |     |     |  G  |     |     |
|     |  *  |  W  |     |     |     |
|     |     |     |  P  |     |     |

--------------Game Over--------------
YOU LOSS
```

Figure 11: Loss the Wumpus World