CSE 402: Offline Assignment 4

Solving Sudoku and Futoshiki using CSP techniques

In this assignment, you will have to solve a logic puzzle game involving number placement in a square grid. For both sudoku and futoshiki, the puzzle setter provides a partially completed grid. The solution is a a type of Latin square with an additional constraint. A Latin square is an n × n array filled with n different symbols, each occurring exactly once in each row and exactly once in each column. The additional constraint is different for sudoku and futoshiki.

Sudoku: The objective is to fill a 9×9 grid with digits so that each column, each row, and each of the nine 3×3 sub-grids that compose the grid (also called "boxes", "blocks", "regions", or "sub-squares") contains all of the digits from 1 to 9. So, For sudoku, the additional constraint is that each of the nine boxes contains all of the digits from 1 to 9.

Futoshiki: The puzzle is played on a square grid, such as 5 x 5. So, you have to take the dimension *n* as an input. The objective is to place the numbers 1 to 5 (or whatever the dimensions are) such that each row, and column contains each of the digits 1 to 5. In addition, inequality constraints are also specified between some of the squares, such that one must be higher or lower than its neighbour. So, for futoshiki, the additional constraint is that a set of inequality constraints involving adjacent squares is satisfied.

An example 3X3 Latin square:

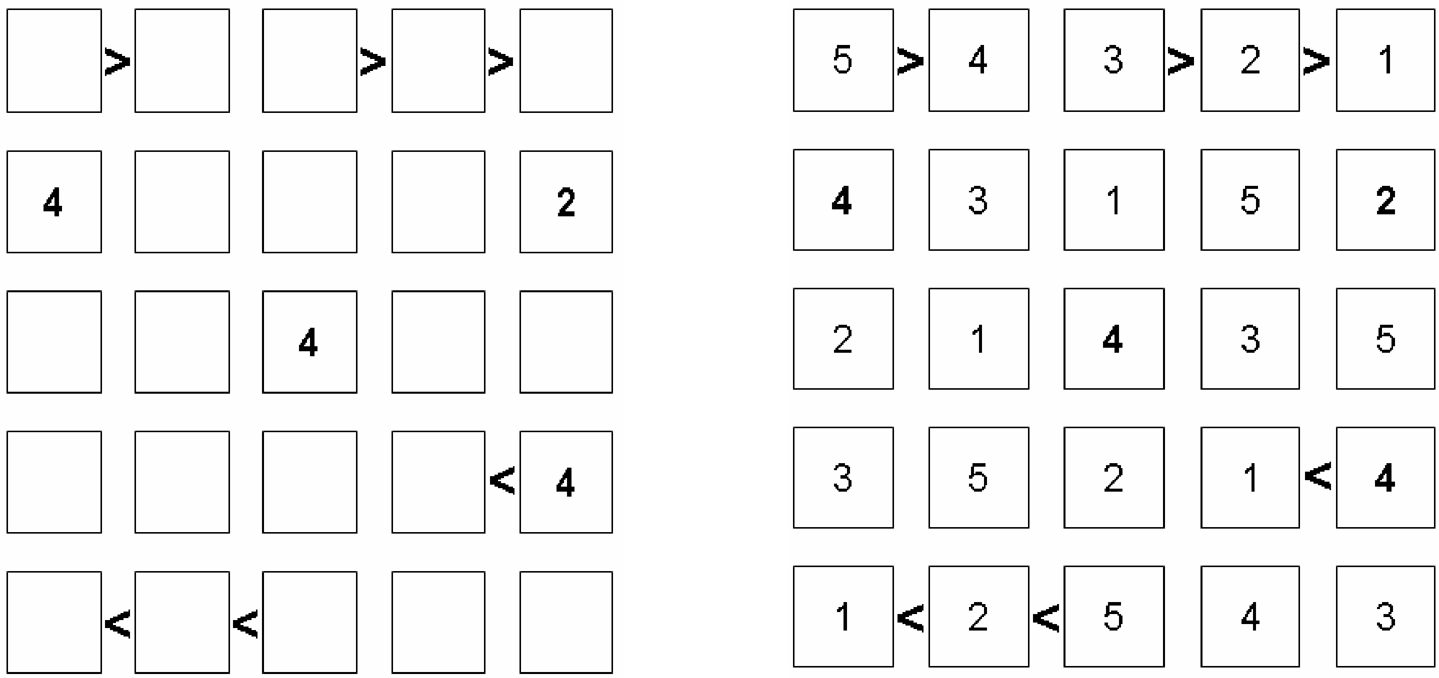
3	1	2
1	2	3
2	3	1

An example sudoku (input and output):

5	3			7				
6			1	9	5			
	9	8					6	
8				6				3
4			8		3			1
7				2				6
	6					2	8	
			4	1	9			5
				8			7	9

5	3	4	6	7	8	9	1	2
6	7	2	1	9	5	3	4	8
1	9	8	3	4	2	5	6	7
8	5	9	7	6	1	4	2	3
4	2	6	8	5	3	7	9	1
7	1	3	9	2	4	8	5	6
9	6	1	5	3	7	2	8	4
2	8	7	4	1	9	6	3	5
3	4	5	2	8	6	1	7	9

An example futoshiki (input and output):



Background knowledge

In a constraint satisfaction problem (csp), we have a set of variables, each of which has a value. The problem is solved by an assignment of values to all the variables such that all the constraints are satisfied.

You should study the chapter-6 (constraint satisfaction problems) from Russell and Norvig's AI text book. Also, http://norvig.com/sudoku.html contains helpful hints for solving both sudoku and futoshiki.

Before you start solving this assignment, you should have a clear understanding of the following terms:

a latin square, a consistent assignment, the Alldiff constraint, the backtracking search for CSP solving, the variable ordering and the value ordering, the min-remaining-values (mrv) heuristic for the variable ordering, the degree heuristic for the variable ordering, the least-constraint-value (lrv) heuristic for the value ordering, interleaving search and inference, forward checking, maintaining arc consistentcy

<u>The assignment task and the report</u>

For section (A1 and B1), the sudoku is to be solved using CSP techniques.

For section (A2 and B2), the futoshiki is to be solved using CSP techniques.

When solving a logic puzzle using CSP techniques, we begin by formulating it as a CSP. This means, we define the variables, the domains, and the constraints. Next, we attempt to solve it by inference and constraint propagation. This allows us to reduce the domains. For hard problems, we need to use interleaving of search and inference.

1. Write the csp formulation of the problem.

2. You will have to implement a backtracking search with constraint propagation. For the constraint propagation, you should have (a) forward checking (b) maintaining-arc-consistency (MAC) (c) powerful constraint propagators of your choice

Write in the report what propagators are being used by your solver.

3. For the search, you should keep a count of how many times you have done a variable selection. Also, you should keep a count of how many times you have done a value selection. For variable selection, you can use (a) MRV heuristic (b) degree heuristic (c) random selection (d) selection of the first available variable. For value selection, you can use (a) LRV heuristic (b) random selection (c) selection of the first available value

Write in the report how you have done the variable and value selection. Also, show (use graph/charts/tables) how your choice affects the performance of the solver. For performance indicators, use runtime and the count of variable and value selection during search. While reporting the performance of random selection, an average should be taken.