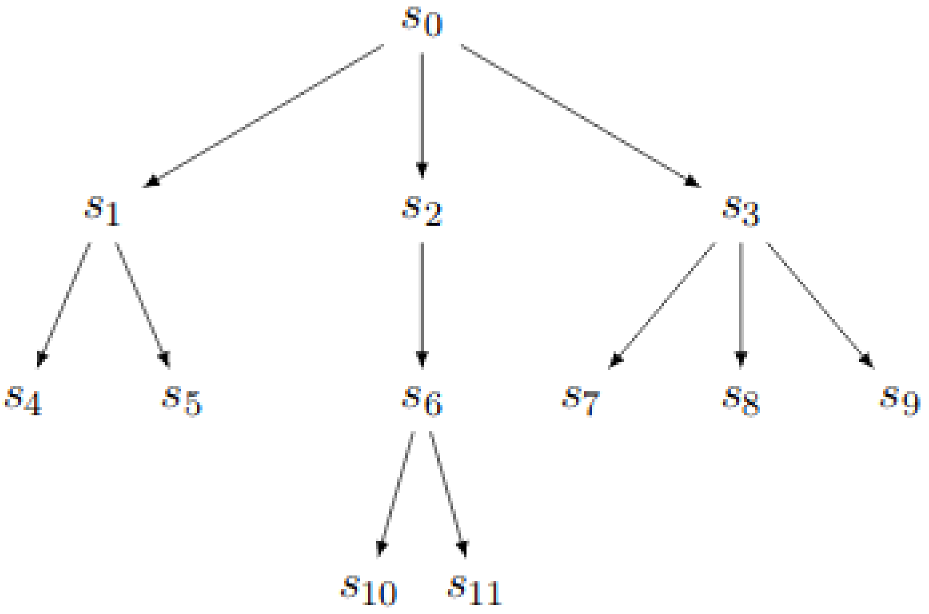
AI tutorial 2

Brandon Skerritt

December 30, 2017

1 Questions

Questions 1 - 5 refer to the following search graph with start state $s_(start) = s_0$



1.1 Question 1

Suppose A is a search algorithm applied to the search graph. If

$$s0s1s2s3s4s5s6s7s8s9s10s11$$

is the sequence in which the states are expanded by A, then A:

- Performs a depth-first search strategy
- Performs a breadth-first search strategy
- Does not perform either a depth-first or a breadth-first searchsrategy

So the search algorithm starts at the start state s0, then it reads s1, s2, s3 (because it expands s0). The frontier looks like {s1, s2, s3} because s0 is not the goal node and it has been expanded. The algorithm then moves down level by level, selecting each frontier using a first in first out queue. Which makes it a BFS. The full frontier looks something like this for BFS.
s0
S0 not goal, frontier = {s1, s2, s3}
S1 not goal, frontier = {s2, s3, s4, s5}
s2 not goal, frontier = {s3, s4, s5, s6}
s3 not goal, frontier = {s4, s5, s6, s7, s8, s9}
s4 not goal, frontier = {s5, s6, s7, s8, s9}

s5 not goal, frontier = {s6, s7, s8, s9}
s6 not goal, frontier = {s7, s8, s9, s10, s11}
s7 not goal, frontier = {s8, s9, s10, s11}
s8 not goal, frontier = {s9, s10, s11}
s9 not goal, frontier = {s10, s11}
s10 not goal, frontier = {s11}
s11 - goal presumably found BFS uses a queue, it is a first in first out.

1.2 Question 2

Assume A performs a depth-first strategy. Give a (possible) sequence in which the states are expanded by A.

DFS uses a stack, a last in first out style.
s0
s0 not goal, frontier = {s3, s2, s1}
s1 not goal, frontier = {s3, s2, s5, s4}
s4 not goal, frontier = {s3, s2, s5}
s5 not goal, frontier = {s3, s2}
s2 not goal, frontier = {s3, s6}
s6 not goal, frontier = {s3, s11, s10}
s10 not goal, frontier = {s3, s11}
s11 not goal, frontier = {s3}
s3 not goal, frontier = {s9, s8, s7}
s7 not goal, frontier = {s9, s8}
s8 not goal, frontier = {s9}
s9 presumably goal

DFS is backwards, presumably because of its LIFO queue. It always goes down the left most path first, therefore its frontier must be backwards. You must insert items into the frontier counting from the right, not the left like in BFS.

