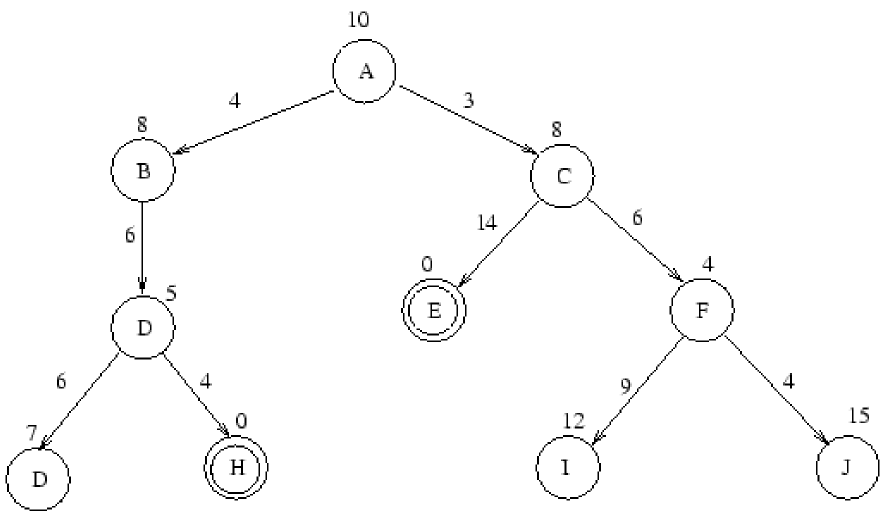
COMP 131 – Artificial Intelligence – Midterm 1 Review

Monday, Oct 6, 2008

These problems have been taken from previous COMP 131 exams (credit to Jim Schmolze), as well as from MIT's OpenCourseWare materials.

1. Search

The figure below shows a small search space. Each node in the space is labeled with a letter. The nodes represented by double circles are the only goal nodes. Above each node is the heuristic estimate for that node. Each link represents an operator, and is labeled with the cost of applying that operator.



For each type of search listed below, show the order in which the nodes are examined (i.e., pulled off of the open list). If there is ambiguity about which node goes next, pick the node that is higher in the tree, followed by the node that is leftmost in the tree.

1. greedy

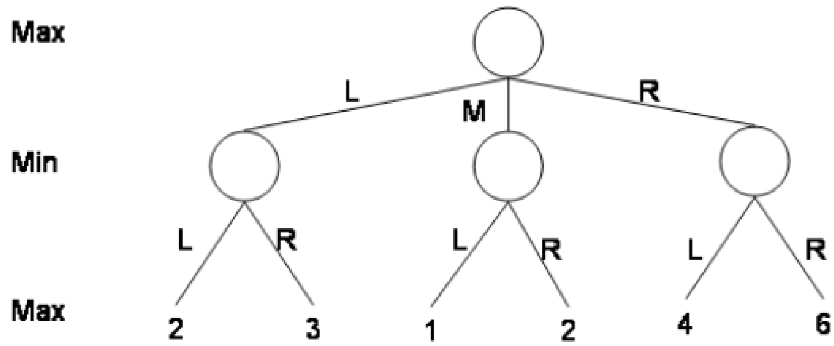
2. uniform cost

3. A*

Is this heuristic admissible? Is it consistent? Explain why.

2. Games

Consider the game tree shown below. The top node is a max node. The labels on the arcs are the moves. The numbers in the bottom layer are the values of the different outcomes of the game to the max player.



1. What is the value of the game to the max player?

2. What first move should the max player make? Assuming max plays that move, what move should min play afterwards? Assume this is the entire game tree.

3. Using alpha-beta pruning on nodes **from right to left**, which nodes are cut off and not examined? Circle the nodes that are not examined.

3. CSP

Let's look at the problem of scheduling programs on a set of computers as a constraint satisfaction problem. We have a set of programs (jobs) J_i to schedule on a set of computers (machines) M_j . Each job has a maximum running time R_i. We will assume that jobs (on any machines) can only be started at some pre-specified times T_k . Also, there's a T_{max} time by which all the jobs must be finished running; that is, start time + running time is less than or equal to max time. For now, we assume that any machine can execute any job. Let's assume that we attack the problem by using the jobs as variables and using values that are each a pair (M_j, T_k). Here is a simple example.

- Running time of J_1 is $R_1 = 2$
- Running time of J_2 is $R_2 = 4$
- Running time of J_3 is $R_3 = 3$

- Running time of J_4 is $R_4 = 3$
- Starting times $T_k = \{1, 2, 3, 4, 5\}$
- There are two available machines: M_1 and M_2
- The max time is $T_{max} = 7$
- An assignment would look like $J_1 = (M_2, 2)$, which means: run job J_1 on machine M_2 at time 2

1. What are the constraints for this type of CSP problem? Write a boolean expression using logical connectives and arithmetic operators, that must be satisfied by the assignments to each pair of variables. In particular: J_i with value (M_j, T_k) and J_m with value (M_n, T_p)

2. Write down a complete valid solution to the example problem above.

3. Which variable would be chosen first if we did backtracking with forward checking with dynamic ordering of variables using the most constrained heuristic? Why?

4. If we do constraint propagation in the initial state of the example problem, which domain values (if any) are eliminted? Explain.

5. If we set $J_2 = (M_1, 1)$, what domain values are still legal after forward checking?

6. We could have formulated this problem using the machiens M_j as the variables. What would be the variable values for this formulation, assuming you have N machines and K jobs to schedule?

7. What are some disadvantages of this alternative formulation?

4. Logic

Use first-order resolution theorem proving.

1. Show how each sentence is translated into a set of clauses.
2. Write the steps that prove sentence 4 from sentences 1, 2 and 3. If you use unification at any point, show the substitution σ that resulted.

Given:

1. $\forall y.((\neg\exists x.On(x,y)) \rightarrow Clear(y))$
2. $\forall x,y.On(x,y) \rightarrow Above(x,y)$
3. $\forall x.\neg Above(x,B)$

Prove:

4.$Clear(B)$