5.177 in

DESCRIPTION LINKS GRAPH AUTOMATON

Origin Domain definition.

Constraint in(VAR, VALUES)

Synonyms dom, in_set, member.

Arguments

VAR	:	dvar
VALUES	:	collection(val−int)

Restrictions

$|\texttt{VALUES}| > 0$
required(VALUES, val)
distinct(VALUES, val)

Purpose Enforce the domain variable VAR to take a value within the values described by the VALUES collection.

Example $(3, \langle 1, 3 \rangle)$

The in constraint holds since its first argument $\texttt{VAR} = 3$ occurs within the collection of values $\texttt{VALUES} = \langle 1, 3 \rangle$.

Typical $|\texttt{VALUES}| > 1$

Symmetries

- Items of VALUES are permutable.
- VAR can be set to any value of VALUES.val.
- One and the same constant can be added to VAR as well as to the val attribute of all items of VALUES.

Arg. properties

Extensible wrt. VALUES.

Remark Entailment occurs immediately after posting this constraint.

The in constraint is called dom in **Gecode** (http://www.gecode.org/), and member in **MiniZinc** (http://www.minizinc.org/). In **MiniZinc** the val attribute is not necessarily fixed, i.e. it can be a domain variable.

Systems member in **Choco**, rel in **Gecode**, dom in **Gecode**, in in **JaCoP**, member in **MiniZinc**, in in **SICStus**, in_set in **SICStus**.

Used in among, cardinality_atmost_partition, group, group_skip_isolated_item, in_same_partition, open_among.

See also

common keyword: `domain` *(domain definition)*, `in_interval`, `in_same_partition`, `in_set` *(value constraint)*.

implied by: `maximum`, `minimum`.

implies: `between_min_max`.

negation: `not_in`.

Keywords

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

constraint arguments: unary constraint.

constraint network structure: centered cyclic(1) constraint network(1).

constraint type: value constraint.

filtering: arc-consistency.

modelling: included, domain definition.

Derived Collection	col(VARIABLES−collection(var−dvar), [item(var − VAR)])
Arc input(s)	VARIABLES VALUES
Arc generator	$PRODUCT \mapsto$ collection(variables, values)
Arc arity	2
Arc constraint(s)	variables.var = values.val
Graph property(ies)	**NARC**= 1

Graph model Parts (A) and (B) of Figure 5.415 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.

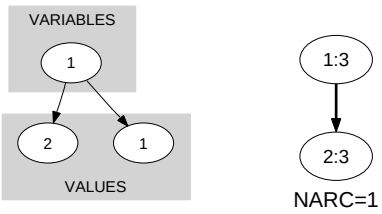

Figure 5.415: Initial and final graph of the in constraint

Signature Since all the val attributes of the VALUES collection are distinct and because of the arc constraint variables.var = values.val the final graph contains at most one arc. Therefore we can rewrite **NARC** = 1 to **NARC** ≥ 1 and simplify $\overline{\mathbf{NARC}}$ to $\overline{\mathbf{NARC}}$.

Automaton

Figure 5.416 depicts the automaton associated with the `in` constraint. Let $\texttt{VAL}_i$ be the `val` attribute of the i^{th} item of the `VALUES` collection. To each pair $(\texttt{VAR}, \texttt{VAL}_i)$ corresponds a 0-1 signature variable S_i as well as the following signature constraint: $\texttt{VAR} = \texttt{VAL}_i \Leftrightarrow S_i$.

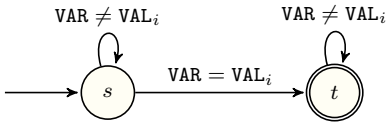

Figure 5.416: Automaton of the `in` constraint

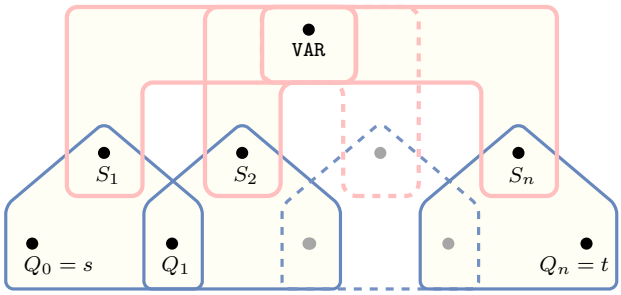

Figure 5.417: Hypergraph of the reformulation corresponding to the automaton of the `in` constraint