5.263 minimum_except_0

DESCRIPTION LINKS GRAPH AUTOMATON

Origin Derived from minimum.

Constraint minimum_except_0(MIN, VARIABLES, DEFAULT)

Arguments

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

Restrictions

MIN > 0
MIN $\leq$ DEFAULT
$|$VARIABLES$| > 0$
required(VARIABLES, var)
VARIABLES.var ≥ 0
VARIABLES.var $\leq$ DEFAULT
DEFAULT > 0

Purpose All variables of the collection VARIABLES are assigned a value that belongs to interval $[0,$ DEFAULT$]$. MIN is the minimum value of the collection of domain variables VARIABLES, ignoring all variables that take 0 as value. When all variables of the collection VARIABLES are assigned value 0, MIN is set to the default value DEFAULT.

Example

$(3, \langle 3, 7, 6, 7, 4, 7\rangle, 1000000)$
$(2, \langle 3, 2, 0, 7, 2, 6\rangle, 1000000)$
$(1000000, \langle 0, 0, 0, 0, 0, 0\rangle, 1000000)$

The three examples of the minimum_except_0 constraint respectively hold since:

- Within the first example, MIN is set to the minimum value 3 of the collection $\langle 3, 7, 6, 7, 4, 7\rangle$.
- Within the second example, MIN is set to the minimum value 2 (ignoring value 0) of the collection $\langle 3, 2, 0, 7, 2, 6\rangle$.
- Finally within the third example, MIN is set to the default value 1000000 since all items of the collection $\langle 0, 0, 0, 0, 0, 0\rangle$ are set to 0.

Typical

$|$VARIABLES$| > 1$
range(VARIABLES.var) > 1
atleast(1, VARIABLES, 0)

Symmetries

- Items of VARIABLES are permutable.
- All occurrences of two distinct values of VARIABLES.var can be swapped.

Arg. properties

Functional dependency: MIN determined by VARIABLES and DEFAULT.

Remark The joker value 0 makes sense only because we restrict the variables of the VARIABLES collection to take non-negative values.

Reformulation By (1) 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 $R_i = 1 \Leftrightarrow V_i = \texttt{MIN}$, and by creating for each pair of variables V_i, V_j ($i, j < i \in [1, |\texttt{VARIABLES}|]$) the reified constraints

$V_i < V_j \Leftrightarrow R_i < R_j$,
$V_i = V_j \Leftrightarrow R_i = R_j$,
$V_i > V_j \Leftrightarrow R_i > R_j$,

and by (2) creating the reified constraint

$V_1 = 0 \wedge V_2 = 0 \wedge \cdots \wedge V_n = 0 \Rightarrow \texttt{MIN} = \texttt{DEFAULT}$,

one can reformulate the minimum_except_0 constraint in term of $3 \cdot \frac{|\texttt{VARIABLES}|\cdot(|\texttt{VARIABLES}|-1)}{2} + 2$ reified constraints.

See also **hard version:** minimum *(value 0 is not ignored any more)*.

Keywords **characteristic of a constraint:** joker value, minimum, automaton, automaton without counters, reified automaton constraint.

constraint arguments: pure functional dependency.

constraint network structure: centered cyclic(1) constraint network(1).

constraint type: order constraint.

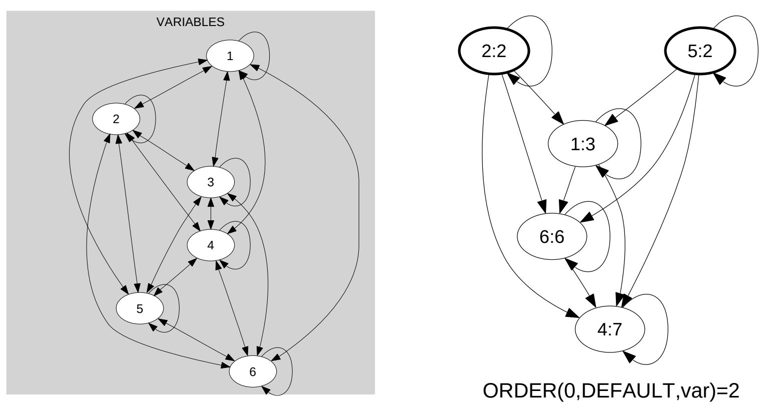
modelling: functional dependency.

