

**5.283 not\_in**

	DESCRIPTION	LINKS	GRAPH	AUTOMATON
<b>Origin</b>	Derived from <a href="#">in</a> .			
<b>Constraint</b>	<code>not_in(VAR, VALUES)</code>			
<b>Arguments</b>	<pre>VAR      : dvar VALUES   : collection(val-int)</pre>			
<b>Restrictions</b>	<pre>required(VALUES, val) distinct(VALUES, val)</pre>			
<b>Purpose</b>	Enforce VAR to be assigned a value different from the values of the VALUES collection.			
<b>Example</b>	<code>(2, &lt;1, 3&gt;)</code>			
	The constraint <code>not_in</code> holds since the value of its first argument <code>VAR = 2</code> does not occur within the collection <code>&lt;1, 3&gt;</code> .			
<b>Typical</b>	<code> VALUES  &gt; 1</code>			
<b>Symmetries</b>	<ul style="list-style-type: none"> <li>Items of VALUES are <a href="#">permutable</a>.</li> <li>One and the same constant can be <a href="#">added</a> to VAR as well as to the <code>val</code> attribute of all items of VALUES.</li> </ul>			
<b>Arg. properties</b>	<a href="#">Contractible</a> wrt. VALUES.			
<b>Remark</b>	<a href="#">Entailment</a> occurs immediately after posting this constraint and removing all values in VALUES from VAR.			
<b>Systems</b>	<a href="#">notMember</a> in <a href="#">Choco</a> , <a href="#">rel</a> in <a href="#">Gecode</a> .			
<b>Used in</b>	<a href="#">group</a> .			
<b>See also</b>	<a href="#">negation</a> : <a href="#">in</a> .			
<b>Keywords</b>	<p><b>characteristic of a constraint:</b> <a href="#">disequality</a>, <a href="#">automaton</a>, <a href="#">automaton without counters</a>, <a href="#">reified automaton constraint</a>, <a href="#">derived collection</a>.</p> <p><b>constraint arguments:</b> <a href="#">unary constraint</a>.</p> <p><b>constraint network structure:</b> <a href="#">centered cyclic(1)</a> <a href="#">constraint network(1)</a>.</p> <p><b>constraint type:</b> <a href="#">value constraint</a>.</p> <p><b>filtering:</b> <a href="#">arc-consistency</a>, <a href="#">entailment</a>.</p> <p><b>modelling:</b> <a href="#">excluded</a>, <a href="#">domain definition</a>.</p>			

**Derived Collection**

$$\text{col}(\text{VARIABLES} - \text{collection}(\text{var} - \text{dvar}), [\text{item}(\text{var} - \text{VAR})])$$
**Arc input(s)**

VARIABLES VALUES

**Arc generator** $PRODUCT \mapsto \text{collection}(\text{variables}, \text{values})$ **Arc arity**

2

**Arc constraint(s)** $\text{variables.var} = \text{values.val}$ **Graph property(ies)** $\text{NARC} = 0$ **Graph model**

Figure 5.592 shows the initial graph associated with the **Example** slot. Since we use the  $\text{NARC} = 0$  graph property the corresponding final graph is empty.

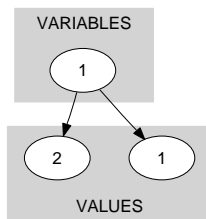


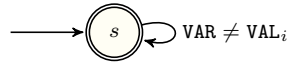
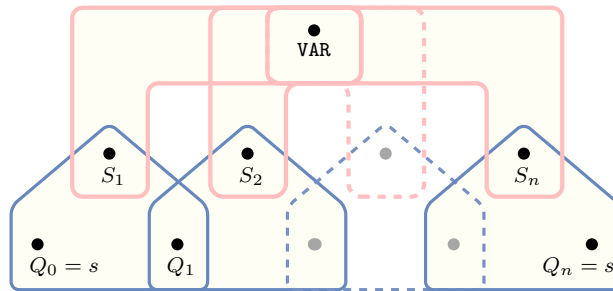
Figure 5.592: Initial graph of the `not_in` constraint (the final graph is empty)

**Signature**

Since 0 is the smallest number of arcs of the final graph we can rewrite  $\text{NARC} = 0$  to  $\text{NARC} \leq 0$ . This leads to simplify  $\text{NARC}$  to  $\text{NARC}$ .

**Automaton**

Figure 5.593 depicts the automaton associated with the `not_in` constraint. Let  $VAL_i$  be the `val` attribute of the  $i^{th}$  item of the `VALUES` collection. To each pair  $(VAR, VAL_i)$  corresponds a 0-1 signature variable  $S_i$  as well as the following signature constraint:  $VAR = VAL_i \Leftrightarrow S_i$ .

Figure 5.593: Automaton of the `not_in` constraintFigure 5.594: Hypergraph of the reformulation corresponding to the automaton of the `not_in` constraint

20030820

1815