Artificial Intelligence (CS F342)
Programming Assignment-2

Submitted to: Dr. Sujith Thomas

Submitted by: Sarthak Ajmera (2018A7PS0236G)

Approach

A literal is represented as an array of size 4. 1st element of which tells if the literal has a not symbol or not (-1 if not is present else 1.

A clause is represented as an array of literals, hence a clause data structure is array of arrays.

Knowledge base is essentially a set or array of clauses.

Knowledge base is initialised with background knowledge like there is exactly one pit and one wumpus, if a cell has wumpus then all its adjacent cells must have stench etc. All this background knowledge is converted into clauses and fed into knowledge base initially.

At every cell visit, we gain knew knowledge with the help of "percept", we add this knowledge to our knowledge base. Thus every cell visit updates our knowledge base.

<u>How to check if a cell is safe to visit?</u>

I did this checking in two simple steps. Let's assume we have to check if cell x,y is safe to visit.

1) Assume that cell x,y has a pit, check if this information is consistent with our knowledge base or not, to do so, add the sentence that x,y has a pit to our knowledge base and check if knowledge base is satisfiable or not using DPLL Algorithm. If it is not satisfiable, then the cell x,y doesn't have a pit.

2) Assume that cell x,y has a wumpus, check if this information is consistent with our knowledge base or not, to do so, add the sentence that x,y has a wumpus to our knowledge base and check if knowledge base is satisfiable or not using DPLL Algorithm. If it is not satisfiable, then the cell x,y doesn't have a wumpus.

DPLL Analysis:

The main idea behind DPLL Algorithms is assigning literals values true and false at each step and checking if solution exists for corresponding assignment or not. The worst case complexity for DPLL algorithm is exponential in terms of number of variables. Since we have 64 variables, this worst case is too much. 3 heuristics were used to improve run time of DPLL algorithms

1) Pure symbol heuristic

2) Unit Clause heuristic (Unit clause : clause having exactly one literal)

3) Early Termination heuristic

As per the analysis according to the following table, it can be concluded that unit clause heuristic and early termination heuristic significantly improve the running time of DPLL algorithm, while pure symbol heuristic has little impact on runtime;

POSITIONS	**TIME ANALYSIS**	**NUMBER OF DPLL CALL**
Position of pit : 2,3 Position of wumpus : 3,2	Without pure symbol heuristic: 613ms Without early termination heuristic: STUCK Without unit clause heuristic: STUCK With all the heuristic: 652ms	Without pure symbol heuristic: 360 Without early termination heuristic: STUCK Without unit clause heuristic: STUCK With all the heuristic: 306
Position of pit : 1,4 Position of wumpus : 4,1	Without pure symbol heuristic: 589ms Without early termination heuristic: STUCK Without unit clause heuristic: STUCK With all the heuristic: 623ms	Without pure symbol heuristic: 407 Without early termination heuristic: STUCK Without unit clause heuristic: STUCK With all the heuristic: 355