

DM877 - Constraint Programming

Exercise 1

Recall the definition of domain consistency and relaxed domain consistency such as bound(Z), bound(D) and range consistency. What is the consistency level of the following CSPs?

$$\mathcal{P} = \langle x_1 \in \{1, 3\}, x_2 \in \{1, 3\}, x_3 \in \{1, 3\}, x_4 \in \{1, 3\}, \mathcal{C} \equiv \text{alldifferent}(x_1, x_2, x_3, x_4) \rangle$$

$$\mathcal{P} = \langle x_1 \in \{1, 2, 3\}, x_2 \in \{2, 3\}, x_3 \in \{2, 3\}, x_4 \in \{1, 2, 3, 4\}, \mathcal{C} \equiv \text{alldifferent}(x_1, x_2, x_3, x_4) \rangle$$

$$\mathcal{P} = \langle x_1 \in \{1, 3\}, x_2 \in \{2\}, x_3 \in \{1, 2, 3\}, \mathcal{C} \equiv \text{alldifferent}(x_1, x_2, x_3) \rangle$$

$$\mathcal{P} = \langle x_1 \in \{1, 3\}, x_2 \in \{2\}, x_3 \in \{1, 3\}, \mathcal{C} \equiv \text{alldifferent}(x_1, x_2, x_3) \rangle$$

$$\mathcal{P} = \langle x_1 \in \{1, 3\}, x_2 \in \{1, 3\}, x_3 \in \{1, 3\}, \mathcal{C} \equiv \text{alldifferent}(x_1, x_2, x_3) \rangle$$

Exercise 2

Let $V = \{x, y, z\}$ and $D(x) = \{3, 4, 5\}, D(y) = \{0, 1, 2, 3\}, D(z) = \{1, 5\}$. Define one or more propagators implementing the constraint $x \leq y$. Compute the propagator on the constraint store defined. Is the propagator strong idempotent? Is it weak idempotent?

Exercise 3

Define a propagator for $x + y = d$ and then for $ax + by = d$. Finally, generalize it to the linear equality $\sum a_i x_i = d$.

- Is it necessary to perform several iterations of your propagator? Is it idempotent?
- What type of consistency it produces? (domain or bound consistency level?)
- Apply the propagator to the following example: $x = 3y + 5z, D(x) = \{2..7\}, D(y) = \{0..2\}, D(z) = \{-1..2\}$. Compare the result with your answer to the previous point.

Exercise 4 Usefulness of Weak Idempotency

Assume $V = \{x, y\}$ and $D(x) = [0..3], D(y) = [0..5]$ Consider the constraint $3x = 2y$ and the propagator p_{32}

$$p_{32}(D)(x) = D(x) \cap \{ \lceil (2 \min D(y))/3 \rceil, \dots, \lfloor (2 \max D(y))/3 \rfloor \}$$

$$p_{32}(D)(y) = D(y) \cap \{ \lceil (3 \min D(x))/2 \rceil, \dots, \lfloor (3 \max D(x))/2 \rfloor \}$$

Apply the propagator three times and state whether the propagator is strong idempotent. If it is not is there a constraint store for which it is weak idempotent?

Exercise 5

How would you implement a propagator for $\max(x, y) = z$?

Exercise 6

The `element` constraint models the following very common situation: we have to decide which among four goods to buy; for each good we have a different price; how to propagate the price while the variable is not yet assigned a good?

- Define a way to handle this situation without using the `element` constraint.
- Define a propagator for the `element(a, x, y)` constraint.
- Is it necessary to perform several iterations of your propagator? Is it idempotent?
- What type of consistency it produces? (domain or bound consistency level?)
- Run your propagator on the following example: $a = [4, 5, 7, 9]$, $D(x) = \{1, 2, 3\}$, $D(y) = \{2..8\}$.
- Can you devise a clever data structure that would speed up the propagation? Would it be worth storing the data structure for successive iterations?
- In `gecode` the `element` constraint is used also to implement a particular type of channeling, namely, $z = x_y$, where y and z are single variables and x is an array of variables. Write formally the propagators for x, y, z for the constraint `element(z, x, y)`.

Exercise 7 Steel Mill Slab Design

Model the following problem and report the model in written form.

Steel is produced by casting molten iron into slabs. A steel mill can produce a finite number, σ , of *slab sizes*. An order has two properties, a *colour* corresponding to the route required through the steel mill and a *weight*. Given d input orders, the problem is to assign the orders to slabs, the number and size of which are also to be determined, such that the total weight of steel produced is minimised. This assignment is subject to two further constraints:

Capacity constraints: The total weight of orders assigned to a slab cannot exceed the slab capacity.

Colour constraints: Each slab can contain at most p of k total colours (p is usually 2).

The colour constraints arise because it is expensive to cut up slabs in order to send them to different parts of the mill.

The above description is a simplification of a real industrial problem. For example, the problem may also include inventory matching, where surplus stock can be used to fulfil some of the orders.