5.258 min_nvalue

DESCRIPTION LINKS GRAPH AUTOMATON

Origin N. Beldiceanu

Constraint min_nvalue(MIN, VARIABLES)

Arguments

```
MIN       : dvar
VARIABLES : collection(var−dvar)
```

Restrictions

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

Purpose MIN is the minimum number of times that the same value is taken by the variables of the collection VARIABLES.

Example

$(2, \langle 9, 1, 7, 1, 1, 7, 7, 7, 7, 9\rangle)$
$(5, \langle 8, 8, 8, 8, 8\rangle)$
$(2, \langle 1, 8, 1, 8, 1\rangle)$

In the first example, values $1, 7, 9$ are respectively used $3, 5, 2$ times. So the minimum number of time MIN that a same value occurs is 2. Consequently the corresponding min_nvalue constraint holds.

Typical

$2 * \texttt{MIN} \leq |\texttt{VARIABLES}|$
$|\texttt{VARIABLES}| > 1$
$\texttt{range}(\texttt{VARIABLES.var}) > 1$

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: MIN 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 less used value without knowing this value in advance and without giving explicitly a lower 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 min_nvalue constraint can be expressed as the conjunction of the following two constraints:

global_cardinality $(\texttt{VARIABLES},$
$\langle \texttt{val} - \alpha\ \texttt{noccurrence} - O_{\alpha},$
$\texttt{val} - \alpha + 1\ \texttt{noccurrence} - O_{\alpha+1},$
$\ldots$
$\texttt{val} - \beta\ \texttt{noccurrence} - O_{\beta}\rangle),$
min_n$(\texttt{MIN}, 1, \langle 0, O_{\alpha}, O_{\alpha+1}, \ldots, O_{\beta}\rangle)$.

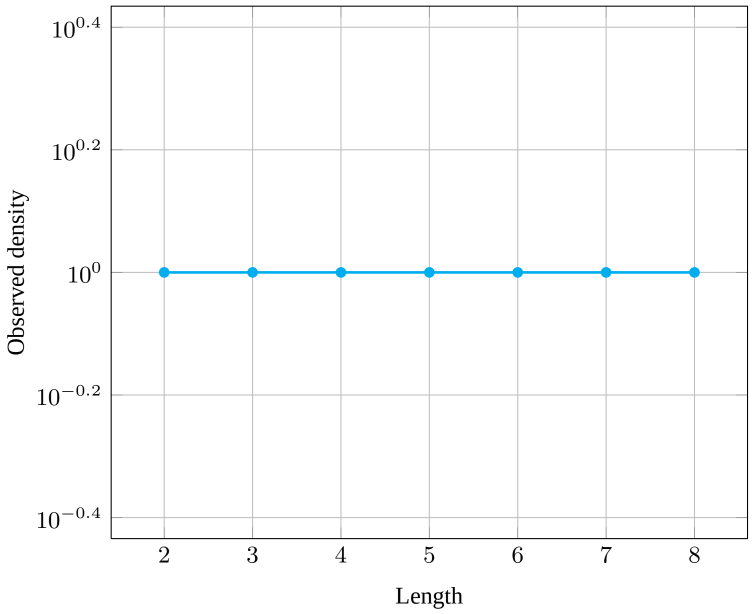
We use a min_n constraint (with its RANK parameter set to 1) instead of a minimum constraint in order to discard the smallest value 0.

