Assignment6 – Constraint Programming

Problem 6.1 (Scheduling CS Classes)

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:

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

The professors are:

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

1. Formulate this problem as a constraint network in which there is one variable per class, stating the domains, and constraints. Constraints should be specified formally and precisely, but may be implicit rather than explicit.
2. Give the constraint graph associated with your constraint network (e.g. by giving the edges or drawing it).
3. Show the domains of the variables after running arc-consistency on this initial graph (after having already enforced any unary constraints).
4. Give one solution to constraint network
5. Your constraint network should look nearly tree-structured. Briefly explain (one sentence or less) why we might prefer to solve tree-structured constraint networks.
6. Name (or briefly describe) a standard technique for turning these kinds of nearly tree-structured problems into tree-structured ones.

Objective: apply constraint graph

Objective: apply acn

Problem 6.2 (CSP as a Search Problem)

We consider a constraint network $P := \langle V, D, C \rangle$ with

- a set V of variables
- a family D of domains D_v for $v \in V$

- a family C of constraints $C_{uv} \subseteq D_u \times D_v$ for $u, v \in V$, $u \neq v$ where C_{uv} is the dual of C_{vu}

Note: We assume here that a constraint C_{uv} is given for all pairs of unequal variables – if we want to omit a constraints, we can simply assume $C_{uv} = D_u \times D_v$ or C_{uv} is satisfied, i.e., all pairs are allowed and thus there is no constraint. That could be problematic in implementations, but is practical on paper.

Define the search problem $\langle \mathcal{S}, \mathcal{A}, \mathcal{T}, \mathcal{I}, \mathcal{G} \rangle$ corresponding to P.

Note: This problem formalizes the informal statement that constraint networks are a special case of search problems.

Problem 6.3 (Basic Definitions)
Consider the following constraint network $\langle V, D, C \rangle$

- $V = \{a, b, c, d\}$
- $D_a = \texttt{bool}$, $D_b = D_c = \{0, 1, 2, 3\}$, $D_d = \{0, 1, 2, 3, 4, 5, 6\}$
- Constraints C:
  - if a, then $b \leq 2$
  - if $c < 2$, then a
  - $b + c < 4$
  - $b > d$
  - $d = 2c$

1. Give all solutions.
2. Give an inconsistent total variable assignment.
3. Give all consistent partial assignments α such that $\text{dom}(\alpha) \subseteq \{a, b\}$.

Problem 6.4 (CSP Formalization)
Consider the following constraint network$\Pi := \langle V, D, C \rangle$:

- Variables $V = \{x, y, z\}$
- Domains D: $D_x = \{0, 1, 2\}$, $D_y = \{1, 2\}$, and $D_z = \{0, 1\}$
- Constraints C: $x \neq y$, $y > z$

Objective: understand acv

1. Give all pairs (v,w) of variables such that v is arc-consistent relative to w.

2. Give all solutions that would remain if we added the constraint $x \neq z$ to Π
3. Without using that additional constraint, now assume we assign $y = 1$ in Π and apply forward checking. Give the resulting domains D_x, D_y, D_z.

Objective: apply solution

Objective: apply domain

Objective: apply forward checking

Problem 6.5 (Kalah Tournament)

This is an extraordinary problem, in which we implement adversarial search as a tournament. You can implement all search methods, e.g., to simulate a move or compute the full game tree etc.
Submission parameters:

- Team size: 3 people per team
- Deadline: 2025-01-07
- Site: The submission site will be opened later.
- Format: The details will be published later on the studon forum. But you can use any programming language, and your program might be subject to resource constraints (overall space for the binary, time per move, etc.).
- Points: Submissions that are better than a relatively low baseline (e.g., win against a player that makes random moves) will receive 10 points. The team with the best agent receives an additional 10 points, the 2nd team 9 points, the 3rd 8 etc.
- The Kalah points are bonus-bonus, i.e., they increase your bonus but are not needed to max out the bonus. 20 Kalah points will be worth roughly 20% of all quizzes combined.

Further details will be announced in the forum as they come up.