1 Course Scheduling

You are in charge of scheduling for computer science classes that meet Mondays, Wednesdays and Fridays. There are 5 classes that meet on these days and 3 professors who will be teaching these classes. You are constrained by the fact that each professor can only teach one class at a time. The classes are:

1. Class 1 - Intro to Programming: meets from 8:00-9:00am
2. Class 2 - Intro to Artificial Intelligence: meets from 8:30-9:30am
3. Class 3 - Natural Language Processing: meets from 9:00-10:00am
4. Class 4 - Computer Vision: meets from 9:00-10:00am
5. Class 5 - Machine Learning: meets from 10:30-11:30am

The professors are:

1. Professor A, who is qualified to teach Classes 1, 2, and 5.
2. Professor B, who is qualified to teach Classes 3, 4, and 5
3. Professor C, who is qualified to teach Classes 1, 3, and 4.

1. Formulate this problem as a CSP problem in which there is one variable per class, stating the domains (after enforcing unary constraints), and binary constraints. Constraints should be specified formally and precisely, but may be implicit rather than explicit.

Solution:
2. Draw the constraint graph associated with your CSP.

\[
\text{Solution:}
\]

\[
\begin{array}{c}
C_1 \\
C_2 \\
C_3 \\
C_4 \\
C_5
\end{array}
\]

---

2 CSP Futoshiki

Futoshiki is a Japanese logic puzzle that is very simple, but can be quite challenging. You are given an \( n \times n \) grid, and must place the numbers 1, \ldots, \( n \) in the grid such that every row and column has exactly one of each. Additionally, the assignment must satisfy the inequalities placed between some adjacent squares.

Left is an instance of this problem, for size \( n = 4 \). Some of the squares have known values, such that the puzzle has a unique solution.

(The letters mean nothing to the puzzle, and will be used only as labels with which to refer to certain squares). Note also that inequalities apply only to the two adjacent squares, and do not directly constrain other squares in the row or column.

Let’s formulate this puzzle as a CSP. We will use \( 4^2 \) variables, one for each cell, with \( X_{ij} \) as the variable for the cell in the \( i \)th row and \( j \)th column (each cell contains its \( i, j \) label in the top left corner). The only unary constraints will be those assigning the known initial values to their respective squares (e.g. \( X_{34} = 3 \)).
a. Complete the formulation of the CSP using only binary constraints (in addition to the unary constraints specified above. In particular, describe the domains of the variables, and all binary constraints you think are necessary. You do not need to enumerate them all, just describe them using concise mathematical notation. You are not permitted to use n-ary constraints where \( n \geq 3 \)

**Solution:**

Domains: \( X_{ij} \in \{1, 2, 3, 4\}, \forall i, j \)

Unary constraints: \( X_{34} = 3, X_{43} = 2 \)

Inequality binary constraints: \( X_{11} < X_{12}, X_{13} < X_{23}, X_{14} < X_{24}, X_{32} < X_{22}, X_{32} < X_{42} \)

Row binary constraints: \( X_{ij} \neq X_{ik}, \forall i, j, k, j \neq k \)

Column binary constraints: \( X_{ij} \neq X_{kj}, \forall i, j, k, i \neq k \)

b. After enforcing unary constraints, consider the binary constraints involving \( X_{14} \) and \( X_{24} \).

Enforce arc consistency on just these constraints and state the resulting domains for the two variables.

**Solution:**

\( X_{14} \in \{1, 2\}, X_{24} \in \{2, 4\} \). Note that both threes are removed from the column constraint with \( X_{34} \).

c. Suppose we enforced unary constraints and ran arc consistency on this CSP, pruning the domains of all variables as much as possible. After this, what is the maximum possible domain size for any variable? [Hint: consider the least constrained variable(s); you should not have to run every step of arc consistency.]

**Solution:**

The maximum possible domain size is 4 (i.e., no values are removed from the original domain). Consider \( X_{21} \) – we will not be able to eliminate any values from its domain through arc consistency.

d. Suppose we enforced unary constraints and ran arc consistency on the initial CSP in the figure above. What is the maximum possible domain size for a variable adjacent to an inequality?

**Solution:**

The maximum domain size is 3 – you must always eliminate either 1 or 4 from a variable participating in an inequality constraint.

e. By inspection of column 2, we find it is necessary that \( X_{32} = 1 \), despite not having found an assignment to any of the other cells in that column. Would running arc consistency find this requirement? Explain why or why not.
Solution:
No, arc consistency would not find this requirement. Enforcing the $X_{32} \Rightarrow X_{42}$ and the $X_{42} \Rightarrow X_{43}$ arc leaves $X_{42}$ with a domain of $\{3, 4\}$. Enforcing the $X_{32} < X_{22}$ constraints leaves $X_{32} \in \{1, 2, 3\}$ and $X_{22} \in \{2, 3, 4\}$. Enforcing that they are all different does not remove any values. After this point, every arc in this column is consistent and $X_{32}$ is not required to be 1.

3 CSPs: Properties

a. When enforcing arc consistency in a CSP, the set of values which remain when the algorithm terminates does not depend on the order in which arcs are processed from the queue (True / False).

Solution:
True

b. In a general CSP with $n$ variables, each taking $d$ possible values, what is the maximum number of times a backtracking search algorithm might have to backtrack (i.e. the number of the times it generates an assignment, partial or complete, that violates the constraints) before finding a solution or concluding that none exists?

$0 \quad O(1) \quad O(nd^2) \quad O(n^2d^3) \quad O(d^n)$

Solution:
$O(d^n)$