Simple Sudoko

3	4		
		3	
		4	2

The board is a 4-by-4 square, and each box can have a number from 1 through 4. In each row and column, a number can only appear once. Furthermore, in each group of 2-by-2 boxes outlined with a solid border, each of the 4 numbers may only appear once as well. Every cell is represented by a variable. We denote the square at i^{th} row and j^{th} column as X_{ij}, and we index the row and column from 1.

Backtracking

(1) Write down constraints of this simple Sudoko problem.

(2) Consider the backtracking search that will assign X_{13}, X_{14}, X_{21} in order during the search procedure. Draw all the branches in search tree for X_{13}, X_{14}, X_{21}.

3	4	X_{13}	X_{14}
X_{21}		3	
		4	2

Solutions

The constraints are:

1. Each box can only take on values 1, 2, 3, or 4: For all i, j, $X_{ij} \in \{1, 2, 3, 4\}$
2. 1, 2, 3, and 4 may only appear once in each row: For all i, j, k, $X_{ij} \neq X_{ik}$ if $j \neq k$
3. 1, 2, 3, and 4 may only appear once in each column: For all i, j, k, $X_{ji} \neq X_{ki}$ if $j \neq k$
4. 1, 2, 3, and 4 may only appear once in each set of 2-by-2 boxes with solid borders.

   $X_{ij} \neq X_{kl}$ if $i \neq k$ or $j \neq l$, $1 \leq i, j, k, l \leq 2$

   $X_{ij} \neq X_{kl}$ if $i \neq k$ or $j \neq l$, $1 \leq i, k \leq 2$, $3 \leq j, l \leq 4$

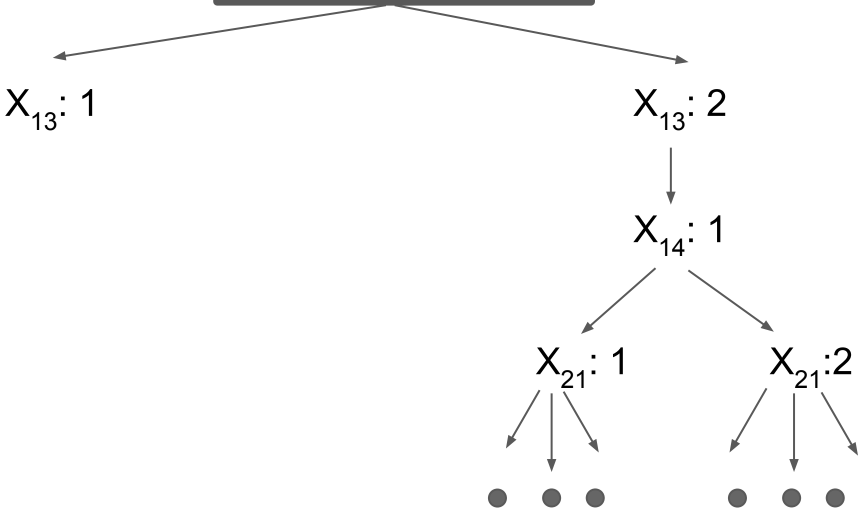
   $X_{ij} \neq X_{kl}$ if $i \neq k$ or $j \neq l$, $1 \leq j, l \leq 2$, $3 \leq i, k \leq 4$

   $X_{ij} \neq X_{kl}$ if $i \neq k$ or $j \neq l$, $3 \leq i, j, k, l \leq 4$

5. $X_{11}=3$, $X_{12}= 4$, $X_{23}= 3$,$X_{33}=4$, $X_{34}= 2$

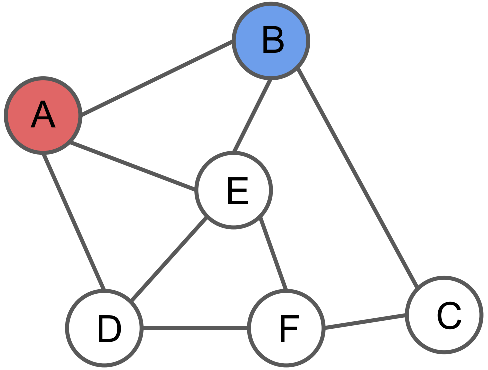
Solutions

3	4	X_{13}	X_{14}
X_{21}		3	
		4	2



Explanation: When assigning variable X_{13}, X_{13}:3 and X_{13}:4 violates the constraint. As a result, we do not assign those values. Similarly, when X_{13} is assigned with 1, no valid value can be assigned to X_{14}, so the subtree does not expand anymore. Same principle to extend the subtree from X_{14}:2.

Filtering & Ordering



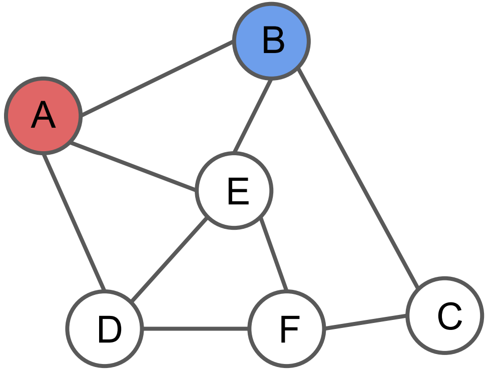
Now we want to color the graph with **Red**, **Blue**, **Green** without coloring two adjacent nodes with the same color. Now we have assigned Red to node A and Blue to node B.

(1) If we assign **Red** to node C, what will be the remaining legal values in D, E, F after running (a) forwarding checking (b) arc consistency?

(2) (MRV) Based on the MRV rule, which variable should be chosen next?

(3) (LCV) Based on the LCV principle, which value should we choose if we want to assign color to Node D first? (Assume we use *forward checking* for the filtering step)

Solutions



Forward checking

D	B, G
E	G
F	B, G

Arc consistency

D	B
E	G
F	Empty

or

D	Empty
E	G
F	B

MRV: In order to decide which variable to assign based on MRV principle, we can list all the legal values left in the domain. Legal values for each node is listed in the table on the right. As we can, node E has the fewest options for color, therefore we will choose E.

C	R, G
D	B, G
E	G
F	R, B, G

Solutions

Based on **LCV** principle, we list all left legal values for C, E, F based on different values for D. As we can see from the table, when we choose Blue for node D, it has more legal values left in the domain. So we should assign **Blue** to node D if we are going to assign node D next.

D	B	G
C	R, G	R, G
E	G	empty
F	R, G	R, B