Or, DFS is backwards because it uses a stack. In a queue, items are added to the front and removed from the back. In a stack, items are added to the front and removed from the front. The very last item added is the first item removed.

1.3 Question 3

Suppose A is a search algorithm applied to the search graph. Before A terminates the frontier contains the following paths {s0s1, s0s2, s0s3} and s0s1 is the path selected the path selected to continue the search. Is it possible to decide whether A is a depth-first or a breadth-first search algorithm? Explain your answer.
Answer First thing to note is that the frontier contains the full path up to that point in the search graph, make sure to include this in all my other frontiers.

It is not possible because BFS would select s0s1 to expand since it is a first in first out queue (therefore s0s1 would be expanded) and DFS is always selects the left most path, which in this case is s0s1 to expand.

1.4 Question 4

Suppose A is a search algorithm applied to the search graph. Before A terminates the frontier contains the following paths {s0s2, s0s3, s0s1s4, s0s1s5} and {s0s1s4} is the path selected to continue the search. Is it possible to decide whether A is a depth-first or a breadth-first search algorithm? Explain your answer.
Yes, this is a DFS search because it chooses to explore depths instead of go across. If it was BFS, it would have all elements at the level of s0s1s4.

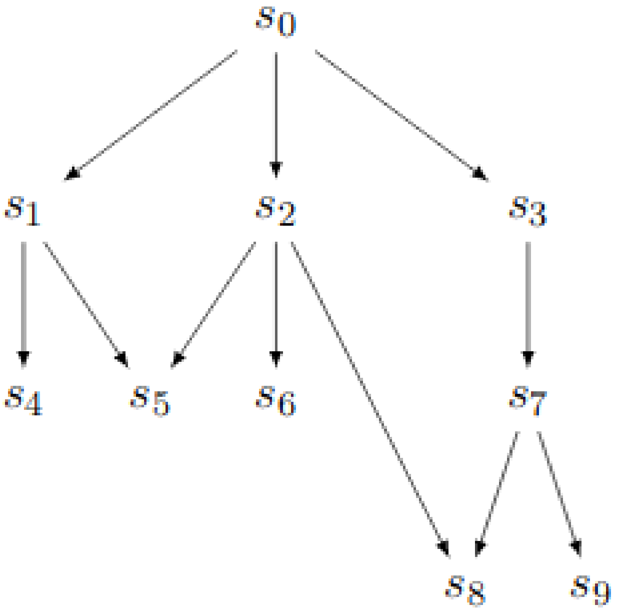
1.5 Question 5

Suppose A is a search algorithm applied to the search graph. Before A terminates the frontier contains the following paths $\{s0s2, s0s3, s0s1s4, s0s1s5\}$ and s0s2 is the path selected to continue the search. Is it possible to decide whether A is a depth-first or a breadth-first search algorithm?

Explain your answer. **Answer** Yes. We know that s0s1 has been removed from the frontier, adding s0s1s4 and s0s1s5 to the frontier. Because it chooses to explore s0s2 it is BFS. There are 2 ways we know this, one, it is using FIFO queue. Two, it is all working on the same level.

2 Questions

Questions 6–9 refer to the following search graph with $s_(start) = s0$.



2.1 Question 6

Suppose the goal is to find a path to s8. What is the difference between applying a depth first search strategy or a breadth-first search strategy?

Frontier for BFS

s0
s0 not goal, frontier = {s1, s2, s3}
s1 not goal, frontier = {s2, s3, s4, s5}
s2 not goal, frontier = {s3, s4, s5, s6, s8}
s3 not goal, frontier = {s4, s5, s6, s8, s7}
s4 not goal, frontier = {s5, s6, s8, s7}
s5 not goal, frontier = {s6, s8, s7}
s6 not goal, frontier = {s8, s7}
s8 is goal, frontier = {s7}
BFS would return s0s2s8 because it first detected that the goal node was in sight at s0s2 when it expanded s2. it does not explore anymore or explore the goal node.

