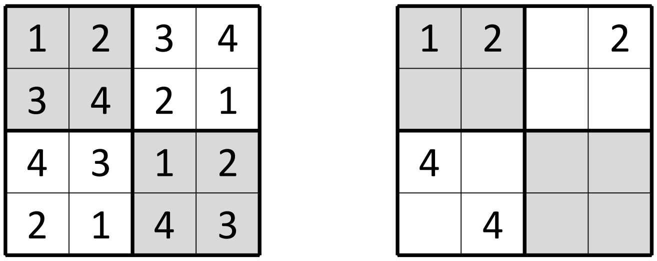
CS188 Spring 2012 Section 3: CSP

1 Mini-Sudoku

Mini-Sudoku is a scaled-down variant of the popular game Sudoku. In mini-Sudoku, one is presented with a 4×4 grid, further partitioned into a 2×2 grid of 2×2 sub-grids, called "regions" (see the figure below). Each cell must be filled in with a number from 1 to 4. Given a partially completed puzzle, the task is to fill in the rest of the numbers subject to the following three constraints:

- A number should only appear once in each row
- A number should only appear once in each column
- A number should only appear once in each 2×2 region



1	2	3	4
3	4	2	1
4	3	1	2
2	1	4	3

(a) Valid Sudoku

1	2		2
4			
	4		

(b) Invalid Sudoku (2's and 4's violate constraints)

Formally, we can view mini-Sudoku as a CSP. Let $s_{i,j} \in \{1, 2, 3, 4\}$ represent the value in the (i, j) cell, with i and j ranging over $\{1, 2, 3, 4\}$ (note that these are indices into the chart, which are distinct from the values that fill the cells!). We can express all of the constraints as binary constraints:

$$\begin{aligned} \text{Rows: } & s_{i,j} \neq s_{i,k} \ \forall i, j, k \text{ with } j \text{ and } k \text{ different} \\ \text{Cols: } & s_{i,j} \neq s_{h,j} \ \forall h, i, j \text{ with } h \text{ and } i \text{ different} \\ \text{Regions: } & s_{i,j} \neq s_{k,l} \ \forall i, j, k, l \text{ with } (i, j) \text{ and } (k, l) \text{ different pairs in the same region} \end{aligned}$$

Now consider the following partially completed Sudoku puzzle (on the next page), focusing on the variables $a = s_{3,3}$, $b = s_{3,4}$, and $c = s_{4,3}$ (you can use a, b, and c as shortcuts to refer to the variables).

3	4	2	
1	2	3	
		a	b
2	3	c	1

(a) Write down the domains of the variables a, b, and c after executing forward-checking.

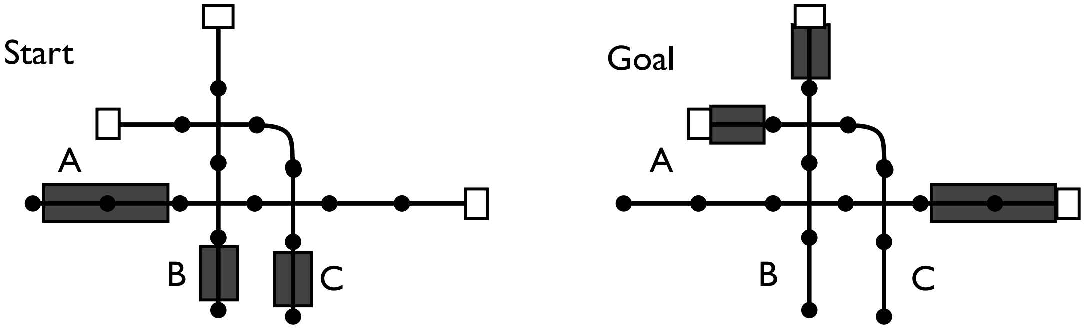
(b) Choose (among a, b, and c) the next variable to assign (according to the MRV heuristic, breaking ties alphabetically if necessary), and assign it (according to the LCV heuristic if necessary).

(c) Write down the domains of the two remaining variables after executing forward-checking again.

(d) What is the test for when a CSP assignment has failed (and the solver must backtrack)?

2 Trains (⋆)

A train scheduler must decide when trains A, B and C should depart. Once a train departs, it moves one space along its track each hour (in discrete jumps) until it arrives at its destination platform. Each train can depart at 1, 2 or 3 pm. The scheduler has two restrictions: All trains must leave at different times, and two trains should not both occupy crossing sections of track after any one hour time step is over. Note that train A is two spaces long. Also note that the collision constraint is enforced only at the conclusion of every hour - time is discrete in this problem.



a) Describe the constraint satisfaction problem that, when solved, will tell the train scheduler when each train should depart. Let the variables A, B and C represent the departure times of the three trains.

b) Draw the constraint graph for the CSP you defined.

c) After selecting $A = 2$, cross out all values for B and C eliminated by forward checking.

A	B	C
2	1 2 3	1 2 3

d) Cross out all values eliminated by arc consistency before assigning any variables.

A	B	C
1 2 3	1 2 3	1 2 3

e) After selecting $A = 2$, cross out all values for B and C eliminated by arc consistency.

A	B	C
2	1 2 3	1 2 3

f) Describe the execution of backtracking search using forward checking and the minimum remaining values (MRV) and least constraining values (LCV) heuristics. Specifically, in what order are the variables assigned and what values do they take? Start by assigning variable A. You may not need to fill all the lines below:
(1) variable A is assigned value . (2) variable is assigned value .
(3) variable is assigned value . (4) variable is assigned value .
(5) variable is assigned value . (6) variable is assigned value .

Note: Lines (2) and (3) may be switched.