

5.46 cardinality_atleast

	DESCRIPTION	LINKS	GRAPH	AUTOMATON
Origin	Derived from global_cardinality .			
Constraint	<code>cardinality_atleast(ATLEAST, VARIABLES, VALUES)</code>			
Arguments	<p>ATLEAST : <code>dvar</code></p> <p>VARIABLES : <code>collection(var-dvar)</code></p> <p>VALUES : <code>collection(val-int)</code></p>			
Restrictions	<p>ATLEAST ≥ 0</p> <p>ATLEAST $\leq VARIABLES$</p> <p>required(VARIABLES, var)</p> <p>required(VALUES, val)</p> <p>distinct(VALUES, val)</p>			
Purpose	<p>ATLEAST is the minimum number of time that a value of VALUES is taken by the variables of the collection VARIABLES.</p>			
Example	<p><code>(1, <3, 3, 8>, <3, 8>)</code></p> <p>In this example, values 3 and 8 are respectively used 2, and 1 times. The <code>cardinality_atleast</code> constraint holds since its first argument ATLEAST = 1 is assigned to the minimum number of time that values 3 and 8 occur in the collection <code><3, 3, 8></code>.</p>			
Typical	<p>ATLEAST > 0</p> <p>ATLEAST $< VARIABLES$</p> <p>$VARIABLES > 1$</p> <p>$VALUES > 0$</p> <p>$VARIABLES > VALUES$</p>			
Symmetries	<ul style="list-style-type: none"> • Items of VARIABLES are permutable. • Items of VALUES are permutable. • An occurrence of a value of VARIABLES.var that does not belong to VALUES.val can be replaced by any other value that also does not belong to VALUES.val. • All occurrences of two distinct values in VARIABLES.var or VALUES.val can be swapped; all occurrences of a value in VARIABLES.var or VALUES.val can be renamed to any unused value. 			
Usage	An application of the <code>cardinality_atleast</code> constraint is to enforce a minimum use of values.			

- Remark** This is a restricted form of a variant of an [among](#) constraint and of the [global_cardinality](#) constraint. In the original [global_cardinality](#) constraint, one specifies for each value its minimum and maximum number of occurrences.
- Algorithm** See [global_cardinality](#) [309].
- See also** **generalisation:** [global_cardinality](#) (*single count variable replaced by an individual count variable for each value*).
- Keywords** **application area:** assignment.
characteristic of a constraint: automaton, automaton with array of counters.
constraint type: value constraint.
filtering: arc-consistency.
final graph structure: acyclic, bipartite, no loop.
modelling: at least.

Arc input(s)	VARIABLES VALUES
Arc generator	<i>PRODUCT</i> \mapsto <code>collection(variables, values)</code>
Arc arity	2
Arc constraint(s)	<code>variables.var \neq values.val</code>
Graph property(ies)	$MAX_ID = VARIABLES - ATLEAST$
Graph class	<ul style="list-style-type: none"> • ACYCLIC • BIPARTITE • NO_LOOP

Graph model

Using directly the graph property $MIN_ID = ATLEAST$, and replacing the disequality of the arc constraint by an equality does not work since it ignores values that are not assigned to any variable. This comes from the fact that isolated vertices are removed from the final graph.

Parts (A) and (B) of Figure 5.86 respectively show the initial and final graph associated with the **Example** slot. Since we use the MAX_ID graph property, the vertex with the maximum number of predecessor (i.e., namely two predecessors) is stressed with a double circle. As a consequence the first argument $ATLEAST$ of the `cardinality_atleast` constraint is assigned to the total number of variables 3 minus 2.

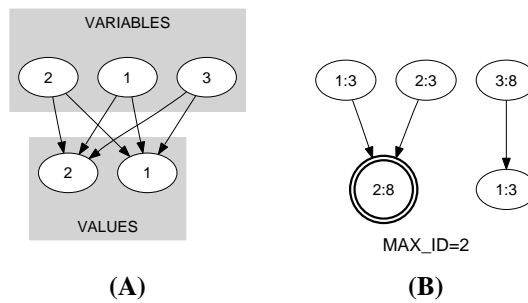


Figure 5.86: Initial and final graph of the `cardinality_atleast` constraint

Automaton

Figure 5.87 depicts the automaton associated with the `cardinality_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 : $VAR_i \in \text{VALUES} \Leftrightarrow S_i$.

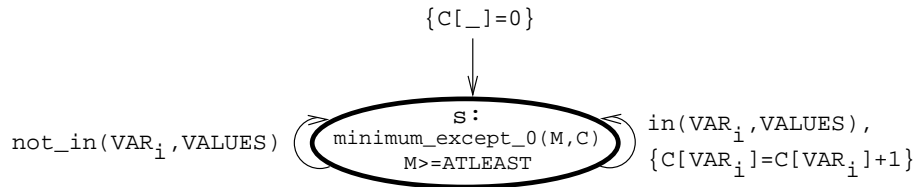


Figure 5.87: Automaton of the `cardinality_atleast` constraint