Assignment A4: Logical Agent using Resolution Theorem Proving

CS 4300
Fall 2017

Assigned: 12 September 2017

Due: 26 October 2017

For this problem, handin a lab report pdf (include name, date, assignment and class number in pdf) which develops and studies the use of logic in a hybrid Wumpus agent. Also, handin all functions used in this study.

Implement the Hybrid-Wumpus-Agent using the resolution theorem prover in A3. Here you should focus on:

- getting the hybrid-agent skeleton to work (e.g., combining the use of the KB with A*.
- develop and demonstrate a *MAKE-PERCEPT-SENTENCE* function.
- develop and demonstrate a *TELL* function.
- develop and demonstrate a *ASK* function.

Develop an appropriate KB to demonstrate the functions above. Develop a Hybrid-Wumpus-Agent function as described in Figure 7.20 (p. 270 in the text) which incorporates the *Tell* and *Ask* functions using propositional logic and resolution theorem proving. The exploitation of the knowledge can be with simple rules, but should use A* path planning where appropriate (e.g, to go back to start and exit). Use the following three Wumpus boards to demonstrate the behavior of your agent (e.g., step through state transitions and KB as actions are selected and executed – see Figure 1); Red is a pit; blue is the Wumpus, Yellow is the Gold; and the agent is facing toward the end with the square. You can use the same reduced KB for these boards. For data, describe the evolution of the agent's knowledge of the world over the time steps it runs, give its actions and its score on each of the three boards.

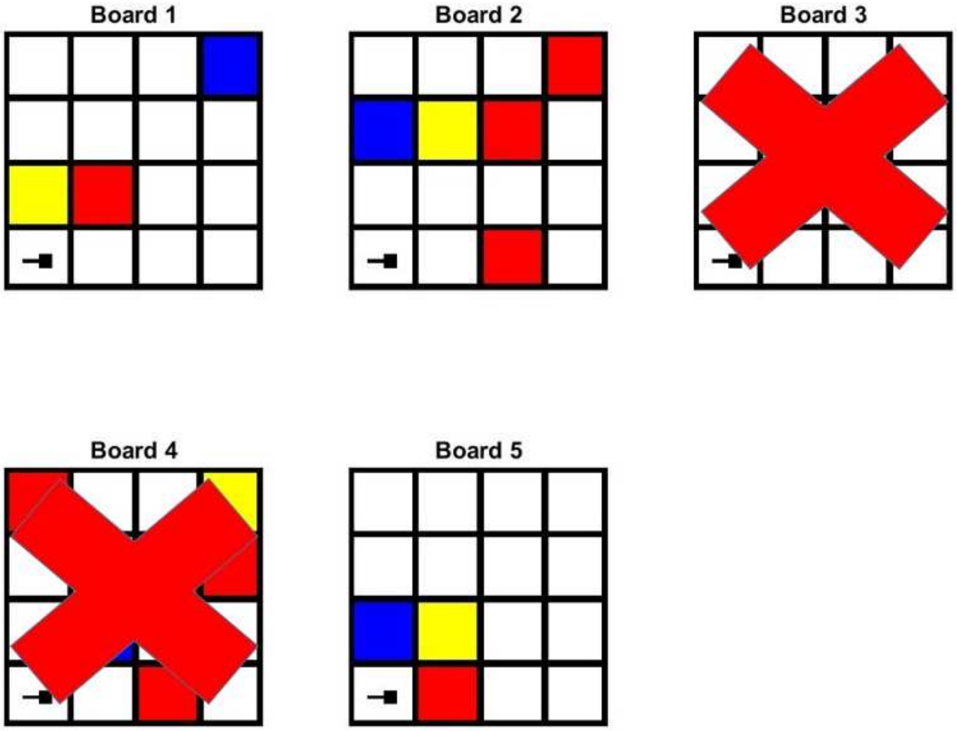

Figure 1: Three Boards to Test.

Discuss the advantages and disadvantages of using the (propositional) logical approach and document based on your results.

For each, you should handin the report pdf as well as the source code developed in the study. The code should conform to the style requested in the class materials.

In addition, please turn in a hardcopy of the report in class before the start of class on October 26, 2017.

Write a lab report in the format (please do not deviate from this format!) described in the course materials.

```
function action = CS4300_hybrid_agent(percept)
% CS4300_hybrid_agent - hybrid random and logic-based agent
% On input:
%     percept( 1x5 Boolean vector): percepts
% On output:
%     action (int): action selected by agent
% Call:
%     a = CS4300_hybrid_agent([0,0,0,0,0]);
% Author:
%     <Your name>
%     UU
%     Fall 2017
%


function b = CS4300_Ask(KB,sentence)
% CS4300_Ask - Ask function for logic KB
% On input:
%     KB (KB struct): Knowledge base (CNF)
%       (k).clauses (1xp vector): disjunction clause
%     sentence (KB struct): query theorem (CNF)
%       (k).clauses (1xq vector): disjunction
% On output:
%     b (Boolean): 1 if KB entails sentence, else 0
% Call:
%     KB(1).clauses = [1];
%     KB(2).clauses = [-1,2];
%     sentence(1).clauses = [2];
%     b = CS4300_Ask(KB,sentence);
% Author:
%     <Your name>
%     UU
%     Fall 2017
%


function KB_out = CS4300_Tell(KB,sentence)
% CS4300_Tell - Tell function for logic KB
% On input:
%     KB (KB struct): Knowledge base (CNF)
%       (k).clauses (1xp vector): disjunction clause
%     sentence (KB struct): query theorem (CNF)
%       (k).clauses (1xq vector): disjunction
% On output:
%     KB_out (KB struct): revised KB
% Call:
%     KB = CS4300_Tell([],[1]);
% Author:
%     <Your name>
%     UU
%     Fall 2017
%


function sentence = CS4300_make_percept_sentence(percept,x,y)
% CS4300_make_percept_sentence - create logical sentence from percept
% On input:
%     percept (1x5 Boolean vector): percept
%       [Stench,Breeze,Glitter,Scream,Bump]
%     x (int): x location of agent
%     y (int): y location of agent
% On output:
%     sentence (KB struct): logical sentence (CNF)
%       (1).clauses (int): c1 (index of Sxy if stench), else -c1
%       (2).clauses (int): c2 (index of Bxy if breeze), else -c2
%       (3).clauses (int): c3 (index of Gxy if glitter), else -c3
%       (4).clauses (int): c4 (index of Cxy if scream), else -c4
%       (5).clauses (int): c5 (index of Exy if bump), else -c5
% Call:
%     s = CS4300_make_percept_sentence([0,1,0,0,0],3,2);
% Author:
%     <Your name>
%     UU
%     Fall 2017
%
```