National University of Singapore
School of Computing
CS3243: Foundations of Artificial Intelligence
Solutions for Tutorial 4

Readings: AIMA Chapter 5

1. Consider the `AC-3(csp)` algorithm (reproduced below), can the last line "add (X_k, X_i) to queue" be replaced with "if $X_k \neq Xj$ then add (X_k, X_i) to queue"? Justify your answer.

```
function AC-3( csp) returns the CSP, possibly with reduced domains
   inputs: csp, a binary CSP with variables {X1, X2, ..., Xn}
   local variables: queue, a queue of arcs, initially all the arcs in csp

   while queue is not empty do
      (Xi, Xj) ← REMOVE-FIRST(queue)
      if REMOVE-INCONSISTENT-VALUES(Xi, Xj) then
         for each Xk in NEIGHBORS[Xi] do
            add (Xk, Xi) to queue
--------------------------------------------------------------------------
function REMOVE-INCONSISTENT-VALUES( Xi, Xj) returns true iff succeeds
   removed ← false
   for each x in DOMAIN[Xi] do
      if no value y in DOMAIN[Xj] allows (x,y) to satisfy the constraint Xi ↔ Xj
         then delete x from DOMAIN[Xi];  removed ← true
   return removed
```

Yes.

Case (1): (X_j, X_i) has not been processed. In this case, (X_j, X_i) is in queue and there is no need to add it.

Case (2): (X_j, X_i) has been processed and some domain value(s) of X_i has just been removed. None of the deleted values xi in the domain of Xi is such that there is some value x_j in the domain of X_j and (x_i, x_j) satisfy the constraint between X_i and X_j. Hence there is no need to add the arc (X_j, X_i) to queue, since there is no x_j that can be removed because of the removal of the just deleted x_i.

2. Consider the following constraint satisfaction problem:

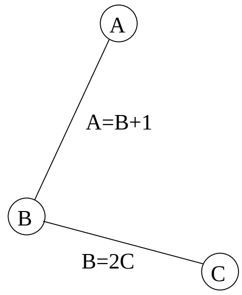
   Variables:
   $$A, B, C$$

   Domains:
   $$D_A = D_B = D_C = \{0, 1, 2, 3, 4\}$$

   Constraints:
   $$\begin{aligned} A &= B + 1 \\ B &= 2C \end{aligned}$$

Construct a constraint graph for this problem. Show a trace of the `AC-3` algorithm on this problem. Assume that initially, the arcs in queue are in the order $\{(A, B), (B, A), (B, C), (C, B)\}$.

Constraint Graph:



Original domains:

$$D_A = D_B = D_C = \{0, 1, 2, 3, 4\}$$

Content of queue and domain of variables at the end of each iteration:

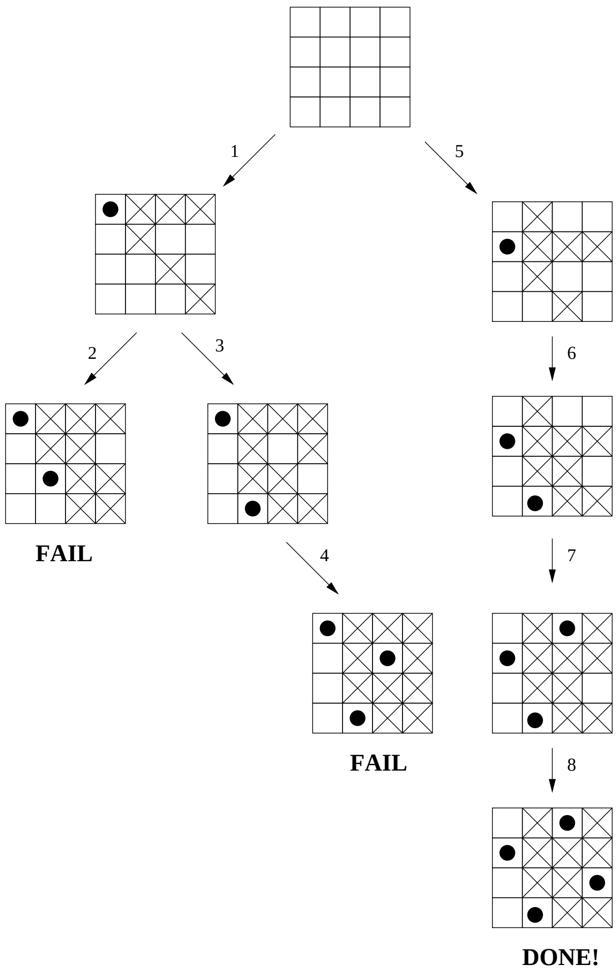
New DV	*queue*
	(A,B) (B,A) (B,C) (C,B)
$D_A = \{1, 2, 3, 4\}$	*(B,A) (B,C) (C,B)*
$D_B = \{0, 1, 2, 3\}$	*(B,C) (C,B)*
$D_B = \{0, 2\}$	*(C,B) (A,B)*
$D_C = \{0, 1\}$	*(A,B)*
$D_A = \{1, 3\}$	