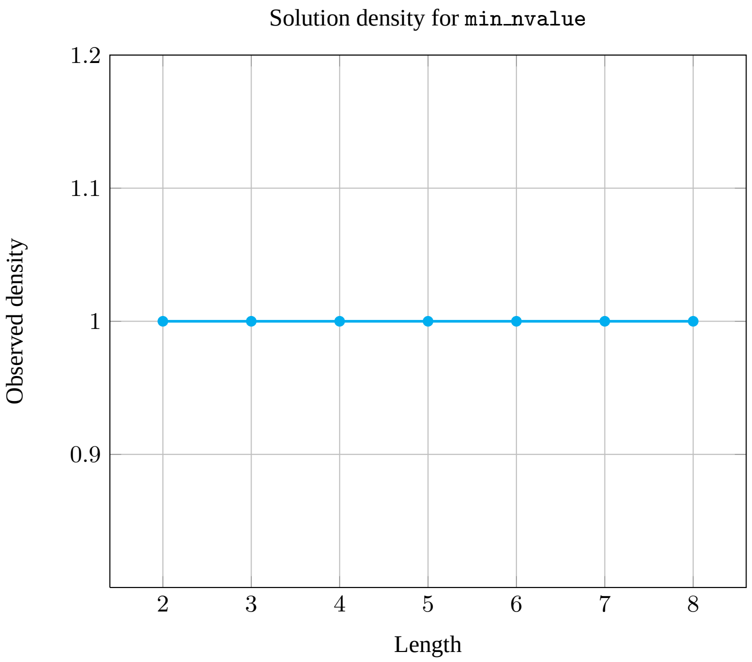
Counting

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

Number of solutions for min_nvalue: domains $0..n$

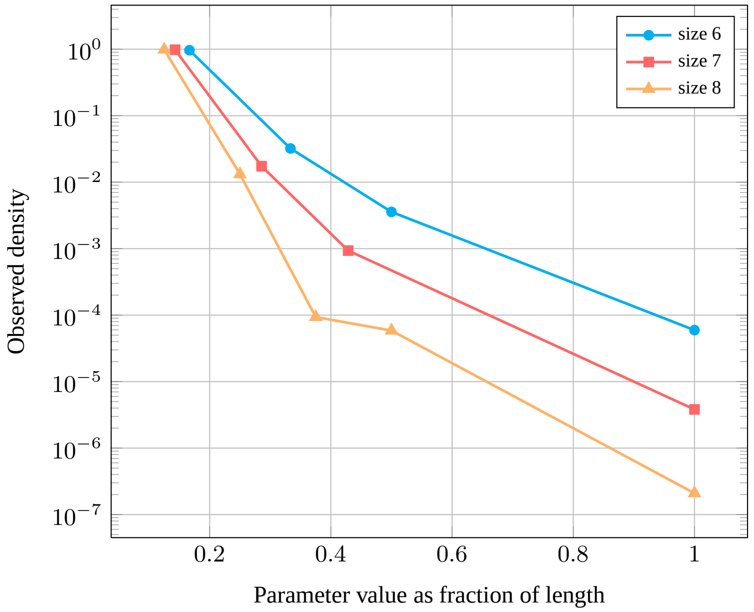
Solution density for min_nvalue



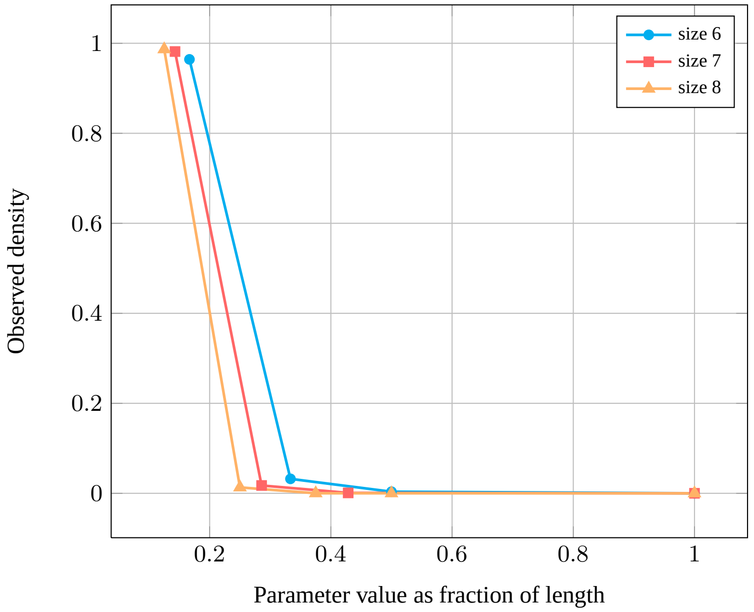


Length (n)		2	3	4	5	6	7	8
Total		9	64	625	7776	117649	2097152	43046721
Parameter value	1	6	60	560	7470	113442	2058728	42473664
	2	3	-	60	300	3780	36456	566496
	3	-	4	-	-	420	1960	4032
	4	-	-	5	-	-	-	2520
	5	-	-	-	6	-	-	-
	6	-	-	-	-	7	-	-
	7	-	-	-	-	-	8	-
	8	-	-	-	-	-	-	9

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



Solution density for `min_nvalue`



Solution density for `min_nvalue`

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

Keywords

application area: assignment.

