

## 5.29 atleast

	DESCRIPTION	LINKS	GRAPH	AUTOMATON
Origin	CHIP			
Constraint	<code>atleast(N, VARIABLES, VALUE)</code>			
Synonym	<code>count.</code>			
Arguments	N : <code>int</code> VARIABLES : <code>collection(var-dvar)</code> VALUE : <code>int</code>			
Restrictions	$N \geq 0$ $N \leq  \text{VARIABLES} $ <code>required(VARIABLES, var)</code>			
Purpose	At least N variables of the VARIABLES collection are assigned value VALUE.			
Example	<code>(2, &lt;4, 2, 4, 5&gt;, 4)</code> The <code>atleast</code> constraint holds since at least 2 values of the collection <code>&lt;4, 2, 4, 5&gt;</code> are equal to value 4.			
Typical	$N > 0$ $N <  \text{VARIABLES} $ $ \text{VARIABLES}  > 1$			
Symmetries	<ul style="list-style-type: none"> <li>Items of VARIABLES are <code>permutable</code>.</li> <li>N can be <code>decreased</code> to any value <math>\geq 0</math>.</li> <li>An occurrence of a value of VARIABLES.var that is different from VALUE can be <code>replaced</code> by any other value.</li> </ul>			
Systems	<code>occurrenceMin</code> in <b>Choco</b> , <code>count</code> in <b>Gecode</b> , <code>count</code> in <b>JaCoP</b> , <code>count</code> in <b>SICStus</b> .			
Used in	<code>alldifferent_except_0</code> , <code>among_diff_0</code> , <code>int_value_precede</code> , <code>ith_pos_different_from_0</code> .			
See also	<b>common keyword:</b> <code>among</code> ( <i>value constraint</i> ). <b>comparison swapped:</b> <code>atmost</code> . <b>implied by:</b> <code>exactly</code> ( $\geq N$ replaced by $= N$ ). <b>related:</b> <code>roots</code> . <b>soft variant:</b> <code>open_atleast</code> ( <i>open constraint</i> ).			

**Keywords**

**characteristic of a constraint:** automaton, automaton with counters.

**constraint network structure:** alpha-acyclic constraint network(2).

**constraint type:** value constraint.

**filtering:** arc-consistency.

**modelling:** at least.

<b>Arc input(s)</b>	VARIABLES
<b>Arc generator</b>	$\text{SELF} \mapsto \text{collection}(\text{variables})$
<b>Arc arity</b>	1
<b>Arc constraint(s)</b>	$\text{variables.var} = \text{VALUE}$
<b>Graph property(ies)</b>	$\text{NARC} \geq N$

**Graph model**

Since each arc constraint involves only one vertex (**VALUE** is fixed), 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 5.55 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.

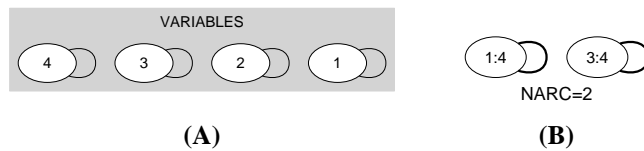


Figure 5.55: Initial and final graph of the atleast constraint

**Automaton**

Figure 5.56 depicts the automaton associated with the `atleast` 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  $S_i$ :  $VAR_i = VALUE \Leftrightarrow S_i$ . The automaton counts the number of variables of the `VARIABLES` collection that are assigned value `VALUE` and finally checks that this number is greater than or equal to `N`.

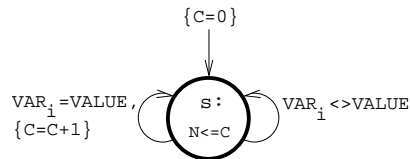


Figure 5.56: Automaton of the `atleast` constraint

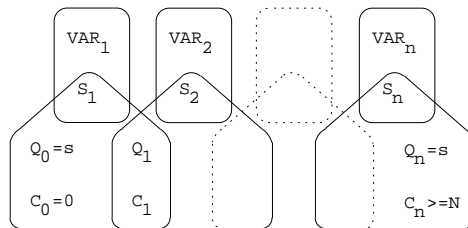


Figure 5.57: Hypergraph of the reformulation corresponding to the automaton of the `atleast` constraint