5.59 cardinality_atmost

DESCRIPTION LINKS GRAPH AUTOMATON

Origin Derived from global_cardinality.

Constraint cardinality_atmost(ATMOST, VARIABLES, VALUES)

Arguments

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

Restrictions

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

Purpose ATMOST is the maximum number of occurrences of each value of VALUES within the variables of the collection VARIABLES.

Example $(2, \langle 2, 1, 7, 1, 2\rangle, \langle 5, 7, 2, 9\rangle)$

In this example, values 5, 7, 2 and 9 occur respectively 0, 1, 2 and 0 times within the collection $\langle 2, 1, 7, 1, 2\rangle$. As a consequence, the cardinality_atmost constraint holds since its first argument ATMOST is assigned to the maximum number of occurrences 2.

Typical

ATMOST > 0
ATMOST $< |$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: ATMOST determined by VARIABLES and VALUES.

Usage An application of the cardinality_atmost constraint is to enforce a maximum use of values.

Remark This is a restricted form of a variant of the 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)*, multi_inter_distance *(window of size* 1 *replaced by window of* DIST *consecutive values)*.

implied by: among.

Keywords **application area:** assignment.

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

constraint arguments: pure functional dependency.

constraint type: value constraint.

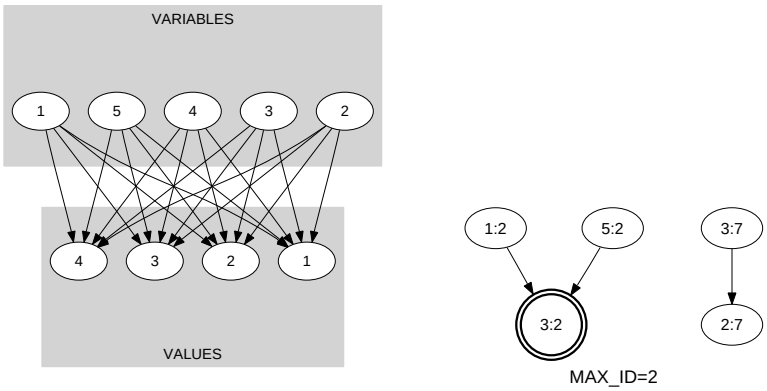
filtering: arc-consistency.

final graph structure: acyclic, bipartite, no loop.

modelling: at most, functional dependency.

Arc input(s)	VARIABLES VALUES
Arc generator	$PRODUCT \mapsto$ collection(variables, values)
Arc arity	2
Arc constraint(s)	variables.var = values.val
Graph property(ies)	MAX_ID= ATMOST
Graph class	• ACYCLIC • BIPARTITE • NO_LOOP

Graph model Parts (A) and (B) of Figure 5.133 respectively show the initial and final graph associated with the **Example** slot. Since we use the MAX_ID graph property, the vertex that has the maximum number of predecessor is stressed with a double circle.



(A) **(B)**

Figure 5.133: Initial and final graph of the cardinality_atmost constraint

Automaton

Figure 5.134 depicts the automaton associated with the `cardinality_atmost` 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: $\texttt{VAR}_i \in \texttt{VALUES} \Leftrightarrow S_i$.

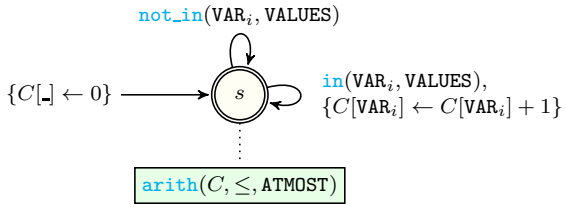


Figure 5.134: Automaton of the `cardinality_atmost` constraint