

**National University of Singapore
School of Computing
CS3243 Introduction to AI**

Tutorial 5: CSPs

Issued: February 13, 2020

Due: Week 7, in the tutorial class

Important Instructions:

- *Your solutions for this tutorial must be TYPE-WRITTEN.*
- *Make TWO copies of your solutions: one for you and one to be SUBMITTED TO THE TUTOR IN CLASS. Your submission in your respective tutorial class will be used to indicate your CLASS ATTENDANCE. Late submission will NOT be entertained.*
- *You may discuss the content of the questions with your classmates. But everyone should work out and write up ALL the solutions by yourself.*
- *None of these questions will be graded, but you may expect similar questions in the MIDTERM.*

1. Consider the following constraint satisfaction problem:

Variables:

$$A, B, C$$

Domains:

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

Constraints:

$$A = B + 1$$

$$B = 2C$$

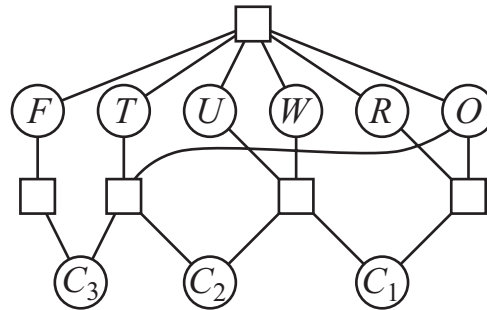
Construct a constraint graph for this problem. Show a trace of the AC-3 algorithm on this problem. Assume that initially, the arcs in queue are in the order $\{(A, B), (B, A), (B, C), (C, B)\}$.

2. 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.

3. Show a trace of the backtracking algorithm with forward checking to solve the cryptarithmic 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, \dots).

$$\begin{array}{r}
 T \ W \ O \\
 + \ T \ W \ O \\
 \hline
 F \ O \ U \ R
 \end{array}$$

(a)



(b)

Figure 1: Cryptarithmic puzzle.

4. Consider the *item allocation problem*. We have a group of people $N = \{1, \dots, n\}$, and a group of items $G = \{g_1, \dots, g_m\}$. Each person $i \in N$ has a utility function $u_i : G \rightarrow \mathbb{R}_+$. The constraint is that every person is assigned *at most one item*, and each item is assigned to *at most one person*. An allocation simply says which person gets which item (if any). In what follows, you *must* use *only* the binary variables $x_{i,j} \in \{0, 1\}$, where $x_{i,j} = 1$ if person i receives the good g_j , and is 0 afterwards.
- (a) Write out the constraints: ‘each person receives no more than item’ and ‘each item goes to at most one person’, using only the $x_{i,j}$ variables¹.
 - (b) Suppose that people are divided into *disjoint types* N_1, \dots, N_k (think of, say, genders or ethnicities), and items are divided into *disjoint blocks* G_1, \dots, G_ℓ . We require that every person type N_p is allowed to take no more than λ_{pq} items from block G_q . Write out this constraint using the $x_{i,j}$ variables.
 - (c) We say that player i *envies* player i' if the utility that player i has from their assigned item is strictly lower than the utility that player i has from the item assigned to player j . Write out the constraints that ensure that in the allocation, no player envies any other player. You may assume that the validity constraints from (a) hold.

¹You may use simple algebraic functions $-, +, \times, \div$, and numbers