Assignment 2 **Kim Hammar**
ID2204 **Mallu**
Constraint Programming Due Date: 30 April 2017

Propagator Rewriting

For a correct rewrite of a propagator p to another propagator p' on a store s, the following must hold:

$$p(s') = p'(s') \quad \forall s' \text{ where } s' \leq s$$

I.e a propagator can be rewritten to another (simpler) propagator if the propagators behave the same on all stronger stores than the current store. Typical usage of propagator rewriting is to simplify a propagator when a store s is reached, where complex propagation is no longer necessary due to the reduced variable domains.

To rewrite a single propagator p to a set of Q simpler propagators it is necessary for correctness that the following holds:

$$p(s') = \bigcap q(s') \quad \forall q \in Q \quad \forall s' \text{ where } s' \leq s$$

A modification to the propagation algorithm to support propagator rewriting might be to allow propagators to send an extra status message: $\{rewrite, Q\}$ and the algorithm can then replace p with the propagators in Q. The correctness of the relation between Q and p would be up to the propagator itself. I.e you add the following line to the algorithm:

if `ms = {rewrite, Q}` **then** $EP := (EP - \{ep\}) \cup Q$

An Addition Propagator

Propagator Definition

Propagator $p_+ \in c \equiv x + y = z$.

$$p_+(s) = \begin{cases} x \mapsto \{n \in s(x) | \min s(z) - \max s(y) \leq n \leq \max s(z) - \min s(y)\} \\ y \mapsto \{n \in s(y) | \min s(z) - \max s(x) \leq n \leq \max s(z) - \min s(x)\} \\ z \mapsto \{n \in s(z) | \min s(x) + \min s(y) \leq n \leq \max s(x) + \max s(y)\} \end{cases}$$

Idempotence

Theorem 0.1. *The propagator p_+ is not idempotent.*

Proof. If p_+ is idempotent then $p_+(p_+(s)) = p_+(s) \quad \forall s$, Counterexample:

$$s = \{x \mapsto \{0, 4\}, y \mapsto \{0, 4\}, z \mapsto \{0, 1, 2\}\}$$

$$p_+(s) = s' = \{x \mapsto \{0\}, y \mapsto \{0\}, z \mapsto \{0,1,2\}\}$$
$$p_+(s') = s'' = \{x \mapsto \{0\}, y \mapsto \{0\}, z \mapsto \{0\}\}$$

$$\therefore s' \neq s'' \quad Q.E.D$$

$\square$

Subsumption

Detect subumption:

$$\begin{gathered} p_+(s) = \textbf{let } s' = p_+(s), \quad s'' = p_+(s') \\ \textbf{if } |s'(x)| = 1 \wedge |s'(y)| = 1 \wedge |s'(z)| = 1 \wedge s'(x) + s'(y) = s'(z) \\ \textbf{then } \langle subsumed, s' \rangle \\ \textbf{else if } s'' = s' \quad \langle fix, s' \rangle \\ \textbf{else if } s'' \neq s' \quad \langle nofix, s' \rangle \end{gathered}$$

Rewriting

Rewrite into a simpler propagator:

Assume for instance that propagation have computed $s(x) = 1$ then p_+ can be rewritten to the following simpler (more efficient) propagator:

$$p_+(s) = \begin{cases} y \mapsto \{n \in s(y) | \min s(z) - 1 \leq n \leq \max s(z) - 1\} \\ z \mapsto \{n \in s(z) | \min s(y) + 1 \leq n \leq \max s(y) + 1\} \end{cases}$$

Generalizing to Arbitrary Equations

$$p_=(s) = \Big\{ \forall i \in \{1, \ldots, n\}, x_i \mapsto \{n \in s(x_i) | s(x_i) = \tfrac{c}{a_i}\}$$

Theorem 0.2. *$p_=$ is idempotent*

Proof. This follows from contracting property of propagator and that a_i, c are constants. After running the propagator once either we have a failed store or $x_i = \frac{c}{a_i}$, then there are no more values to remove and propagator is not allowed to add any values. $\square$

Propagating on paper

Propagator $p \in$ constraint $c \equiv x < y \wedge y < x$.

$$p(s) = \begin{cases} x \mapsto \{n \in s(x) | n \max s(y)\} \\ y \mapsto \{n \in s(y) | n \max s(x)\} \end{cases}$$

$U = \{0, 1, 2, 3\}$

Initial store $s_1 = \{x \mapsto 0, 1, 2, 3, y \mapsto 0, 1, 2, 3\}$

$p(s_1) = \{\texttt{nofix}, s_2\} = \{\texttt{nofix}, \{x \mapsto 0, 1, 2, y \mapsto 0, 1, 2\}\}$

$p(s_2) = \{\texttt{nofix}, s_3\} = \{\texttt{nofix}, \{x \mapsto 0, 1, y \mapsto 0, 1\}\}$

$p(s_3) = \{\texttt{nofix}, s_4\} = \{\texttt{nofix}, \{x \mapsto 0, y \mapsto 0\}\}$

$p(s_4) = \{\texttt{fix}, s_5\} = \{\texttt{fix}, \{x \mapsto \emptyset, y \mapsto \emptyset\}\}$