Computer lab assignment 4

Reasoning Agents

Prof. Michael Thielscher, Sebasian Voigt

International Masters Programme in Computational Logic — summer term 2009

26.06.2009

Assignment 4.1

This lab is concerned with the *Wumpus World* (cf. lecture 4b). Download the files `4_wumpus.pl`, `4_wumpus_simulator.pl`, `4_fluxALP.pl` and `4_fluent.chr` from the course web page and solve the following tasks.

a) Compile the file `4_wumpus.pl` (it will subsequently compile all the other files). The scenario in Figure 1 is predefined. Run the code via the query `?- run(main).` and follow the output to see that the agent actually finds an action sequence to bring the gold to $(1,1)$.

b) Change the considered scenario to the predefined one shown in Figure 2. Activate it by uncommenting the fact `scenario(hunt)` in file `4_wumpus_simulator` (and comment the fact `scenario(test)`). Run the program to verify that the agent is not able to catch the gold because the Wumpus is blocking it.

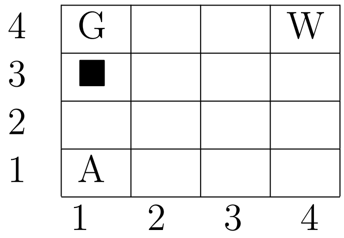
The agent is able to enter the field of the Wumpus safely if it is dead. Your task is to enrich the behaviour of the agent such that it shoots the wumpus eventually if it gets to know where the wumpus is. Do this by extending the predicate `try_wumpus_hunt` (predefined in `4_wumpus.pl`) which is called in the main loop. Test your implementation in the hunt scenario!

c) On its way home the agent follows the track it has come. This is not always efficient. Implement an A^* search such that the agent takes the shortest safe way back home.

Hints:

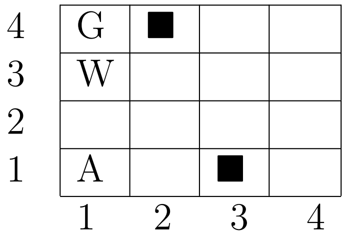
- Define a predicate `expand(Node,SuccNodes)` which for an A^* node `Node` gives all the possible successor nodes in a list `SuccNodes`. A node should contain a list of positions that have been followed during search so far and its A^* heuristic value.
- Define a predicate `sorted_insert(SuccNodes,Nodes,NewNodes)` which inserts `SuccNodes` into a sorted list `Nodes` such that the elements in `NewNodes` are again sorted. Sort the list in increasing order with respect to the heuristic values of the nodes.
- Define a predicate `a_star_plan(BtrSeq)` such that after calling the variable `BtrSeq` is assigned to a list of directions that can be followed backwards to get home. It thus can be used instead of the remembered backtrack sequence after grabbing the gold.

d) Test your implementation using `scenario(random)` which generates a new scenario every time you run the agent. You may also vary the number of generated pits and the dimension of the cave in the file `4_wumpus_simulator.pl`.



4	G			W
3	■			
2				
1	A			
	1	2	3	4

Figure 1: A scenario of the *Wumpus World*, where G represents gold, W the wumpus, A the agent and ■ a pit.



4	G	■		
3	W			
2				
1	A		■	
	1	2	3	4

Figure 2: Another scenarios of the *Wumpus World*, where the gold can only be reached if the wumpus is dead.