5.25 among_interval

DESCRIPTION LINKS GRAPH AUTOMATON

Origin Derived from among.

Constraint among_interval(NVAR, VARIABLES, LOW, UP)

Arguments

NVAR	:	dvar
VARIABLES	:	collection(var−dvar)
LOW	:	int
UP	:	int

Restrictions

$\texttt{NVAR} \geq 0$
$\texttt{NVAR} \leq |\texttt{VARIABLES}|$
required(VARIABLES, var)
$\texttt{LOW} \leq \texttt{UP}$

Purpose NVAR is the number of variables of the collection VARIABLES taking a value that is located within interval [LOW, UP].

Example $(3, \langle 4, 5, 8, 4, 1 \rangle, 3, 5)$

The among_interval constraint holds since we have 3 values, namely $4, 5$ and 4 that are situated within interval $[3, 5]$.

Typical

$\texttt{NVAR} > 0$
$\texttt{NVAR} < |\texttt{VARIABLES}|$
$|\texttt{VARIABLES}| > 1$
$\texttt{LOW} < \texttt{UP}$
$\texttt{LOW} \leq$maxval(VARIABLES.var)
$\texttt{UP} \geq$minval(VARIABLES.var)

Symmetries

- Items of VARIABLES are permutable.
- An occurrence of a value of VARIABLES.var that belongs to [LOW, UP] (resp. does not belong to [LOW, UP]) can be replaced by any other value in [LOW, UP]) (resp. not in [LOW, UP]).

Arg. properties

- Functional dependency: NVAR determined by VARIABLES, LOW and UP.
- Contractible wrt. VARIABLES when $\texttt{NVAR} = 0$.
- Contractible wrt. VARIABLES when $\texttt{NVAR} = |\texttt{VARIABLES}|$.
- Aggregate: NVAR(+), VARIABLES(union), LOW(id), UP(id).

Remark By giving explicitly all values of the interval [LOW, UP] the among_interval constraint can be modelled with the among constraint. However when $\texttt{LOW} - \texttt{UP} + 1$ is a large quantity the among_interval constraint provides a more compact form.

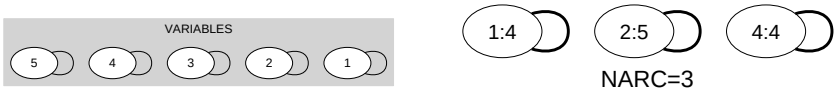
See also **generalisation:** among (`variable` *in* `interval` *replaced by* `variable` ∈ `values`).

Keywords **characteristic of a constraint:** automaton, automaton with counters.
constraint arguments: pure functional dependency.
constraint network structure: alpha-acyclic constraint network(2).
constraint type: value constraint, counting constraint.
filtering: arc-consistency.
modelling: interval, functional dependency.

Arc input(s)	VARIABLES
Arc generator	$SELF \mapsto$ collection(variables)
Arc arity	1
Arc constraint(s)	• LOW $\leq$ variables.var • variables.var $\leq$ UP
Graph property(ies)	NARC= NVAR

Graph model The arc constraint corresponds to a unary constraint. For this reason we employ the *SELF* arc generator in order to produce a graph with a single loop on each vertex.

Parts (A) and (B) of Figure 5.58 respectively show the initial and final graph associated with the **Example** slot. Since we use the NARC graph property, the loops of the final graph are stressed in bold.



(A) **(B)**

Figure 5.58: Initial and final graph of the among_interval constraint

Automaton

Figure 5.59 depicts the automaton associated with the among_interval constraint. To each variable $\texttt{VAR}_i$ of the collection VARIABLES corresponds a 0-1 signature variable S_i. The following signature constraint links $\texttt{VAR}_i$ and S_i: $\texttt{LOW} \leq \texttt{VAR}_i \wedge \texttt{VAR}_i \leq \texttt{UP} \Leftrightarrow S_i$. The automaton counts the number of variables of the VARIABLES collection that take their value in [LOW, UP] and finally assigns this number to NVAR.

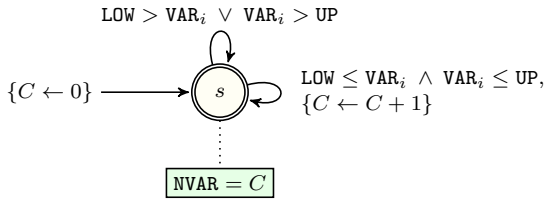


Figure 5.59: Automaton of the among_interval constraint

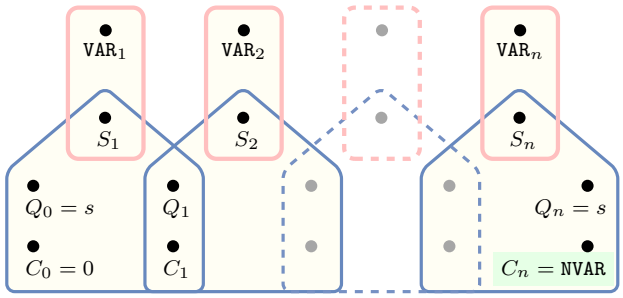


Figure 5.60: Hypergraph of the reformulation corresponding to the automaton (with one counter) of the among_interval constraint: since all states variables $Q_0, Q_1, \ldots, Q_n$ are fixed to the unique state s of the automaton, the transitions constraints share only the counter variable C and the constraint network is Berge-acyclic