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

October 20, 2020
Marco Chiarandini

DM877 - Constraint Programming

Obligatory Assignment 2, Autumn 2020

~~**Deadline: Wednesday, October 21, 2020 at noon**.~~
Deadline: Sunday, October 25, 2020 at noon.

Preliminary Remarks

- This is the *second* of two preparatory assignments on constraint programming. The assignment is compulsory with pass/fail evaluation.
- The aim of this assignment is to work with pen and paper on the concepts and definitions of local consistency and propagation discussed in class.
- You are allowed to discuss with your peers but the final submission must be individual.
- The submission is electronic in BlackBoard under SDU Assignment.
- You have to hand in one file in PDF format containing your responses to the tasks below. You can handwrite your answers and scan them. Alternatively, source code for the figures and tables is linked in the captions of the figures.

Tasks

The following assignment was posed in a written exam by Prof. Pierre Flener for the course on Constraint Programming at Uppsala University.
Consider the following CSP and its labelled constraints:

$$\mathcal{P} := \langle X := \{v, w, x\}; \mathcal{DE} := \{D(v) = D(w) = D(x) = \{1, \dots, 9\}\}; \mathcal{C} := \{$$

$$\textsc{Element}([2,1,8,2,1,0], v, x) \tag{c}$$

$$v + x \leq 7 \tag{d}$$

$$w \cdot w \leq 5 \tag{e}$$

$$\textsc{Distinct}(\{v, w, x\})\}\rangle \tag{f}$$

We will look at the execution of the constraint propagation for this problem. The algorithm uses propagator conditions (events), status messages and keeps track of variables whose domains have been modified.
The following propagation choices are imposed:

- Use **idempotent** propagators achieving **bounds**(**Z**) consistency for the arithmetic constraints and **domain** consistency for the other constraints.
- Post the constraints in the textual order in which they appear above.
- Handle the decision variables in the textual order in which they appear in the CSP above.
- Use a **first-in first-out queue** (FIFO) for implementing the set Q of propagators that are not known to be at fixpoint. (Note that Q is **not** a multi-set.)

Upon denoting the propagator of constraint C by $p_C(\cdot)$, address the following tasks:

1. Formalize the propagators in a similar way as seen in class:

$$p_{\textsc{Element}}(x, \mathcal{DE}) = p_{\textsc{Element}}(D)(x) = \{ \qquad\qquad \}$$

...

2. In the Constraint-Propag fixpoint algorithm seen in the course (slide 15, lecture "Rules Iteration") one needs to know when propagators are to be executed. For each constraint C, give (without proof) the set PropConds$(\cdot)$ of conditions that should trigger the scheduling of the propagators of $C \in \mathcal{C}$, as a strictly tighter CSP might then be obtained. Each propagator condition is of the form 'Any(α)', 'Fixed(α)', or 'Min/Max(α)', where α is a decision variable of C. Write '(none)' rather than nothing, where appropriate. Write the most general condition (eg, Any() is more general than Min/Max()).

   PropConds(c) = { }

   PropConds(d) = { }

   PropConds(e) = { }

   PropConds(f) = { }

3. Using a procedure like the Constraint-Propag fixpoint algorithm, perform the pre-search propagation to compute the fixpoint of the root of the search tree. Fill in the data in a table like Table 1 for the initialisation (in the first row) and every pre-search step of your Constraint-Propag. (The array argument of the Element constraint is **indexed from 1**, not from 0.) Write the status message of the propagators after their execution as precisely as possible, the options being 'AtFixpt', 'Unknwon' (short for "not known to be at fixpoint"), 'Subsumed' and 'Failed'. In addition, write the modification event that occurs. The modification event is of the form 'None(α)', 'Failed(α)', 'Fixed(α)', 'Min/Max(α)', or 'Any(α)', where α is a decision variable of the propagated constraint. Write '(none)' rather than nothing, where appropriate. Finally, write the dependant propagators that are triggered by the events occurred and how the new queue of propagators will look like.

