16.410-13 Principles of Automated Reasoning and Decision Making

Problem Set 3

Problem 1: Modeling for Constraint Programming (40 Points)

You are a screen writer designing the story board for an episode of a sitcom “Buddies”, a “Friends” knockoff, with the ever adorable John, Mary, Tim, Lisa and Bill. You have decided to center the hour episode around an intimate table in a restaurant that has only three chairs. You would like the story to show each “buddy” at some point during the episode (part of their contract), and you want them interacting happily together. But there seems to be some dissension amongst the cast:

- John refuses to be seated with Mary or Tim,
- There is a dialogue with John talking with Lisa,
- The romance between Lisa and Tim requires time at the table for a kiss,
- Bill only gets along with John if Lisa is there as well,
- John and Lisa, as the stars, should be on camera for 40 minutes, while the remainder should be on for 20 minutes each.

You may assume that the characters change locations only at the breakpoints between the show’s 20-minute segments.

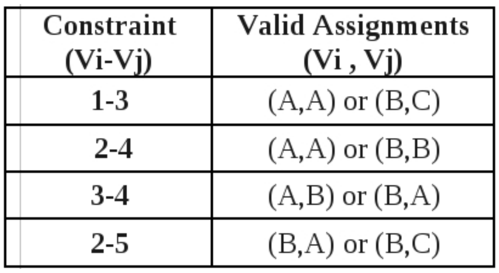
You should express this problem in terms of a constraint satisfaction problem. You should identify the variables, domains and the constraints.

Part A: First assume that there are nine variables, each representing one of the chairs during one of the three 20-minute segments of the show. Write down the domain of these variables and express the constraints. How many different possible assignments are there?

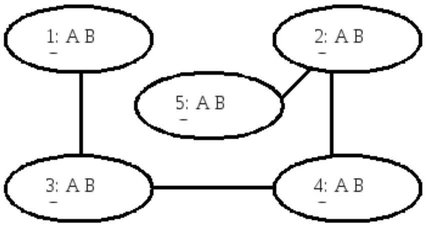
Part B: Now take a people-centric approach. Assume that there exists one variable for each person, and the value a variable takes is the chair assignments for each 20-minute segment (note that at a given segment a person might be assigned no chair). Write down the domain of all the variables and comment on the number assignments.

Problem 2: Constraint Propagation (20 points)

Consider the following constraint graph. There are five variables, denoted 1-5. Each variable has a domain of three values: $\{A, B, C\}$. The only valid assignments to pairs of constraints variables are given in the following table.



Constraint (Vi-Vj)	Valid Assignments (Vi , Vj)
1-3	(A,A) or (B,C)
2-4	(A,A) or (B,B)
3-4	(A,B) or (B,A)
2-5	(B,A) or (B,C)



Part A: Repeatedly perform constraint propagation on the above constraint graph until you achieve arc consistency, by crossing out the eliminated values on each node of the graph. List the remaining elements for each domain.

Part B: What is the maximum number of possible solutions, based only on the knowledge of the remaining values?

Part C: In general, does constraint propagation guarantee that all infeasible solutions are pruned? Explain.

Problem 3: Correctness of Constraint Propagation (40 Points)

Recall that the **constraint satisfaction problem** was defined as follows.

An constraint satisfaction problem is a tuple $\langle V, D, R\rangle$, where V is the set of variables, D is a set of domains, and R is a set of constraints. Each variable $v_i \in V$ is denoted with its index i. For each variable v_i, there exists a domain D_i. For any pair of variables, v_i and v_j, there exists a set of constraints denoted as R_{ij}. If two variable assignments d_i and d_j (for variables v_i and v_j, respectively) satisfy the constraint $R_{i,j}$, we denote this as $\langle d_i, d_j\rangle \in R_{ij}$.

The **arc consistency problem** is given as follows.

Given a constraint satisfaction problem $\langle V, D, R\rangle$, find a set of restricted domains $D'_i \subset D_i$ for all variables $v_i \in V$ such that for any value $d_i \in D'_i$ there exists some variable $v_j \in V$ and some domain value $d_j \in D'_j$ of v_j such that $\langle d_i, d_j\rangle \in R_{ij}$, i.e., $\langle d_i, d_j\rangle$ satisfies a constraint in R_{ij}.

The two arc consistency algorithms we have discussed in the class were AC-1 (given in Algorithm 2) and AC-3 (given in Algorithm 3). Notice that both algorithms use the `REVISE` procedure given in Algorithm 1. In all the algorithms, `Arcs` denotes the set of all constraints.

Prove that AC-1 and AC-3 algorithms are complete: if there is an set of restricted domains (each of them different than the empty set, i.e., $D'_i \neq \emptyset$ for all i) as described in the *arc consistency problem*, then both AC-1 and AC-3 will find a set of restricted domains (each domain is different than the empty set).

Algorithm 1: REVISE$(\langle x_i, x_j\rangle)$

```
DELETED ← FALSE;
for each $a_i \in D_i$ do
    if there is no $a_j \in D_j$ such that $\langle a_i, a_j\rangle \in R_{ij}$ then
        Delete $a_i$ from $D_i$;
        DELETED ← TRUE;
return DELETED
```

Algorithm 2: AC $-$ 1

```
$Q \leftarrow \{\langle x_i, x_j\rangle \mid \langle x_i, x_j\rangle \in$ Arcs$\}$;
repeat
    CHANGE ← FALSE;
    for each $\langle x_i, x_j\rangle \in Q$ do
        if REVISE($\langle x_i, x_j\rangle$) = TRUE then
            CHANGE ← TRUE;
until CHANGE = FALSE ;
```

Algorithm 3: AC − 3

```
$Q \leftarrow \{\langle x_i, x_j \rangle \mid \langle x_i, x_j \rangle \in \texttt{Arcs}\}$;
while $Q$ is not empty do
    Select and delete any arc $\langle x_i, x_j \rangle$ from $Q$;
    if REVISE($\langle x_i, x_j \rangle$) then
        $Q \leftarrow Q \cup \{\langle x_k, x_i \rangle \mid k \neq i, k \neq j\}$;
```