Techniques of Artificial Intelligence

Exercises – Search Space

Dipankar Sengupta
Dipankar.Sengupta@vub.ac.be

Roxana Rădulescu
Roxana.Radulescu@vub.ac.be

February 15, 2016

1. **Fundamentals for search**
   Suppose you have a data structure D with the push and pop operation. Suppose you have the following sequence of push operations: push(1), push(3), push(2).

   (a) Which elements are returned by two consecutive pop operations if D is a stack, if D is a queue?

   (b) Now suppose that D is a priority queue and that the priority of each element is the value of the element itself. Which elements are returned by two consecutive pop operations now?

   Answer:

   (a) If D is a stack, the first pop operation returns 2, the second one returns 3. If D is a queue, the first pop operation returns 1, the second one returns 3.

   (b) If D is a priority queue, the first pop operation returns 3, the second one returns 2.

2. **General search algorithms**

   add initial state to agenda;
   while *agenda not empty and solution not found* **do**
       remove the first node N from the agenda;
       if *N is the goal* **then**
           stop searching and return the solution;
       else
           find all successors of N and add them to the agenda;
       end
   end

   The outline of a general tree search algorithm is sketched above. How can this outline be turned into:

   (a) depth first search

   (b) breadth first search

   (c) best first search

   (d) A* search

   For each of those search algorithms, start from the general algorithm above and add more details to it, such that it becomes the desired search algorithm.

   Answer:

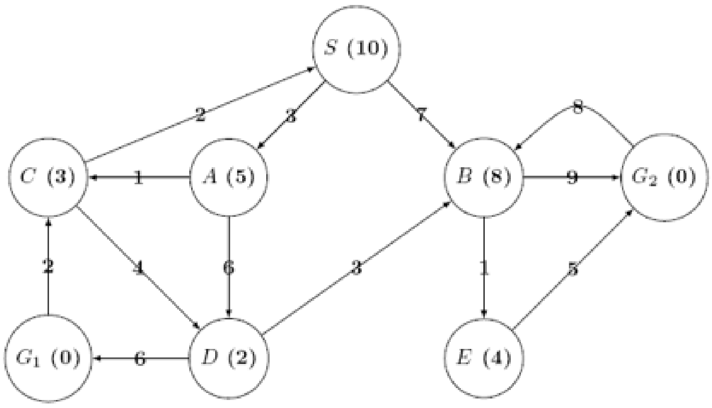
   The working of the agenda is underspecified in the above sketch:

   (a) for depth first search, the agenda should operate as a stack: new nodes are added to the front of the agenda and taken from the front.

   (b) for breadth first search, the agenda should operate as a queue

(c) for best first search, the agenda should be a priority queue with priorities f(n) = h(n)
(d) for A*, the agenda should be a priority queue with priorities f(n) = g(n) + h(n)

3. **Search spaces**
Consider the state space illustrated in the figure below.
   (a) S is the initial state
   (b) G1 and G2 are goal states
   (c) Arcs show actions between states (e.g., the successor function for state S returns {A, B}).
   (d) Arcs are labelled with actual cost of the action (e.g., the action from S to A has a cost of 3).
   (e) The numeric value in parentheses in each state is the states h-value (e.g.,h(A) = 5).



Give the sequence of states that will be visited, together with the total cost of reaching the goal state, for (1) depth first search, (2) breadth first search and (3) A*. You should assume the following on the operational details of the algorithms:
   (a) The algorithm does not check if a state is revisited, so there may be several nodes with the same state in the search tree.
   (b) The algorithm terminates only when it selects a goal node for expansion, not when it is generated by the successor function.
   (c) The successor function always orders states alphabetically.

Answer:

Depth first search, path found: S – A – C – D – B – E – G2 = 17

Breadth first search, path found: S – B – G2 = 16

A*, path found: S – A – C – D – G1 = 14

For A*, the queue contains the following items after each pass:

1 S = 10
2 A = 8, B = 15
3 C = 7, D = 11, B = 15
4 D = 10, B = 15, (S = 16)
5 G1 = 14, B = 15, (S = 16, B = 20)
6 G1

Note that A* does not find the path with the least cost (S – B – E – G2 with a cost of only 13). This is because the heuristic is not admissible: A* is guaranteed to find the shortest path only when the heuristic used is admissible. This means that the heuristic should never overestimate the true cost. In this assignment, the heuristic is not admissible. Example: In B the cost is estimated to be 8 while the true cost to reach G2 is only 6 (B – E – G2).