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

constraint arguments: pure functional dependency.

constraint type: value constraint, counting constraint.

final graph structure: equivalence.

modelling: minimum number of occurrences, functional dependency.

Cond. implications

$\texttt{min\_nvalue}(\texttt{MIN}, \texttt{VARIABLES})$
  with $\texttt{MIN} < |\texttt{VARIABLES}|$
implies $\texttt{atleast\_nvalue}(\texttt{NVAL}, \texttt{VARIABLES})$
  when $\texttt{NVAL} = 2$.

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

Graph model Parts (A) and (B) of Figure 5.558 respectively show the initial and final graph associated with the first example of the **Example** slot. Since we use the **MIN_NSCC** graph property, we show the smallest strongly connected component of the final graph associated with the first example of the **Example** slot.

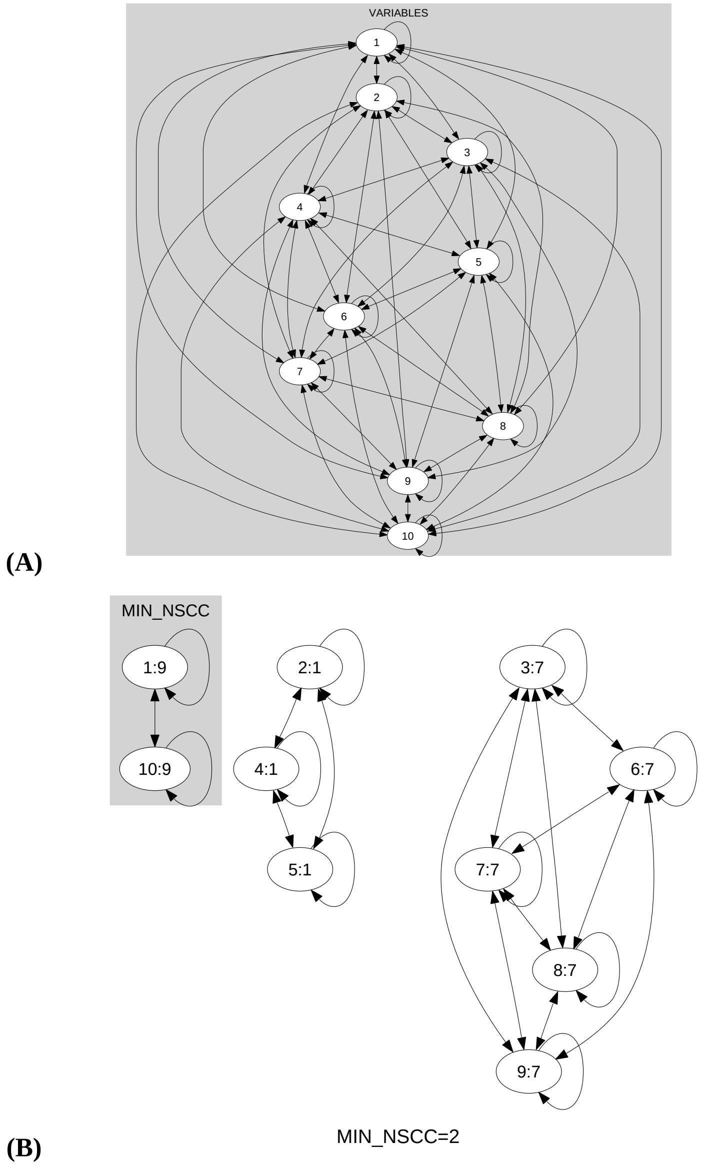


Figure 5.558: Initial and final graph of the `min_nvalue` constraint

Automaton Figure 5.559 depicts the automaton associated with the `min_nvalue` constraint. To each item of the collection `VARIABLES` corresponds a signature variable S_i that is equal to 0.

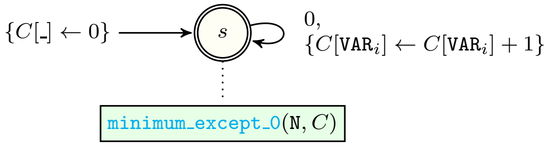

$\{C[\_] \leftarrow 0\}$
s
$0,$
$\{C[\mathtt{VAR}_i] \leftarrow C[\mathtt{VAR}_i] + 1\}$
minimum_except_0$(\mathtt{N}, C)$

Figure 5.559: Automaton of the `min_nvalue` constraint