Frontier for DFS

s0
s0 not goal, frontier = {s3, s2, s1}
s1 not goal, frontier = {s3, s2, s5, s4}
s4 not goal, frontier = {s3, s2, s5}
s5 not goal, frontier = {s3, s2}
s2 not goal, frontier = {s3, s6, s8}
goal reached, returning s0s2s8

3 Question 7

Suppose the goal is to find a path to s8. Is it the case that a depth first strategy is guaranteed to return the shortest path? Explain your answer.

4 Question 8

Suppose the goal is to find a path to s8. Is it the case that a breadth first strategy is guaranteed to return the shortest path? Explain your answer.
Yes, BFS always returns the shortest path.

5 Question 9

Suppose the goal is to find a path to s5. Is there a difference between applying a depth-first search strategy or a breadth-first search strategy? Explain your answer

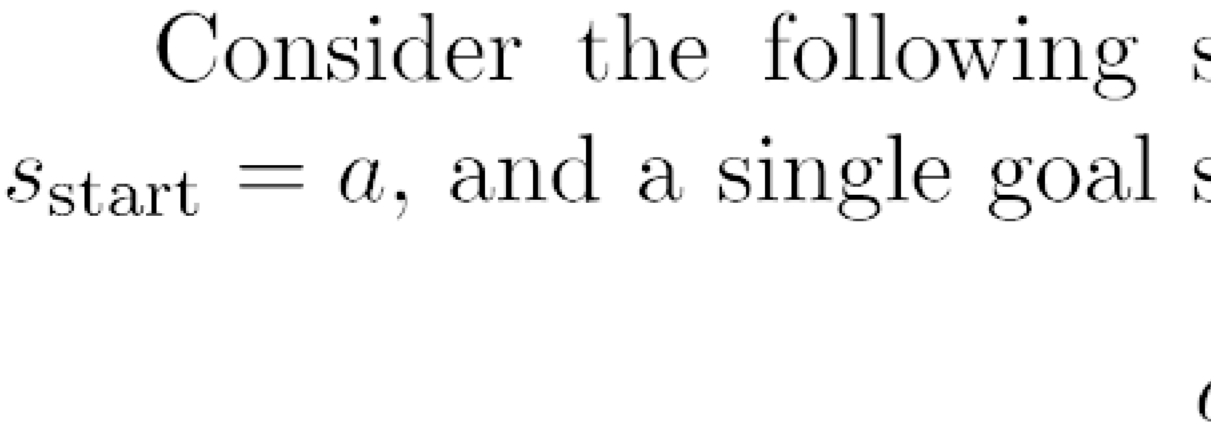
5.1 Frontier for BFS

S0 s0 not goal, frontier = {s1, s2, s3} s1 not goal, frontier = {s2, s3, s4, s5} goal found. goal is s0s1s5.

5.2 Frontier for DFS

S0 s0 not goal, frontier = {s3, s2, s1} s1 not goal, frontier = {s3, s2, s5, s4} Goal found, terminating. S0s1s5

6 Questions



Consider the following s

$s_{\text{start}} = a$, and a single goal s

(

6.1 Question 10

Assume a breadth-first search algorithm is applied to this search graph. Give a sequence of frontiers that could be computed by the algorithm. a {a, ab} {a, ab, aba, abc}

6.2 Question 11

Assume a depth-first algorithm is applied to the search graph. Give two sequences of frontiers that could be computed by the algorithm The algorithm will always start by computing the frontiers {a}, {ab}, {aba, abc}. Now it could select aba and obtain the frontier {abab, abc}. Then it selects abab and obtains the frontier {ababa, ababc, abc}. It could move on like this by selecting ababa without terminating. However, it could now instead select ababc and obtain the frontier {ababa, abc} and terminate with output ababc .