Homework 1
Computer Science
Spring 2017
B351

Steven Myers

February 2, 2017

All the work herein is mine.

Answers

1. Here are the definitions for the terms listed in Problem 3.10:

   (a) **State** - A particular assignment of value to all variables for a given situation.

   (b) **State space** - All of the possible assignments of values to variable in a scenario.

   (c) **Search tree** - An arrangement of states such that no state is repeated twice when traversing all states.

   (d) **Search node** - A particular node in a search tree that has a cost and ingoing or outgoing paths. Each node contains the cost of the current node, its data (state values), and the cost of visiting other accessible states.

   (e) **Goal** - A final search node with an ideal or target state that we want to obtain to resolve a problem.

   (f) **Action** - An operation that results in a change of state.

   (g) **Transition model** - The process of moving from one state to another when an action is applied.

   (h) **Branching factor** - The maximum number of "branches" or pathways that some search node can have. This is used to limit the search space and prevent a machine from running out of memory before resolving a problem when the search space is too large.

2. A state space where iterative deepning search would perform worse than DFS is one in which the following are true:

   (a) The branching factor is large.

   (b) The depth of each branch is shallow, and the goal node exists in a branch which is deeper than the others.

   A state space such as Figure 1 could produce a circumstance where DFS would perform significantly better than iterative deepening search.

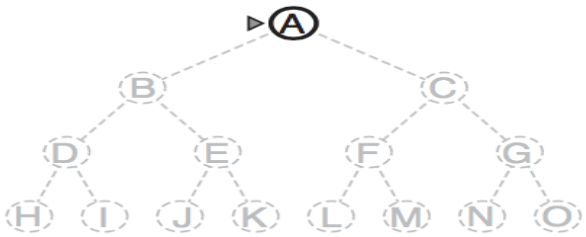

Figure 1: Attach a goal node **S** as a node **O**'s right child. (Image taken from textbook figure 3.16 on page 86, Russel & Norvig)

3. Yes, the graph is consistent. Here are the statements for each node and its sucessors:
   (a) AB: $1 \leq 6$
   (b) AC: $1 \leq 8$
   (c) BC: $4 \leq 4$
4. Solutions provided in rv1.py
5. I extended the rock, paper, scissors game from the last homework assignment. The file is included as *rpgs.py* as requested. This version of the game differs from the previous version since the computer and the human take bets. There is more human-interaction with the game, as you must decide when you should take bets, etc. The computer takes bets based on whether or not it's on a losing or winning streak. You lose the game if you run out of money (less than 10 dollars), and the computer loses the game if it runs out of money.