CS410: Artificial Intelligence 2021 Fall
Homework 2: Constraint Satisfaction Problems
Due date: 23:59:59 (GMT +08:00), October 22 2021

1. **A* graph search**. Consider the following undirected graph shown in Figure 1 where we are searching from start state A to goal state G. The number over each edge is the transition cost. Additionally, we are given a heuristic function h as follows: $\{h(A) = 7, h(B) = 5, h(C) = 6, h(D) = 4, h(E) = 3, h(F) = 3, h(G) = 0\}$. Assume that, in case of ties, the search procedure uses an alphabetical order for tie-breaking.

   Find the sequence of nodes expanded by A* graph search algorithm, with problem-solving steps (i.e., updates for the frontier and explored set).

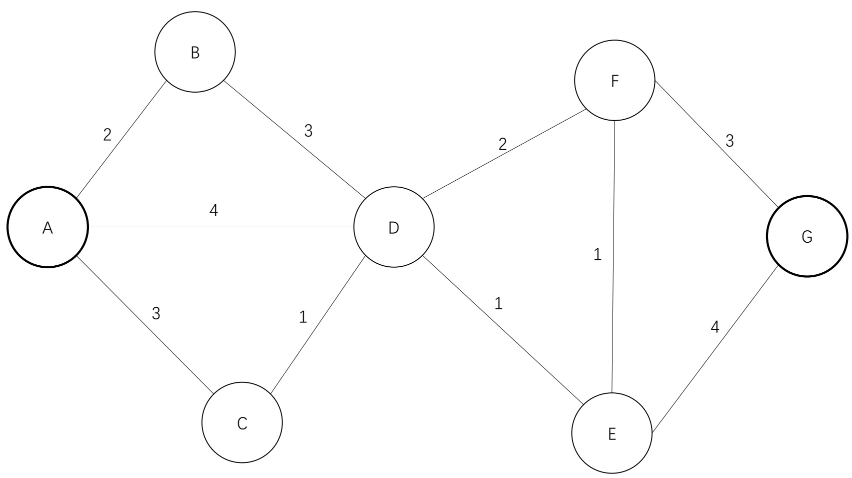


Figure 1: Problem 1.

2. **CSP formulation**. Consider the following three problems:

   (a) Rectilinear floor-planning: find non-overlapping places in a large rectangle for a number of smaller rectangles.

   (b) Class scheduling: There is a fixed number of professors and classrooms, a list of classes to be offered, and a list of possible time slots for classes. Each professor has a set of classes that he or she can teach.

(c) Hamiltonian tour: given a network of cities connected by roads, choose an order to visit all cities in a country without repeating any.

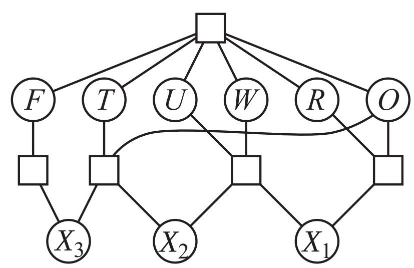
Determine each as a planning problem or an identification problem and explain why. Give precise formulations for each problem as constraint satisfaction problems.

Recall a CSP consists of three components, X, D, and C. Note that there are many valid formulations, but you only need to provide one.

3. **Forward checking**. Solve the cryptarithmetic problem shown in Figure 3 step by step, using the strategy of backtracking with forward checking. Assume the variable order is $X_3 \to F \to X_2 \to X_1 \to O \to T \to R \to U \to W$, and the value order is increasing. Note that different variables have different domains (e.g., the domain for X_3 is $\{0, 1\}$, the domain for O is $\{0, 1, \ldots, 9\}$, and the domain for F is $\{1, 2, \ldots, 9\}$).

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

(a)



(b)

(a) A cryptarithmetic problem. Each letter stands for a distinct digit; the aim is to find a substitution of digits for letters such that the resulting sum is arithmetically correct, with the added restriction that no leading zeroes are allowed. (b) The constraint hypergraph for the cryptarithmetic problem, showing the *Alldiff* constraint (square box at the top) as well as the column addition constraints (four square boxes in the middle). The variables X_1, X_2, and X_3 represent the carry digits for the three columns.

Figure 2: Problem 3.

4. **AC-3**. Consider the following CSP:

- Variables: A, B, C, D
- Domain: $\{1, 2, 3\}$
- Constraints: $A \neq B, B > C, C < D$

Solve the problem using the strategy of backtracking search with AC-3 algorithm. Give the problem-solving steps, specifying each assignment when backtracking and the consequence of each pop operation in AC-3 (i.e., what values you cross off and which arcs you push to the queue).

Suppose the value order is descending and the variable order is alphabetical, which both queue initialization and neighbor consideration follow (i.e., the queue is initialized as $A \to B, B \to A, \dots$).

5. **K-consistency & tree structure**. Consider the following three questions:

   (a) Give a concrete CSP example to show that k-consistency does not imply $(k+1)$-consistency for some $k \geq 2$.

   (b) Give a concrete CSP example to show that k-consistency does not imply $(k-1)$-consistency for some $k \geq 3$.

   (c) Why graphs with cycles can not be applied with the algorithm for tree-structured CSPs (introduced in Page 77-83, Lecture 4)? What step in the analysis fails? Why this step holds for trees? Give an example to explain the failure.