5.245 max_nvalue

DESCRIPTION LINKS GRAPH AUTOMATON

Origin Derived from nvalue.

Constraint max_nvalue(MAX, VARIABLES)

Arguments

MAX	:	dvar
VARIABLES	:	collection(var−dvar)

Restrictions

$\texttt{MAX} \geq 1$
$\texttt{MAX} \leq |\texttt{VARIABLES}|$
required(VARIABLES, var)

Purpose MAX is the maximum number of times that the same value is taken by the variables of the collection VARIABLES.

Example $(3, \langle 9, 1, 7, 1, 1, 6, 7, 7, 4, 9\rangle)$
$(1, \langle 9, 1, 7, 3, 2, 6\rangle)$
$(6, \langle 5, 5, 5, 5, 5, 5\rangle)$

In the first example, values $1, 4, 6, 7, 9$ are respectively used $3, 1, 1, 3, 2$ times. So the maximum number of time MAX that a same value occurs is 3. Consequently the corresponding max_nvalue constraint holds.

Typical

$\texttt{MAX} > 1$
$\texttt{MAX} < |\texttt{VARIABLES}|$
$|\texttt{VARIABLES}| > 1$
range(VARIABLES.var) > 1

Symmetries

- Items of VARIABLES are permutable.
- All occurrences of two distinct values of VARIABLES.var can be swapped; all occurrences of a value of VARIABLES.var can be renamed to any unused value.

Arg. properties

Functional dependency: MAX determined by VARIABLES.

Usage This constraint may be used in order to replace a set of count or among constraints were one would have to generate explicitly one constraint for each potential value. Also useful for constraining the number of occurrences of the mostly used value without knowing this value in advance and without giving explicitly an upper limit on the number of occurrences of each value as it is done in the global_cardinality constraint.

Reformulation Assume that VARIABLES is not empty. Let α and β respectively denote the smallest and largest possible values that can be assigned to the variables of the VARIABLES collection. Let the variables $O_\alpha, O_{\alpha+1}, \ldots, O_\beta$ respectively correspond to the number of occurrences of values $\alpha, \alpha + 1, \ldots, \beta$ within the variables of the VARIABLES collection.

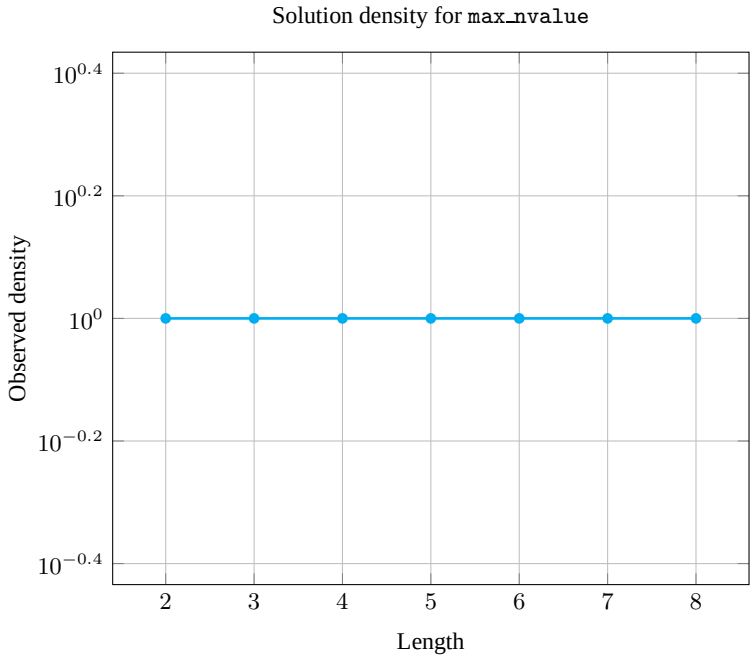
The `max_nvalue` constraint can be expressed as the conjunction of the following two constraints:

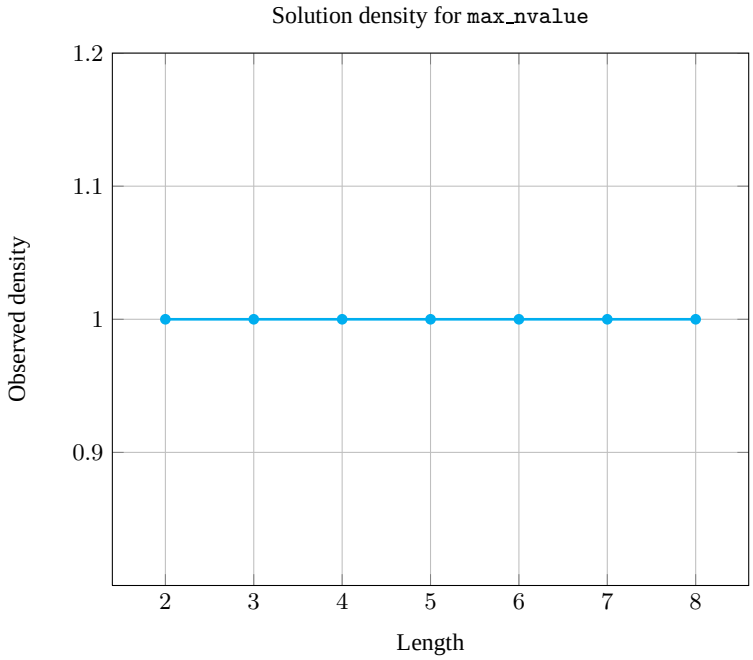
`global_cardinality` (`VARIABLES`,
$\langle$`val` $-\ \alpha$ `noccurrence` $-\ O_\alpha$,
`val` $-\ \alpha + 1$ `noccurrence` $-\ O_{\alpha+1}$,
$\ldots$
`val` $-\ \beta$ `noccurrence` $-\ O_\beta\rangle$),
`maximum`(`MAX`, $\langle O_\alpha, O_{\alpha+1}, \ldots, O_\beta\rangle$).

Counting

Length (n)	2	3	4	5	6	7	8
Solutions	9	64	625	7776	117649	2097152	43046721

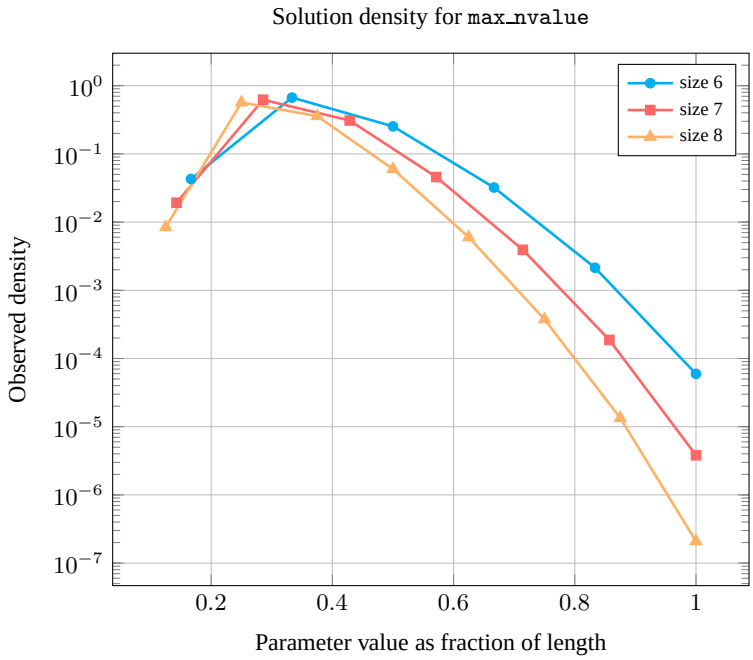
Number of solutions for `max_nvalue`: domains $0..n$