Allowable domain values:

$$\begin{aligned} D_A &= \{1, 3\} \\ D_B &= \{0, 2\} \\ D_C &= \{0, 1\} \end{aligned}$$

3. Consider the 4-queens problem on a 4×4 chess board. Suppose the leftmost column is column 1, and the topmost row is row 1. Let Q_i denote the row number of the queen in column i, $i = 1, 2, 3, 4$. Assume that variables are assigned in the order Q_1, Q_2, Q_3, Q_4, and the domain values of Q_i are tried in the order $1, 2, 3, 4$. Show a trace of the backtracking algorithm with forward checking to solve the 4-queens problem.

   The following is the trace of the search tree:



4. Show a trace of the backtracking algorithm with forward checking to solve the cryptarithmetic problem shown in Figure 1. Use the most constrained variable heuristic, and assume that the domain values (digits) are tried in ascending order (i.e., 0, 1, 2, $\cdots$).

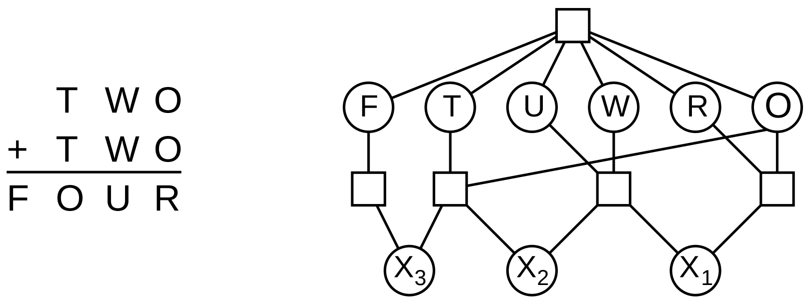


Figure 1: Cryptarithmetic puzzle.

The following is the trace:

Assignments									Remarks
X1=0	X2=0	X3=0							X1, X2, X3 are the most constrained variables. X3=0 forces F=0, which is not possible.
		X3=1	F=1	T=2					T is the most constrained variable, with domain = {2,,9}, whereas the remaining variables have domain = {0,2,,9}. T=2 results in no satisfying value for O.
				T=3					T=3 results in no satisfying value for O.
				T=4					T=4 results in no satisfying value for O.
				T=5	O=0				T=5 uniquely determines the value for O. Fail since R has no satisfying value.
				T=6	O=2	R=4	W=0		T=6 uniquely determines the value for O and R. Fail since there is no satisfying value for U.
							W=3		Fail since there is no satisfying value for U.
							W=5		Fail since there is no satisfying value for U.
							W=7		Fail since there is no satisfying value for U.
							W=8		Fail since there is no satisfying value for U.
									Fail since there is no satisfying value for U.
				T=7	O=4	R=8	W=0		uniquely determines the value for O and R. Fail since there is no satisfying value for U.
							W=2		Fail since there is no satisfying value for U.
							W=3	U= 6	Succeeds!