Assignment5 – Constraint Programming

Problem 5.1 (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.

Solution:

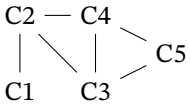
Variables	Domains
C1	C
C2	B,C
C3	A,B,C
C4	A,B,C
C5	B,C

Constraints: C1 $\neq$ C2, C2 $\neq$ C3 , C3 $\neq$ C4, C4 $\neq$ C5, C2 $\neq$ C4, C3 $\neq$ C5
BTW, the Solution of the CSP: C_1 = C, C_2 = B, C_3 = C, C_4 = A, C_5 = B. One other solution is possible (where C_3 and C_4 are switched).

Objective: apply constraint graph

2. Give the constraint graph associated with your constraint network (e.g. by giving the edges or drawing it).

Solution:



Objective: apply acn

3. Show the domains of the variables after running arc-consistency on this initial graph (after having already enforced any unary constraints).

Solution:

Variable	Domain
C_1	C
C_2	B
C_3	A,C
C_4	A,C
C_5	B,C

Note that C_5 cannot possibly be C, but arc consistency does not rule it out.

4. Give one solution to constraint network

Solution: C1 = C, C2 = B, C3 = C, C4 = A, C5 = B. One other solution is possible (where C3 and C4 are switched).

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.

Solution: Minimal answer: Can solve them in polynomial time. If a graph is tree structured (i.e. has no loops), then the CSP can be solved in quadratic time as compared to general CSPs, where worst-case time is exponential. For tree-structured CSPs you can choose an ordering such that every node's parent precedes it in the ordering. Then you can greedily assign the nodes in order and will find a consistent assignment without backtracking.

6. Name (or briefly describe) a standard technique for turning these kinds of nearly tree-structured problems into tree-structured ones.

Solution: Minimal answer: cutset conditioning. One standard technique is to instantiate cutset, a variable (or set of variables) whose removal turns the problem into a tree structured CSP. To instantiate the cutset you set its values in each possible way, prune neighbors, then solve the reduced tree structured problem (which is fast).

Problem 5.2 (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.

Solution: The search problems is defined as follows:

- states are the variable assignments, i.e., $\mathcal{S}$ is the set of the partial functions with domain V that map each $v \in V$ to an element of D_v. More formally, we can write this as

$$\mathcal{S} = \{a \ : \ V \rightharpoonup \bigcup_{v \in V} D_v | \forall v \in \mathrm{dom}(a).a(v) \in D_v\}$$

  Typically, V is finite. In that case, we can use the simpler definition

$$\mathcal{S} = \prod_{v \in V} (D_v \cup \{\bot\})$$

  where $a(v) = \bot$ represents that a is undefined at v.

- An action assigns to a variable a concrete value of its domain. So

$$\mathcal{A} = \{(v,x) | v \in V, x \in D_v\}$$

- The transition model simply updates the variable assignment: $\mathcal{T}((v,x), a) = \{a'\}$ where $a'(v) = x$ and $a'(w) = a(w)$ if $w \neq v$.

  Note: alternatively, we could put $\mathcal{T}((v,x), a) = \emptyset$ if a already assigns a value to v.

- In the initial state, no variable is assigned: $\mathcal{I} = \{i\}$ where i is undefined everywhere.
- The goal states are the solutions, i.e., $\mathcal{G}$ is the set of all $a \in S$ such that
  - a is total
  - for all $u, v \in V$ with $u \neq v$, we have $(a(u), a(v)) \in C_{uv}$

Problem 5.3 (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.

   Solution: There are 2 solutions: a true, $b \in \{1, 2\}$, $c = 0$, $d = 0$

   ---

2. Give an inconsistent total variable assignment.

   Solution: Any total variable assignment that is not a solution, (see previous question) e.g., a false, $b = 0$, $c = 0$, $d = 0$

   ---

3. Give all consistent partial assignments α such that $\text{dom}(\alpha) \subseteq \{a, b\}$.

   Solution: We classify the possibly assignments a by their domain:

   - $\text{dom}(\alpha) = \emptyset$: 1 assignment, namely α undefined everywhere
   - $\text{dom}(\alpha) = \{a\}$: 2 assignments, namely $\alpha(a) \in D_a$, undefined elsewhere
   - $\text{dom}(\alpha) = \{b\}$: 4 assignments, namely $\alpha(b) \in D_b$, undefined elsewhere
   - $\text{dom}(\alpha) = \{a, b\}$: 7 assignments, namely

- 4 assignments with $\alpha(a) = false$, $\alpha(b) \in D_b$, undefined elsewhere
- 3 assignments with $\alpha(a) = true$, $\alpha(b) \in \{0, 1, 2\}$, undefined elsewhere

---

Problem 5.4 (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: $x \neq y$, $y > z$

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

Objective: understand acv

Solution: (x,y), (x,z), (y,x), (y,z), (z,x), (z,y)

---

2. Give all solutions that would remain if we added the constraint $x \neq z$ to Π

Objective: apply solution

Solution: Solutions for $\langle x, y, z \rangle$ are $\langle 0, 2, 1 \rangle$, $\langle 1, 2, 0 \rangle$, $\langle 2, 1, 0 \rangle$

---

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 domain

Objective: apply forward checking

Solution: $D_x = \{0, 2\}$, $D_y = \{1\}$, $D_z = \{0\}$

Problem 5.5 (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.