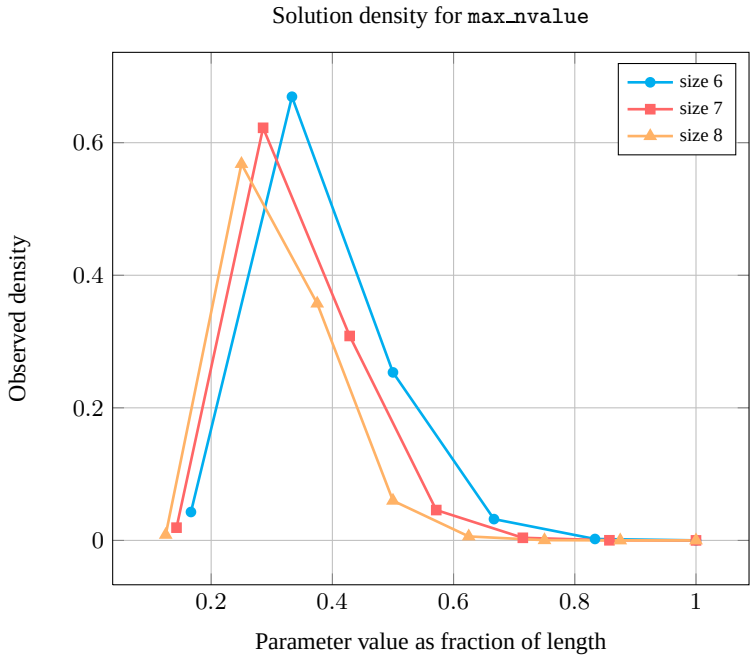




Length (n)		2	3	4	5	6	7	8
Total		9	64	625	7776	117649	2097152	43046721
Parameter value	1	6	24	120	720	5040	40320	362880
	2	3	36	420	5400	78750	1305360	24449040
	3	-	4	80	1500	29820	646800	15382080
	4	-	-	5	150	3780	96040	2577960
	5	-	-	-	6	252	8232	258048
	6	-	-	-	-	7	392	16128
	7	-	-	-	-	-	8	576
	8	-	-	-	-	-	-	9

Solution count for `max_nvalue`: domains $0..n$





See also **common keyword:** among (*counting constraint*), count, global_cardinality (*value constraint*,*counting constraint*), min_nvalue, nvalue (*counting constraint*).

Keywords

application area: assignment.

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

constraint arguments: pure functional dependency.

constraint type: value constraint, counting constraint.

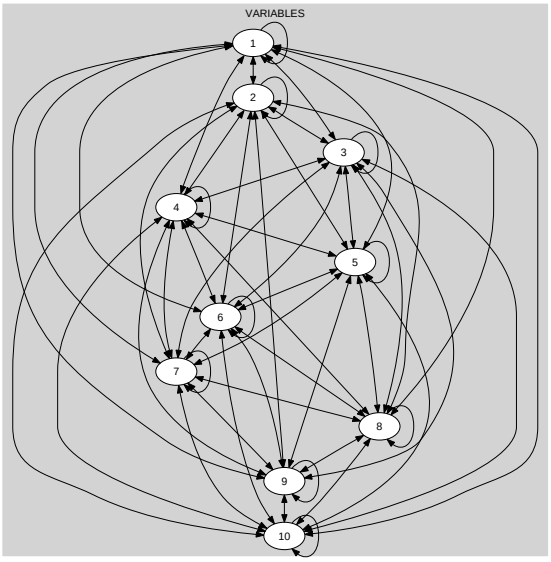
final graph structure: equivalence.

modelling: maximum number of occurrences, functional dependency.

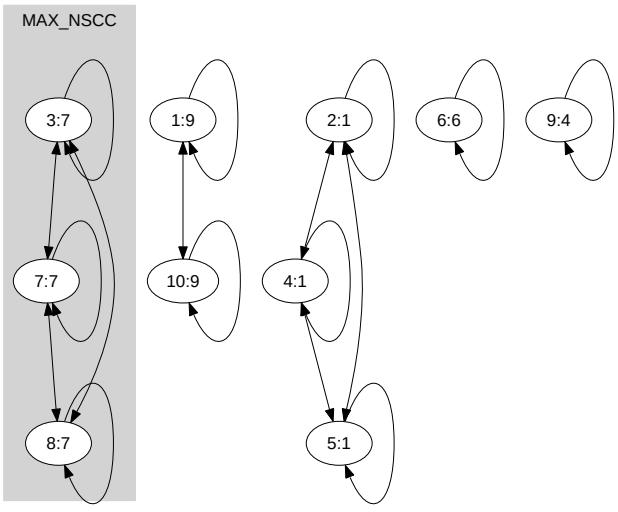
Arc input(s)	VARIABLES
Arc generator	$CLIQUE \mapsto$ collection(variables1, variables2)
Arc arity	2
Arc constraint(s)	variables1.var = variables2.var
Graph property(ies)	**MAX_NSCC** = MAX

Graph model Because of the arc constraint, each strongly connected component of the final graph corresponds to a distinct value that is assigned to a subset of variables of the VARIABLES collection. Therefore the number of vertices of the largest strongly connected component is equal to the mostly used value.

Parts (A) and (B) of Figure 5.536 respectively show the initial and final graph associated with the first example of the **Example** slot. Since we use the **MAX_NSCC** graph property, we show the largest strongly connected component of the final graph.



(A)



(B)

Figure 5.536: Initial and final graph of the `max_nvalue` constraint

Automaton

Figure 5.537 depicts the automaton associated with the `max_nvalue` constraint. To each item of the collection `VARIABLES` corresponds a signature variable S_i that is equal to 0.

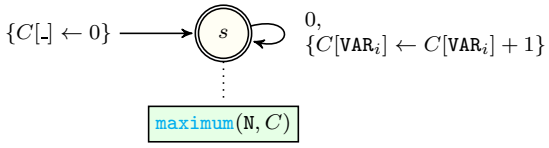


Figure 5.537: Automaton of the `max_nvalue` constraint