Department of Mathematics and Computer Science
University of Southern Denmark, Odense

October 8, 2020

Marco Chiarandini

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?

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

Solution

The weakiest form is bound(**Z**) consistency. Accordingly we should ask whether the bounds of x_i, $i = 1,2,3,4$ have a bounded support. For $x_1 = 1$ we can have $x_2 = 2$, $x_3 = 3$ but no value would be left for x_4. Hence, $\mathcal{P}$ is not bound(**Z**) consistent. Since this is the weakiest form of those asked all the others are also not satisfied.

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

Solution

Same as above, it is not consistent in any of the forms.

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

Solution

It is bound(**Z**). For being bound(**D**) it must be that for all x_i, $i = 1,2,3$ its bounds belong to a support. This holds true for this case. For being range consistent all values of x_i $i = 1,2,3$ must belong to a bounded support. This is not true since $x_3 = 2$ has no support in x_2. Hence it is also not arc consistent.

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

Solution

It is bound(**Z**) and bound(**D**). Since 2 is not anymore in the domain of x_3 then it is also range consistent.

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

Solution

bound(**Z**): yes, bounded support hence we can reintroduce the value 2 in the support variables
bound(**D**): no.
range: yes, bounded support hence we can reintroduce the value 2 in the support variables

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?

Solution

A propagator p_{leq} for $x \leq y$ can be defined as follows:

$$\begin{aligned} p_{\leq}(D)(x) &= \{n \in D(x) | n \leq \max D(y)\} \\ p_{\leq}(D)(y) &= \{n \in D(y) | n \geq \min D(x)\} \\ p_{\leq}(D)(z) &= D(z) \end{aligned}$$

The propagator computes:

$$\begin{aligned} &p_{\leq}(D)(x) = \{n \in D(x) | n \leq 3\} \\ &p_{\leq}(D)(y) = \{n \in D(y) | n \geq 3\} \\ &p_{\leq}(D)(z) = D(z) \\ &\{D(x) = \{3\}, D(y) = \{3\}, D(z) = \{1, 5\}\} \end{aligned}$$

The propagator p_{leq} is weakly idempotent because the current space cannot be tighten by another application of the propagator. It is not strongly idempotent because if the domain of a variable changes then it might propagate again.

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.

Solution

Domain consistency:

$$D(x) \leftarrow \{n \in D(x) | \exists n_y \in D(y), n_z \in D(z) : n = n_y + n_z\}$$

It reaches fixpoint:

$$\begin{array}{l} \forall n_x \in D(x) \exists n_y \in D(y), n_z \in D(z) : n = n_y + n_z \\ \forall n_y \in D(y) \exists n_x \in D(x), n_z \in D(z) : n = n_x - n_z \\ \forall n_z \in D(z) \exists n_x \in D(x), n_y \in D(y) : n = n_x - n_y \end{array}$$

Bound consistency bound(Z):

$$D(x) \leftarrow \{n \in D(x) | n \leq \max D(y) + \max D(z) \wedge n \geq \min D(y) + \min D(z)\}$$

$$\begin{array}{l} \forall n_x \in \{\min D(x), \max D(x)\} \\ \exists n_y \in [\min D(y).. \max D(y)] \\ \exists n_z \in [\min D(z).. \max D(z)] \\ n_x = n_z + n_y \end{array}$$

$x + y = z$

$$\begin{array}{llll} x \in \{1, 2, 4, 8\} & y \in \{1, 2, 4\} & z \in \{1, 2, 3, 4, 6\} & \\ x \in \{2, 4, 8\} & y \in \{1, 3, 4\} & z \in \{1, 3, 4\} & \text{domain} \\ x \in \{2, 4, 8\} & y \in \{1, 3, 4\} & z \in \{1, 2, 3, 4\} & \text{bound(Z)} \end{array}$$

$$\begin{array}{l} D(x) \leftarrow \{n \in D(x) | \min D(z) - \max D(y) \leq n \leq \max D(z) - \min D(y)\} \\ D(y) \leftarrow \{n \in D(y) | \min D(z) - \max D(x) \leq n \leq \max D(z) - \min D(x)\} \\ D(z) \leftarrow \{n \in D(z) | \min D(x) + \min D(y) \leq n \leq \max D(x) + \max D(y)\} \end{array}$$

This is not idempotent

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}

$$\begin{array}{l} p_{32}(D)(x) = D(x) \cap \{\lceil (2 \min D(y))/3 \rceil, \ldots, \lfloor (2 \max D(y))/3 \rfloor\} \\ p_{32}(D)(y) = D(y) \cap \{\lceil (3 \min D(x))/2 \rceil, \ldots, \lfloor (3 \max D(x))/2 \rfloor\} \end{array}$$

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?

Solution

The propagator p_{32} is not idempotent. Consider Then $D' = p_{32}(D)$ is

$$D'(x) = [0..3] \cap [0..\lfloor 10/3 \rfloor] = [0..3]$$
$$D'(y) = [0..5] \cap [0..\lfloor 9/2 \rfloor] = [0..4]$$

Now $D'' = p_{32}(D')$ is

$$D''(x) = [0..3] \cap [0..\lfloor 8/3 \rfloor] = [0..2]$$
$$D''(y) = [0..4] \cap [0..\lfloor 9/2 \rfloor] = [0..4]$$

Hence $p_{32}(p_{32}(s)) = s'' \neq s' = p_{32}(D)$ and the propagator is not strong idempotent. Further $D''' = p_{32}(D'')$ is

$$D'''(x) = [0..2] \cap [0..\lfloor 8/3 \rfloor] = [0..2]$$
$$D'''(y) = [0..4] \cap [0..\lfloor 6/2 \rfloor] = [0..3]$$

The new bound for y is 3 and is obtained without rounding. In this case we are guaranteed that the proagator is at a fixedpoint. Knwoing that it is idempotent on D''' is useful since we can avoid runnig the propagator.

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.

Solution

Some input data:

Order	1	2	3	4	5
Size	5	8	3	4	6
Color	1	3	2	1	2

A solution:

Slab Capacity	12	10	7
Orders	1,3,4	2	5
Load	12	8	6
Loss	0	2	1

```
IntVarArray x[Orders](Slabs); // the slab in which the order is placed
IntVarArray l[Slabs](0..maxCap); // the load in each slab

post(multiknapsack(x,weight,l));
forall (s in Slabs)
  post(sum(c in Colors) (or(o in colorOrders[c])(x[o] == s)) <= 2);

cost(sum(s in Slabs) loss[l[s]])

branch(x)
```