4. Indicate what changes in your answers to the previous sub-tasks when instead achieving **domain** consistency for **all** the constraints. Why? (Note that you are **not** asked to fill in another table.)

$$\textsc{Element}\left(\begin{array}{|c|c|c|c|c|c|} \multicolumn{1}{c}{\scriptstyle 1} & \multicolumn{1}{c}{\scriptstyle 2} & \multicolumn{1}{c}{\scriptstyle 3} & \multicolumn{1}{c}{\scriptstyle 4} & \multicolumn{1}{c}{\scriptstyle 5} & \multicolumn{1}{c}{\scriptstyle 6} \\ \hline 2 & 1 & 8 & 2 & 1 & 0 \\ \hline \end{array}, v, x\right) \text{ (c)}, \quad v + x \leq 7 \text{ (d)}, \quad w \cdot w \leq 5 \text{ (e)}, \quad \textsc{Distinct}(\{v, w, x\}) \text{ (f)}$$

Chosen prop.	Resulting $\mathcal{DE}$	Status message	Modification events	Dependent prop.s DEPPROPS	Non-subsumed propagators R	FIFO queue Q of propagators
(none)	$v \mapsto \{\}, w \mapsto \{\}, x \mapsto \{\}$	(none)	(none)	(none)	$\{\}$	[]
	$v \mapsto \{\quad\}, w \mapsto \{\quad\}, x \mapsto \{\quad\}$			$\{\quad\}$	$\{\quad\}$	$[\quad]$
	$v \mapsto \{\quad\}, w \mapsto \{\quad\}, x \mapsto \{\quad\}$			$\{\quad\}$	$\{\quad\}$	$[\quad]$
	$v \mapsto \{\quad\}, w \mapsto \{\quad\}, x \mapsto \{\quad\}$			$\{\quad\}$	$\{\quad\}$	$[\quad]$
	$v \mapsto \{\quad\}, w \mapsto \{\quad\}, x \mapsto \{\quad\}$			$\{\quad\}$	$\{\quad\}$	$[\quad]$
	$v \mapsto \{\quad\}, w \mapsto \{\quad\}, x \mapsto \{\quad\}$			$\{\quad\}$	$\{\quad\}$	$[\quad]$
	$v \mapsto \{\quad\}, w \mapsto \{\quad\}, x \mapsto \{\quad\}$			$\{\quad\}$	$\{\quad\}$	$[\quad]$
	$v \mapsto \{\quad\}, w \mapsto \{\quad\}, x \mapsto \{\quad\}$			$\{\quad\}$	$\{\quad\}$	$[\quad]$
	$v \mapsto \{\quad\}, w \mapsto \{\quad\}, x \mapsto \{\quad\}$			$\{\quad\}$	$\{\quad\}$	$[\quad]$
	$v \mapsto \{\quad\}, w \mapsto \{\quad\}, x \mapsto \{\quad\}$			$\{\quad\}$	$\{\quad\}$	$[\quad]$
	$v \mapsto \{\quad\}, w \mapsto \{\quad\}, x \mapsto \{\quad\}$			$\{\quad\}$	$\{\quad\}$	$[\quad]$
	$v \mapsto \{\quad\}, w \mapsto \{\quad\}, x \mapsto \{\quad\}$			$\{\quad\}$	$\{\quad\}$	$[\quad]$
	$v \mapsto \{\quad\}, w \mapsto \{\quad\}, x \mapsto \{\quad\}$			$\{\quad\}$	$\{\quad\}$	$[\quad]$
	$v \mapsto \{\quad\}, w \mapsto \{\quad\}, x \mapsto \{\quad\}$			$\{\quad\}$	$\{\quad\}$	$[\quad]$
	$v \mapsto \{\quad\}, w \mapsto \{\quad\}, x \mapsto \{\quad\}$			$\{\quad\}$	$\{\quad\}$	$[\quad]$
	$v \mapsto \{\quad\}, w \mapsto \{\quad\}, x \mapsto \{\quad\}$			$\{\quad\}$	$\{\quad\}$	$[\quad]$
	$v \mapsto \{\quad\}, w \mapsto \{\quad\}, x \mapsto \{\quad\}$			$\{\quad\}$	$\{\quad\}$	$[\quad]$
	$v \mapsto \{\quad\}, w \mapsto \{\quad\}, x \mapsto \{\quad\}$			$\{\quad\}$	$\{\quad\}$	$[\quad]$
	$v \mapsto \{\quad\}, w \mapsto \{\quad\}, x \mapsto \{\quad\}$			$\{\quad\}$	$\{\quad\}$	$[\quad]$
	$v \mapsto \{\quad\}, w \mapsto \{\quad\}, x \mapsto \{\quad\}$			$\{\quad\}$	$\{\quad\}$	$[\quad]$
	$v \mapsto \{\quad\}, w \mapsto \{\quad\}, x \mapsto \{\quad\}$			$\{\quad\}$	$\{\quad\}$	$[\quad]$
	$v \mapsto \{\quad\}, w \mapsto \{\quad\}, x \mapsto \{\quad\}$			$\{\quad\}$	$\{\quad\}$	$[\quad]$
	$v \mapsto \{\quad\}, w \mapsto \{\quad\}, x \mapsto \{\quad\}$			$\{\quad\}$	$\{\quad\}$	$[\quad]$
	$v \mapsto \{\quad\}, w \mapsto \{\quad\}, x \mapsto \{\quad\}$			$\{\quad\}$	$\{\quad\}$	$[\quad]$
	$v \mapsto \{\quad\}, w \mapsto \{\quad\}, x \mapsto \{\quad\}$			$\{\quad\}$	$\{\quad\}$	$[\quad]$

Table 1: Latex source code for this table.

