

Course: CS131 Artificial Intelligence

Assignment: Constraint Satisfaction Problems

Name: Mingwei Cui

Design Part:

isPossible:

Check conditions for filling each column, row, and 3x3 grid with numbers from 1 to 9 without repeating. If there is a duplicate value in a row, the function returns False. If there is a duplicate value in a column, the function returns False. If there is a duplicate value in a 3x3 grid, the function also returns False. But if none of the three conditions is executed, there is no duplicate, and is valid to put the number. Then the function returns True.

solve:

The function will check if a sudoku is solved. If it is solved then the function will return True. But if it is not solved it will return False.

solve\_sudoku:

For each sudoku grid, each value is checked in the given domain. If the value assigned is within the assignment's limitations, add it to the assignment. If the lower recursion call returns success, the function returns the solution. If the lowest recursive call failed, our temp solution reaches a wrong answer, then we assign the grid value to 0 and repeat the loop. If the loop fails to return a solution, this function return failure and backtrack.

select:

Because this is simple backtracking with no variable changing, we just pick the first empty grid we find and assume that the domain of each grid contains all possible numbers and remove wrong numbers recursively.

Test Part:

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

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

Using the sudoku the assignment given.