5.58 cardinality_atleast

DESCRIPTION LINKS GRAPH AUTOMATON

Origin Derived from global_cardinality.

Constraint cardinality_atleast(ATLEAST, VARIABLES, VALUES)

Arguments

ATLEAST	:	dvar
VARIABLES	:	collection(var−dvar)
VALUES	:	collection(val−int)

Restrictions

ATLEAST ≥ 0
ATLEAST $\leq |$VARIABLES$|$
required(VARIABLES, var)
required(VALUES, val)
distinct(VALUES, val)

Purpose ATLEAST is the minimum number of time that a value of VALUES is taken by the variables of the collection VARIABLES.

Example $(1, \langle 3, 3, 8\rangle, \langle 3, 8\rangle)$

In this example, values 3 and 8 are respectively used 2, and 1 times. The cardinality_atleast 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 $\langle 3, 3, 8\rangle$.

Typical

ATLEAST > 0
ATLEAST $< |$VARIABLES$|$
$|$VARIABLES$| > 1$
$|$VALUES$| > 0$
$|$VARIABLES$| > |$VALUES$|$

Symmetries

- 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.

Arg. properties

Functional dependency: ATLEAST determined by VARIABLES and VALUES.

Usage An application of the cardinality_atleast 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 [342].

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 arguments: pure functional dependency.

constraint type: value constraint.

filtering: arc-consistency.

final graph structure: acyclic, bipartite, no loop.

modelling: functional dependency, at least.

Arc input(s)	VARIABLES VALUES
Arc generator	$PRODUCT \mapsto$ collection(variables, values)
Arc arity	2
Arc constraint(s)	variables.var $\neq$ values.val
Graph property(ies)	MAX_ID $=$ |VARIABLES| $-$ ATLEAST
Graph class	• 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 5.148 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.

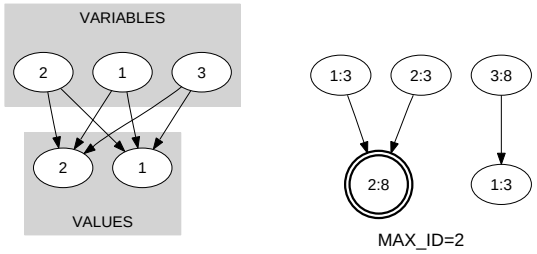

Figure 5.148: Initial and final graph of the cardinality_atleast constraint

Automaton

Figure 5.149 depicts the automaton associated with the `cardinality_atleast` constraint. To each variable $\texttt{VAR}_i$ of the collection `VARIABLES` corresponds a 0-1 signature variable S_i. The following signature constraint links $\texttt{VAR}_i$ and S_i: $\texttt{VAR}_i \in \texttt{VALUES} \Leftrightarrow S_i$.

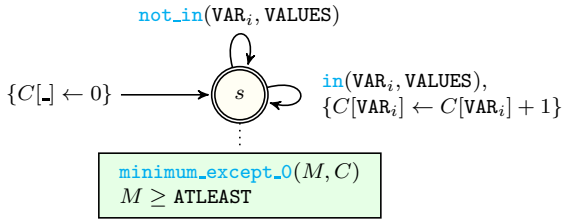

Figure 5.149: Automaton of the `cardinality_atleast` constraint