

5.18 among_interval

	DESCRIPTION	LINKS	GRAPH	AUTOMATON
Origin	Derived from among .			
Constraint	<code>among_interval(NVAR, VARIABLES, LOW, UP)</code>			
Arguments	<pre>NVAR : dvar VARIABLES : collection(var-dvar) LOW : int UP : int</pre>			
Restrictions	<pre>NVAR ≥ 0 NVAR ≤ VARIABLES required(VARIABLES, var) LOW ≤ UP</pre>			
Purpose	<div style="border: 1px solid pink; padding: 5px;"> <p>NVAR is the number of variables of the collection VARIABLES taking a value that is located within interval [LOW, UP].</p> </div>			
Example	<div style="border: 1px solid blue; padding: 5px; display: inline-block;"> $(3, \langle 4, 5, 8, 4, 1 \rangle, 3, 5)$ </div> <p>The <code>among_interval</code> constraint holds since we have 3 values, namely 4, 5 and 4 that are situated within interval [3, 5].</p>			
Typical	<pre>NVAR > 0 NVAR < VARIABLES VARIABLES > 1 LOW < UP</pre>			
Symmetries	<ul style="list-style-type: none"> • 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]). 			
Remark	By giving explicitly all values of the interval [LOW, UP] the <code>among_interval</code> constraint can be modelled with the among constraint. However when $LOW - UP + 1$ is a large quantity the <code>among_interval</code> constraint provides a more compact form.			
See also	generalisation : <code>among</code> (variable in interval replaced by variable \in values).			
Keywords	<p>characteristic of a constraint: automaton, automaton with counters.</p> <p>constraint network structure: alpha-acyclic constraint network(2).</p> <p>constraint type: value constraint, counting constraint.</p> <p>filtering: arc-consistency.</p> <p>modelling: interval.</p>			

Arc input(s)	VARIABLES
Arc generator	$SELF \mapsto \text{collection}(\text{variables})$
Arc arity	1
Arc constraint(s)	<ul style="list-style-type: none"> • $LOW \leq \text{variables.var}$ • $\text{variables.var} \leq UP$
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 5.30 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.

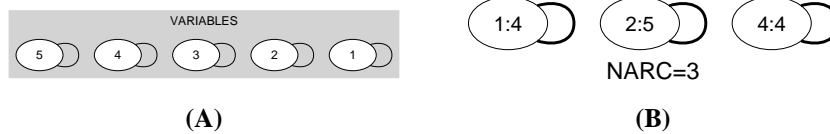


Figure 5.30: Initial and final graph of the `among_interval` constraint

Automaton

Figure 5.31 depicts the automaton associated with the `among_interval` constraint. To each variable VAR_i of the collection `VARIABLES` corresponds a 0-1 signature variable S_i . The following signature constraint links VAR_i and S_i : $LOW \leq VAR_i \wedge VAR_i \leq UP \Leftrightarrow S_i$. 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`.

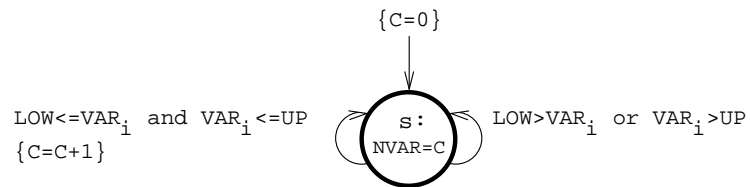


Figure 5.31: Automaton of the `among_interval` constraint

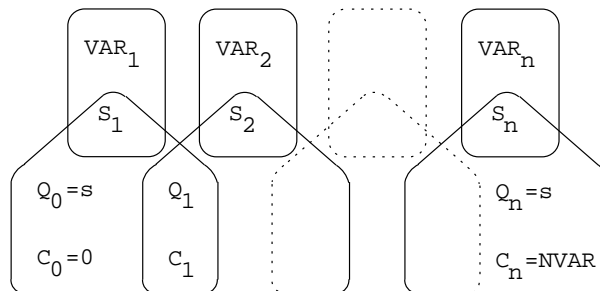


Figure 5.32: Hypergraph of the reformulation corresponding to the automaton of the `among_interval` constraint

20030820

455