5.256 min_n

DESCRIPTION LINKS GRAPH AUTOMATON

Origin [27]

Constraint min_n(MIN, RANK, VARIABLES)

Arguments

MIN : dvar
RANK : int
VARIABLES : collection(var−dvar)

Restrictions

$|\texttt{VARIABLES}| > 0$
$\texttt{RANK} \geq 0$
$\texttt{RANK} < |\texttt{VARIABLES}|$
required(VARIABLES, var)

Purpose

MIN is the minimum value of rank RANK (i.e., the $\texttt{RANK}^{th}$ smallest distinct value, identical values are merged) of the collection of domain variables VARIABLES. The minimum value has rank 0.

Example

$(3, 1, \langle 3, 1, 7, 1, 6\rangle)$

The min_n constraint holds since its first argument MIN = 3 is fixed to the second (i.e., RANK + 1) smallest distinct value of the collection $\langle 3, 1, 7, 1, 6\rangle$. Note that identical values are only counted once: this is why the minimum of order 1 is 3 instead of 1.

Typical

$\texttt{RANK} > 0$
$\texttt{RANK} < 3$
$|\texttt{VARIABLES}| > 1$
range(VARIABLES.var) > 1

Symmetries

- Items of VARIABLES are permutable.
- One and the same constant can be added to MIN as well as to the var attribute of all items of VARIABLES.

Arg. properties

Functional dependency: MIN determined by RANK and VARIABLES.

Algorithm [27].

Reformulation

The constraint among_var$(1, \langle\texttt{MIN}\rangle, \texttt{VARIABLES})$ enforces MIN to be assigned one of the values of VARIABLES. The constraint nvalue(NVAL, VARIABLES) provides a hand on the number of distinct values assigned to the variables of VARIABLES. By associating to each variable V_i $(i \in [1, |\texttt{VARIABLES}|])$ of the VARIABLES collection a *rank* variable $R_i \in [0, |\texttt{VARIABLES}| - 1]$ with the reified constraint $R_i = \texttt{RANK} \Leftrightarrow V_i = \texttt{MIN}$, the inequality $R_i < \texttt{NVAL}$, and by creating for each pair of variables V_i, V_j $(i, j < i \in [1, |\texttt{VARIABLES}|])$

the reified constraints
$V_i < V_j \Leftrightarrow R_i < R_j$,
$V_i = V_j \Leftrightarrow R_i = R_j$,
$V_i > V_j \Leftrightarrow R_i > R_j$,
one can reformulate the `min_n` constraint in term of $3 \cdot \frac{|\mathtt{VARIABLES}|\cdot(|\mathtt{VARIABLES}|-1)}{2} + 1$ reified constraints.

See also

comparison swapped: `max_n`.

generalisation: `minimum` *(absolute minimum replaced by minimum or order* `n`*).*

used in reformulation: `among_var`, `nvalue`.

Keywords

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

constraint arguments: pure functional dependency.

constraint type: order constraint.

modelling: functional dependency.

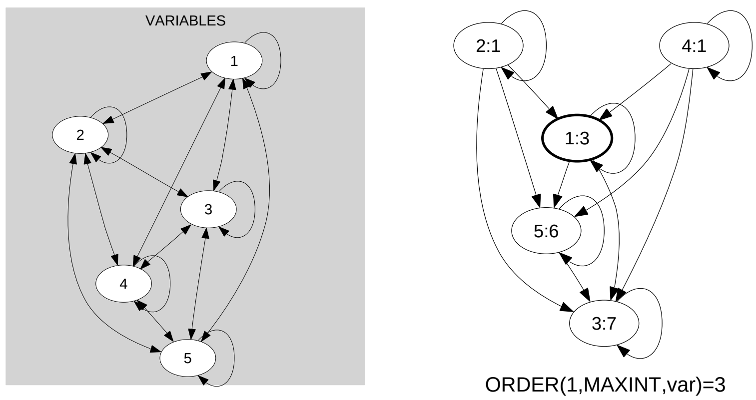
Cond. implications

- $\mathtt{min\_n}(\mathtt{MIN}, \mathtt{RANK}, \mathtt{VARIABLES})$
  implies $\mathtt{atleast}(\mathtt{N}, \mathtt{VARIABLES}, \mathtt{MIN})$
  when $\mathtt{N} = 1$.

- $\mathtt{min\_n}(\mathtt{MIN}, \mathtt{RANK}, \mathtt{VARIABLES})$
  with $\mathtt{RANK} = 1$
  and $\mathtt{minval}(\mathtt{VARIABLES.var}) = 1$
  implies $\mathtt{minimum\_greater\_than}(\mathtt{VAR1}, \mathtt{VAR2}, \mathtt{VARIABLES})$.

Arc input(s)	VARIABLES
Arc generator	$CLIQUE \mapsto$ collection(variables1, variables2)
Arc arity	2
Arc constraint(s)	$\bigvee\left(\begin{array}{l}\texttt{variables1.key} = \texttt{variables2.key},\\ \texttt{variables1.var} < \texttt{variables2.var}\end{array}\right)$
Graph property(ies)	**ORDER**(RANK, MAXINT, var) = MIN

Graph model Parts (A) and (B) of Figure 5.537 respectively show the initial and final graph associated with the **Example** slot. Since we use the **ORDER** graph property, the vertex of rank 1 (without considering the loops) of the final graph is shown in grey.



(A) (B)

Figure 5.537: Initial and final graph of the min_n constraint

Automaton Figure 5.538 depicts the automaton associated with the `min_n` constraint. Figure 5.538 depicts the automaton associated with the `min_n` constraint. To each item of the collection `VARIABLES` corresponds a signature variable S_i that is equal to 1.

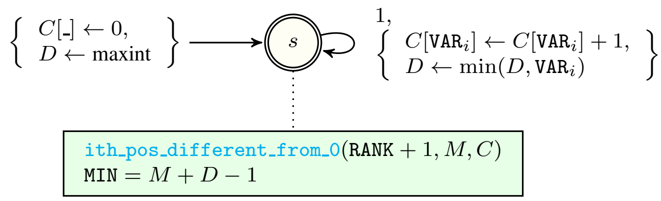


Figure 5.538: Automaton of the `min_n` constraint