CS1571
Fall 2019
9/11 Homework

Simran Gidwani

Introduction

Read Russell & Norvig, Chapters 6.1-6.3, 3.5.

- We will be covering 6.1-6.3 in one class (the 9/11 class), so we won't be covering everything in depth. 6.1 will give you a sense of how CSPs are formalized. In class, we will be focusing on 6.3.1 and 6.3.2, and then time permitting, 6.2.1 and 6.2.2.
- 3.5 compares several Informed Search Strategies. We will be focusing on those described in 3.5.1 and 3.5.2.

Answer the following questions:

1. In AC-3, as described in the textbook, the REVISE method is

   a) $O(d^2)$

2. Explain how both variable and value ordering apply to deciding how to fill in a square in the Sudoku.

   Variable ordering can apply to deciding how to fill in the sudoku board because if you use the minimum remaining values heuristic, it chooses the variable that is most likely to result in a failure thus enabling you to prune the search tree. In the case that a variable has no legal values left, the algorithm will automatically detect a failure which would avoid pointless searches through other variables. With the sudoku puzzle, filling in a square using variable ordering will eliminate the possibilities of numbers you have to try because its starts with a square that has the most constraints thus eliminating paths that lead to failure.

   After choosing a variable to examine, the algorithm must decide on the order. This is where value ordering comes into play. The least constraining value heuristic will chose the value that rules out the fewest choices for neighboring variables. This is done to leave the max amount of flexibility for the next variable assignments. Value ordering applies to this problem when choosing what number to place into each square. It allows you to choose the value that has the least amount of constraints making it easier to find just one solution.

3. Identify which type of search algorithm the following approaches represent, choosing from the search algorithms already covered in class and in the readings. The first row of the table is filled in for you:

Approach	**Algorithm**
Expand the deepest node *n* in the frontier of the search tree.	*Depth-first search*
Expand the node *n* with the smallest estimated solution cost.	Breadth-first search
Expand the node *n* that represents the shallowest unexplored node.	Breadth-first search
Expand the node *n* with the lowest path cost.	Breadth-first search
Expand the node *n* that is estimated to be closest to the goal.	Depth-first search