

5.153 in_interval

	DESCRIPTION	LINKS	GRAPH	AUTOMATON
Origin	Domain definition.			
Constraint	<code>in_interval(VAR, LOW, UP)</code>			
Synonyms	<code>dom</code> , <code>in</code> .			
Arguments	VAR : <code>dvar</code> LOW : <code>int</code> UP : <code>int</code>			
Restriction	$LOW \leq UP$			
Purpose	Enforce the domain variable VAR to take a value within the interval [LOW, UP].			
Example	(3, 2, 5)			
	The <code>in_interval</code> constraint holds since its first argument <code>VAR = 3</code> is greater than or equal to its second argument <code>LOW = 2</code> and less than or equal to its third argument <code>UP = 5</code> .			
Typical	$LOW < UP$ $VAR > LOW$ $VAR < UP$			
Symmetries	<ul style="list-style-type: none"> • 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 <code>in_interval</code> constraint is referenced under the name <code>dom</code> in Gecode .			
Systems	<code>member</code> in Choco , <code>in</code> in JaCoP , <code>in</code> in SICStus .			
See also	common keyword: <code>domain</code> , <code>in</code> (<i>domain definition</i>). generalisation: <code>in_interval_reified</code> (<i>reified version</i>), <code>in_intervals</code> (<i>single interval replaced by a set of intervals</i>), <code>in_set</code> (<i>interval replaced by set variable</i>).			
Keywords	characteristic of a constraint: <code>automaton</code> , <code>automaton without counters</code> , <code>reified automaton constraint</code> , <code>derived collection</code> . constraint arguments: unary constraint.			

20060317

1017

constraint network structure: Berge-acyclic constraint network.

constraint type: value constraint.

filtering: arc-consistency.

modelling: interval, domain definition.

Derived Collections

```

col(VARIABLE-collection(var-dvar), [item(var - VAR)])
col( INTERVAL-collection(low-int, up-int),
    [item(low - LOW, up - UP)] )

```

Arc input(s)

VARIABLE INTERVAL

Arc generator*PRODUCT* \mapsto collection(variable, interval)**Arc arity**

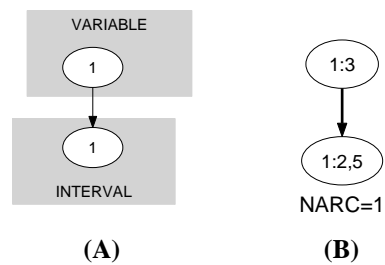
2

Arc constraint(s)

- variable.var \geq interval.low
- variable.var \leq interval.up

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

Parts (A) and (B) of Figure 5.310 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 5.310: Initial and final graph of the *in_interval* constraint

Automaton

Figure 5.311 depicts the automaton associated with the `in_interval` constraint. We have one single 0-1 signature variable S as well as the following signature constraint: $\text{VAR} \geq \text{LOW} \wedge \text{VAR} \leq \text{UP} \Leftrightarrow S$.

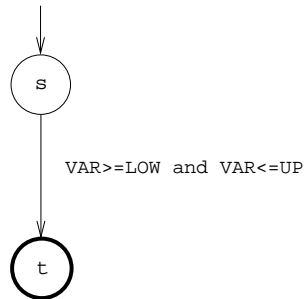


Figure 5.311: Automaton of the `in_interval` constraint

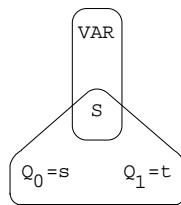


Figure 5.312: Hypergraph of the reformulation corresponding to the automaton of the `in_interval` constraint