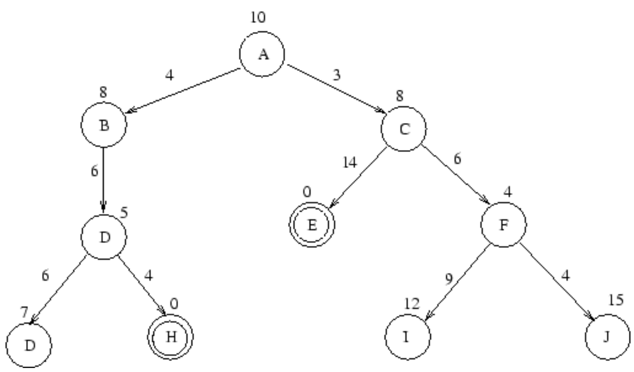
COMP 131 – Artificial Intelligence – Midterm 1 Review – With Answers

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

Answer: A B D H. Gredy (best-first) search uses an evaluation function that is equal to the heuristic function, and picks the next best node from the fringe (open list) according to the evaluation function.

2. uniform cost

Answer: A C B F D J H. Uniform cost search uses the cost of path so far (from the start), and picks the smallest-cost node next.

3. A*

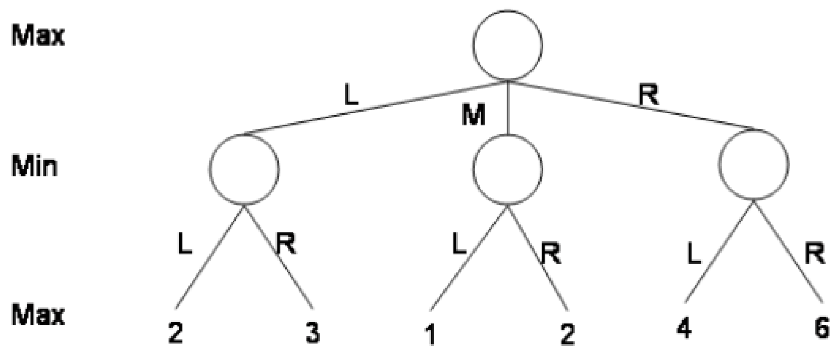
Answer: A C B F D H. A* uses eval(node) = cost-so-far(node) + h(node).

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

Answer: The heuristic is not admissible. To be admissible, it must always be less than or equal to the true cost to goal. In the graph, e.g., node h(D) = 5, but the real cost from D to the goal H is only 4. To be consistent, it must be the case that $h(n) \leq cost(n, n') + h(n')$. The same example (n = D, n' = H) shows that this heuristic is not consistent, because $h(D) > cost(D, H) + h(H)$.

2. Games

Consider teh 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?

   Answer: 4

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.

   Answer: Max should play R. Min then should play L.

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.

   Answer: The nodes not examined are the left-most node labeled "2" and the node labeled "1".

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)

   Answer: There is a unary constraint on legeal values for a single variable: $T_k + R_i \leq T_{max}$. This is not a binary constraint on pairs of values. The binary constraint is the one that says that jobs on the same machines must not be double-booked (overlap in time). It can be expressed as:
   $(M_j = M_n) \rightarrow (T_k + R_i \leq T_p \vee T_p + R_m \leq T_k)$
   So, either the machines are different, or the times don't overlap.

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

   Answer: Several legal assignments exist. Here's one:

   - $J_1 = (M_1, 1)$
   - $J_2 = (M_1, 3)$
   - $J_3 = (M_2, 1)$
   - $J_4 = (M_2, 4)$

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?

   Answer: J_2 would be chosen since it has the smallest domain of legal values. That job since it takes 4 time steps, can only be started at times less than or equal to 3 so that it will finish before $T_{max} = 7$.

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

   Answer: Assume that the domain values inconsistent with the unary constraint (T_{max}) have been eliminated from the domains before constraint propagation started. Then no further domain values are eliminated. We can always run a pair of jobs on different machines and so the binary constraints do not reduce the domains further. If you assume that the unary constraints are checked during propagation, then indeed those values will be eliminated. But it's better practice to restrict the **initial domains** to values satisfying unary constraints only, before any other search.

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

   Answer: Forward checking works like this: whenever a variable x is assigned a value, it looks at all other variables y connected to x by a constraint, and removes values inconsistent with x from y's domains.

- $J_1 \in (M_1, 5), (M_2, t), t \in \{1, 2, 3, 4, 5\}$
- $J_2 \in (M_1, 1)$
- $J_3 \in (M_2, t), t \in \{1, 2, 3, 4\}$
- $J_4 \in (M_2, t), t \in \{1, 2, 3, 4\}$

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?

   Answer: A value would be a complete schedule for each machine, that is, a list of all the jobs to run on the machine. One could also specify the starting times of each job, but that is redundant, since the running time could be used.

7. What are some disadvantages of this alternative formulation?

   Answer: There would be a very large number of possible values in the domain of each variable (every way of splitting K jobs among M machines so that the sum of the running times is less than T_{max}.

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.$Clear(B)$

Answer:

Clausal form:
1. $On(f(y), y) \vee Clear(y)$ – skolemization of an existentially quantified variable (x), that has a universally quantified variable (y) in its outer scope
2. $\neg On(x, y) \vee Above(x, y)$
3. $\neg Above(x, B)$
4. $\neg Clear(B)$ – negated conclusion for refutation by resolution.

Resolution proof (there are several ways to do it):
5. (1 and 4) $On(f(B), B), \sigma = \{y \leftarrow B\}$
6. (2 and 5) $Above(f(B), B), \sigma = \{x \leftarrow f(B), y \leftarrow B\}$
7. (3 and 6) empty clause, $\sigma = \{x \leftarrow f(B)\}$

Therefore, 4 is entailed by (1-3).