Part II Software Systems

C. AI

1. Search: (25 points)

Having passed your prelims and finished your PhD, you find yourself teaching an introductory AI course. For your first homework assignment, you need to create a graph to search to test your student's knowledge of Depth First Search, Breadth First Search, and A* Search.

- So that it will be easy to grade, you need a graph that will be searched DIFFERENTLY under each of the three methods.

- To make sure your students understand where Depth- and Breadth-First searches fail, you want these answers to be non-optimal.

- One problem is, to make sure that all students will expand new nodes in the same order. You will label each node with a letter, and tell them that the nodes will be GENERATED in alphabetical order. This means that if a node P has children X and Y, then when P is expanded, it will find X and then Y. If you were to push them onto a stack, you would push X, then push Y (i.e. Y would POP first), etc.

A) Create such a search graph that will be searched differently by EACH of the three search algorithms, and will be searched non-optimally by depth- and breadth-first search. Label each node with a letter and an admissible h value. Indicate which nodes correspond to goal states (you can have more than one).

B) Create an answer for your problem. Clearly show in what order the nodes are expanded and what the open list looks like for each of the three algorithms. Briefly explain to the students why the depth- and breadth-first algorithms fail to be optimal in this case.

2. Logic, Representation, Planning: (25 points)

Situation Calculus is the name for a particular way of describing change in first order logic. It conceives of the world as a series of *situations*, each of which is a snapshot of the current state. A new situation is generated from the previous situation by an *action*. Every logical relation or property that can change over time, as the result of an action, is called a *fluent*, and is given an extra situation argument.

For example, we could represent that Robbie the Robot is in room 102 in situation S0 by writing the sentence:

At(ROBBIE, 102, S0)

The situation that results from taking an action a in previous situation s is written (note that this is a function with an object value, not a logical sentence with a true/false value):

Result(a,s)

Thus if Robbie goes from room 102 to room 203 by the action Go(203), then we can write the new location of Robbie as the sentence:

At(ROBBIE, 203, Result(Go(203), S0))

To use the situation calculus to represent a changing world, we need to write axioms that

- describe the effect of actions on fluents
- deal with the frame problem with respect to actions and fluents.

For each fluent, then, we can write a *successor-state axiom* that says:

the fluent is true in the resulting situation if and only if

- the last action made it true
- OR it was true already and the last action didn't make it false.

For example, for the At fluent in Robbie's Robot World (all lowercase variables are universally quantified):

At(r,l,Result(a,s)) <=>
[a=Go(l) ^ At(Robbie,m,s) ^ Pathfrom(m,l)] OR
[At(Robbie,l,s) ^ a/=Go(x) ^ x/=l]

Question: ("The Yale Shooting Problem") Consider a world with a gun permanently pointed at a live,immobile wumpus. There are two actions that your agent can take: loading a gun and shooting. There are two interesting predicates, whether the wumpus is alive and whether the gun is loaded. Write a successor state axiom for each of the two fluents.

3. Resolution Theorem Proving: (25 points)

A. Write a predicate calculus statement for the following:

(3 pts) All hotels in New York City that offer a Penthouse suite have a janitor who is older than any waiter at the hotel.

B. Suppose that you are given the following premises:

$(\forall z)(\forall w)(\forall x)(\exists y)\ [P(A,x,y) \vee Q(y,w) \vee L(x,y,z,w)]$

$(\forall w)(\exists x)(\forall z)\ [\sim P(w,x,z) \wedge \sim R(z,B)]$

$(\exists x)(\forall z)(\exists y)(\forall w)(\forall v)\ [L(w,v,y,x) \Rightarrow R(y,z)]$

(9 pts) 1. Convert the above premises to clause form

(13 pts) 2. Use resolution theorem proving to answer the following question:

"For what values of the variables s and t is Q(s,t) true?"

Be sure to show all details of your work, including the substitutions made at each step and how you eventually determine the values of s,t.

4. Pruning: (25 pts.)

(8 pts.) A. Give an example of a portion of a game tree that illustrates an alpha cutoff and another that illustrates a beta cutoff. (Be sure to show where your alpha and beta cutoffs are on your tree.) Explain why it is pointless to search the portion of the tree that is to be cut off.

(17 pts.) B. In standard minimax search there is an assuption that the rules of each game define a utility function that is used by both players, and that a utility of x for player1 means a utility of -x for player2. Games with this property are called zero-sum games. Describe how the minimax and alpha-beta algorithms change when we have non-zero-sum games--this is, when each player has his/her own utility function. You may assume that each player knows the other's utility function.

(Hint: Consider that the utility function at each node might be represented as a vector of values--the first representing the value for player1 and the second representing the value for player2. Because the game is non-zero-sum, both values in the vector associated with a node may be positive.)

As part of your answer to this question, consider the following game tree and describe how the values should be backed up. The vectors representing the utility function at the leaves are shown.