Cond. implications minimum_except_0(MIN, VARIABLES, DEFAULT)
with maxval(VARIABLES.var) < DEFAULT
implies atmost(N, VARIABLES, VALUE).

Arc input(s)	VARIABLES
Arc generator	$CLIQUE \mapsto$ collection(variables1, variables2)
Arc arity	2
Arc constraint(s)	• variables1.var $\neq 0$ • variables2.var $\neq 0$ • $\bigvee\left(\begin{array}{l}\texttt{variables1.key} = \texttt{variables2.key},\\ \texttt{variables1.var} < \texttt{variables2.var}\end{array}\right)$
Graph property(ies)	**ORDER**(0, DEFAULT, var) = MIN

Graph model Because of the first two conditions of the arc constraint, all vertices that correspond to 0 will be removed from the final graph.

Parts (A) and (B) of Figure 5.562 respectively show the initial and final graph of the second example of the **Example** slot. Since we use the **ORDER** graph property, the vertices of rank 0 (without considering the loops) of the final graph are outlined with a thick circle.



(A) **(B)**

Figure 5.562: Initial and final graph of the minimum_except_0 constraint

Since the graph associated with the third example does not contain any vertex, **ORDER** returns the default value DEFAULT.

Automaton Figure 5.563 depicts the automaton associated with the `minimum_except_0` constraint. Let $\texttt{VAR}_i$ be the i^{th} variable of the `VARIABLES` collection. To each pair $(\texttt{MIN}, \texttt{VAR}_i)$ corresponds a signature variable S_i as well as the following signature constraint:

$((\texttt{VAR}_i = 0) \wedge (\texttt{MIN} \neq \texttt{DEFAULT})) \Leftrightarrow S_i = 0 \wedge$

$((\texttt{VAR}_i = 0) \wedge (\texttt{MIN} = \texttt{DEFAULT})) \Leftrightarrow S_i = 1 \wedge$

$((\texttt{VAR}_i \neq 0) \wedge (\texttt{MIN} = \texttt{VAR}_i)) \Leftrightarrow S_i = 2 \wedge$

$((\texttt{VAR}_i \neq 0) \wedge (\texttt{MIN} < \texttt{VAR}_i)) \Leftrightarrow S_i = 3 \wedge$

$((\texttt{VAR}_i \neq 0) \wedge (\texttt{MIN} > \texttt{VAR}_i)) \Leftrightarrow S_i = 4.$

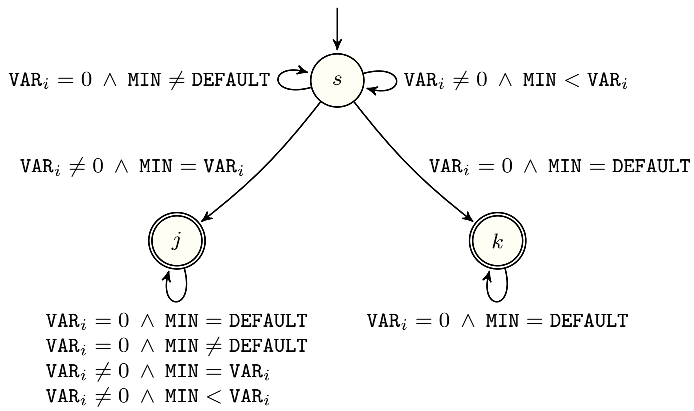


Figure 5.563: Automaton of the `minimum_except_0` constraint

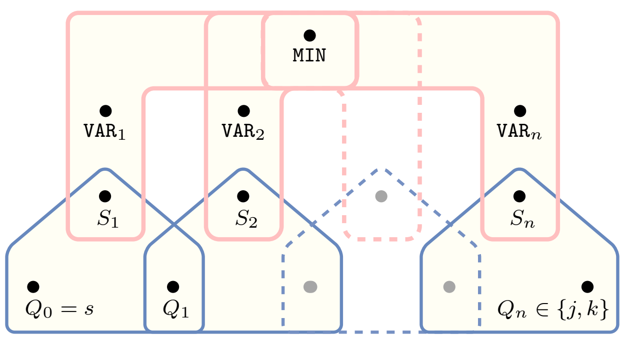


Figure 5.564: Hypergraph of the reformulation corresponding to the automaton of the `minimum_except_0` constraint