CS 2710/ISSP 2160 Fall 2015

Problem-Solving Search: Self Exercises

Solutions will be posted after you have had a chance to answer the questions yourselves. Note that these are more difficult questions than will be on the exam.

Q1 Suppose we were to change treesearch (or graphsearch) as follows: it applies the goal test to a node when the node is first generated, before the node is added to the fringe. Call this treesearchV2 (graphsearchV2).

- For breadth-first-search using treesearchV2, this change does not affect its completeness and optimality. And, this change can save time and space. In the worst case, when the goal is on the right frontier of the search tree, my version of breadth-first search generates, and adds to the fringe, an extra level of nodes than breadth-first-search using treesearchV2 does. In figure 3.21 (time and space), breadth-first search is $O(b^d)$, and uniform-cost search is $O(b^{(d+1)})$ if all edge-costs are equal. Figure 3.21 assumes that treesearchV2 is used for breadth-first-search.

  Note: the wrap-up in the chapter3part1 slides points out that the book writes separate code for breadth-first search that makes the above change (in addition, their separate version performs graphsearch rather than treesearch).

  Argue that breadth-first-search using treesearchV2 maintains the completeness and optimality properties of breadth-first search.

  * Completeness Answer: The change from the treesearch to the treesearchV2 version of breadth-first-search cannot affect completeness: the change is that the goal test is applied to nodes earlier than before. Until it finds a goal, the treesearchV2 version does not fail to apply the goal test to any node that the treesearch version applies it to. Therefore, the change cannot make the treesearchV2 version miss a goal that the treesearch version finds.
  * Optimality Answer: For this property to be lost, the treesearchV2 version would need to return a suboptimal goal. To do this, it would need to apply the goal test to a suboptimal goal before it applies the test to an optimal goal (both versions return whenever the goal test succeeds). However, the treesearchV2 version applies the goal test earlier than the treesearch version does. Breath-first-search is optimal because it expands (and thus goal-tests) all the nodes on level l before it expands any nodes on level p, p > l. Therefore, changing the algorithm so that it goal-tests *sooner* cannot make it lose its optimality property.

Q1.b However, for some of the other search algorithms we are covering, optimality is lost if treesearchV2 is used. Give an example. Specifically, first choose one of the search algorithms. Then, give the state space, and show a trace of the algorithm finding a suboptimal goal (specifically, show the fringe during each iteration). Your answer should clearly state why the goal is suboptimal. Give a **small** example.

Answer: Uniform-cost-search. Simple state-space: successors(start) are A and B (generated in that order). Both are goals. The edge costs are start to A: 10; start to B: 5. Thus, B is optimal and A is suboptimal. The treesearchV2 version would return suboptimal goal A, since it tests nodes when they are generated and A is generated first. However, the treesearch version first places the nodes on the fringe, ordered by path cost, and then tests the node that is ordered first on the fringe. Thus, it finds and returns optimal goal B.

Q2 In class, we said that breadth-first search is optimal if the edgecosts are all equal. On page 82 in R&N just below figure 3.11, we learn that this is too strict – it is optimal if the path cost is a nondecreasing function of the depth of the node. Explain why this is so.

Background notes: If g(n) is a nondecreasing function of the depth of the node, then, by the definition of nondecreasing functions, $g(b) \geq g(a)$ for all $b \geq a$. Breadth-first search applies the goal test to all nodes on level l before applying it to any node on level p, $p > l$.

Breadth-first search is optimal if the path cost is a nondecreasing function of the depth of the node.

Proof: Assume by way of contradiction that breadth-first search returns a sub-optimal goal, g2. Let d2 be the depth of g2. Since breadth-first search returns goal g2, the search did not earlier find an optimal goal (or the search would have already returned it and stopped). Thus, all optimal goals are at depths greater than or equal to d2. Suppose g0 is an optimal goal and is at depth d0. $d0 \geq d2$, so $g(d0) \geq g(d2)$, since g is non-decreasing. But this is a contradiction with our assumption: if $g(d0) \geq g(d2)$, then g2 is not a suboptimal goal.