5. If the pre-search propagation of sub-task 3. has not solved the problem, then draw (**below**) the search tree with **all** the solutions, with pairs of non-subsumed propagator sets and domain extensions as nodes, and decisions (which are constraints) as labelled arcs. The previous choices in this task and the following branching heuristics are imposed:

   - Use **largest-minimum** variable selection
     (`int_search( [v,w,x], largest, <assignmentannotation>)` in Minizinc).
   - Use **bottom-up** value selection
     (`int_search( [v,w,x], <varchoiceannnotation>, indomain_min)` in Minizinc).

   Continue to **use the table on the previous page** when propagating a branching decision or a constraint, and mark there the starting row of each call to Constraint-Propag.

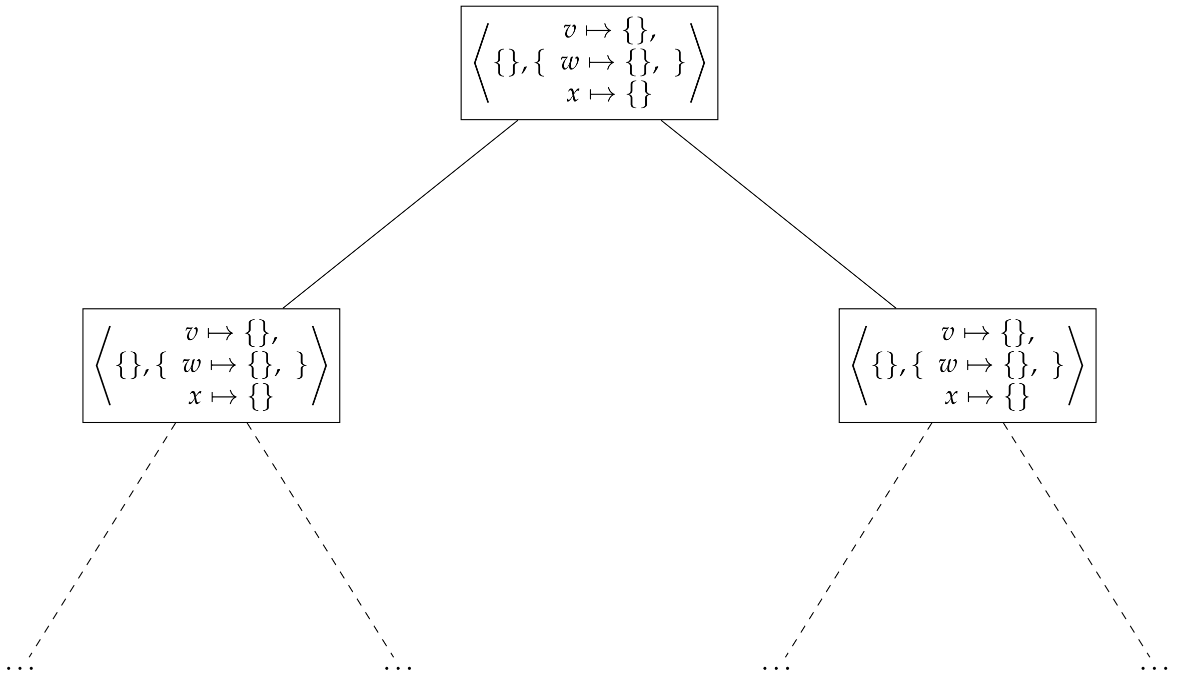
$\left\langle \{\}, \{ \begin{array}{c} v \mapsto \{\}, \\ w \mapsto \{\}, \\ x \mapsto \{\} \end{array} \} \right\rangle$
$\left\langle \{\}, \{ \begin{array}{c} v \mapsto \{\}, \\ w \mapsto \{\}, \\ x \mapsto \{\} \end{array} \} \right\rangle$
$\left\langle \{\}, \{ \begin{array}{c} v \mapsto \{\}, \\ w \mapsto \{\}, \\ x \mapsto \{\} \end{array} \} \right\rangle$
...
...
...
...

Figure 1: Latex source code for this figure.