## 5.178 in_interval

DESCRIPTION LINKS GRAPH AUTOMATON

**Origin** Domain definition.

**Constraint** in_interval(VAR, LOW, UP)

**Synonyms** dom, in.

**Arguments**

| | | |
|---|---|---|
| VAR | : | dvar |
| LOW | : | int |
| UP | : | int |

**Restriction** LOW ≤ UP

**Purpose** Enforce the domain variable VAR to take a value within the interval [LOW, UP].

**Example** $(3, 2, 5)$

The in_interval constraint holds since its first argument VAR $= 3$ is greater than or equal to its second argument LOW $= 2$ and less than or equal to its third argument UP $= 5$.

**Typical**

LOW < UP
VAR > LOW
VAR < UP

**Symmetries**

- LOW can be decreased.
- UP can be increased.
- An occurrence of a value of VAR can be replaced by any other value in [LOW, UP].
- One and the same constant can be added to VAR, LOW and UP.

**Remark** Entailment occurs immediately after posting this constraint.

The in_interval constraint is referenced under the name dom in **Gecode**.

**Systems** member in **Choco**, dom in **Gecode**, in in **JaCoP**, in in **SICStus**.

**See also** **common keyword:** domain, in *(domain definition)*.

**generalisation:** in_interval_reified *(reified version)*, in_intervals *(single interval replaced by a set of intervals)*, in_set *(interval replaced by set variable)*.

**Keywords** **characteristic of a constraint:** automaton, automaton without counters, reified automaton constraint, derived collection.

**constraint arguments:** unary constraint.

**constraint network structure:** Berge-acyclic constraint network.

**constraint type:** value constraint.

**filtering:** arc-consistency.

**modelling:** interval, domain definition.

**Derived Collections**

$\texttt{col}(\texttt{VARIABLE}-\texttt{collection}(\texttt{var}-\texttt{dvar}), [\texttt{item}(\texttt{var} - \texttt{VAR})])$

$\texttt{col}\left(\begin{array}{l}\texttt{INTERVAL}-\texttt{collection}(\texttt{low}-\texttt{int}, \texttt{up}-\texttt{int}),\\ [\texttt{item}(\texttt{low} - \texttt{LOW}, \texttt{up} - \texttt{UP})]\end{array}\right)$

**Arc input(s)** VARIABLE INTERVAL

**Arc generator** $PRODUCT \mapsto \texttt{collection}(\texttt{variable}, \texttt{interval})$

**Arc arity** 2

**Arc constraint(s)**
- $\texttt{variable.var} \geq \texttt{interval.low}$
- $\texttt{variable.var} \leq \texttt{interval.up}$

**Graph property(ies)** **NARC**$= 1$

**Graph model** Parts (A) and (B) of Figure 5.418 respectively show the initial and final graph associated with the **Example** slot. Since we use the **NARC** graph property, the unique arc of the final graph is stressed in bold.

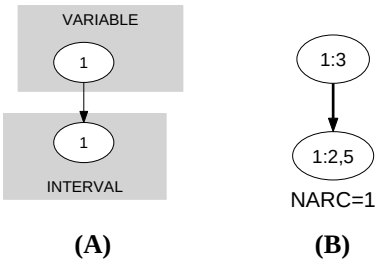


Figure 5.418: Initial and final graph of the in_interval constraint

**Automaton**

Figure 5.419 depicts the automaton associated with the `in_interval` constraint. We have a single 0-1 signature variable $S$ as well as the following signature constraint: $\texttt{VAR} \geq \texttt{LOW} \wedge \texttt{VAR} \leq \texttt{UP} \Leftrightarrow S$.

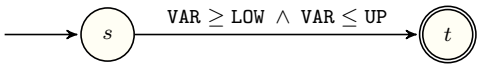


Figure 5.419: Automaton of the `in_interval` constraint

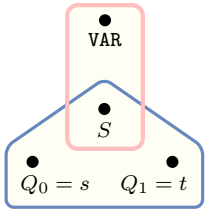


Figure 5.420: Hypergraph of the reformulation corresponding to the automaton of the `in_interval` constraint