

5.300 open_global_cardinality_low_up

	DESCRIPTION	LINKS	GRAPH
Origin	[427]		
Constraint	open_global_cardinality_low_up(S, VARIABLES, VALUES)		
Arguments	<div>S : svar</div> <div>VARIABLES : collection(var-dvar)</div> <div>VALUES : collection(val-int, omin-int, omax-int)</div>		
Restrictions	<div>$S \geq 1$</div> <div>$S \leq \text{VARIABLES}$</div> <div>required(VARIABLES, var)</div> <div>$\text{VALUES} > 0$</div> <div>required(VALUES, [val, omin, omax])</div> <div>distinct(VALUES, val)</div> <div>$\text{VALUES.omin} \geq 0$</div> <div>$\text{VALUES.omax} \leq \text{VARIABLES}$</div> <div>$\text{VALUES.omin} \leq \text{VALUES.omax}$</div>		
Purpose	<div>Each value $\text{VALUES}[i].\text{val}$ ($1 \leq i \leq \text{VALUES}$) should be taken by at least $\text{VALUES}[i].\text{omin}$ and at most $\text{VALUES}[i].\text{omax}$ variables of the VARIABLES collection for which the corresponding position belongs to the set S. Positions are numbered from 1.</div>		
Example	<div>$\left(\begin{array}{l} \{2, 3, 4\}, \\ \langle 3, 3, 8, 6 \rangle, \\ \left\langle \begin{array}{lll} \text{val} - 3 & \text{omin} - 1 & \text{omax} - 3, \\ \text{val} - 5 & \text{omin} - 0 & \text{omax} - 1, \\ \text{val} - 6 & \text{omin} - 1 & \text{omax} - 2 \end{array} \right\rangle \end{array} \right)$</div>		
<div>The open_global_cardinality_low_up constraint holds since:</div> <div><ul style="list-style-type: none">Values 3, 5 and 6 are respectively used 1 ($1 \leq 1 \leq 3$), 0 ($0 \leq 0 \leq 1$) and 1 ($1 \leq 1 \leq 2$) times within the collection $\langle 3, 3, 8, 6 \rangle$ (the first item 3 of $\langle 3, 3, 8, 6 \rangle$ is ignored since value 1 does not belong to the first argument $S = \{2, 3, 4\}$ of the open_global_cardinality_low_up constraint).No constraint was specified for value 8.</div>			
Typical	<div>$\text{VARIABLES} > 1$</div> <div>range(VARIABLES.var) > 1</div> <div>$\text{VALUES} > 1$</div> <div>$\text{VALUES.omin} \leq \text{VARIABLES}$</div> <div>$\text{VALUES.omax} > 0$</div> <div>$\text{VALUES.omax} \leq \text{VARIABLES}$</div> <div>$\text{VARIABLES} > \text{VALUES}$</div>		

Symmetries

- 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.

Usage

In their article [427], W.-J. van Hoeve and J.-C. Régim motivate the `open_global_cardinality_low_up` constraint by the following scheduling problem. Consider a set of activities (where each activity has a fixed duration 1 and a start variable) that can be processed on two factory lines such that all the activities that will be processed on a given line must be pairwise distinct. This can be modelled by using one `open_global_cardinality_low_up` constraint for each line, involving all the start variables as well as a set variable whose final value specifies the set of activities assigned to that specific factory line.

Note that this can also be directly modelled by a single `diffn` constraint. This is done by introducing an assignment variable for each activity. The initial domain of each assignment variable consists of two values that respectively correspond to the two factory lines.

Algorithm

A slight adaptation of the [flow](#) model that handles the original `global_cardinality` constraint [342] is described in [427].

See also

common keyword: `global_cardinality` (*assignment, counting constraint*).

generalisation: `open_global_cardinality` (*fixed interval replaced by variable*).

hard version: `global_cardinality_low_up`.

specialisation: `open_alldifferent` (*each active value¹⁵ should occur at most once*).

used in graph description: `in_set`.

Keywords

application area: `assignment`.

constraint arguments: constraint involving set variables.

constraint type: `open constraint`, `value constraint`, `counting constraint`.

filtering: `flow`.

¹⁵An *active value* corresponds to a value occurring at a position mentioned in the set S.

	For all items of VALUES:
Arc input(s)	VARIABLES
Arc generator	SELF \mapsto collection(variables)
Arc arity	1
Arc constraint(s)	<ul style="list-style-type: none">• variables.var = VALUES.val• in_set(variables.key, S)
Graph property(ies)	<ul style="list-style-type: none">• <u>NVERTEX</u> \geq VALUES.omin• <u>NVERTEX</u> \leq VALUES.omax

Graph model

Since we want to express one unary constraint for each value we use the “For all items of VALUES” iterator. The only difference with the graph model of the `global_cardinality_low_up` constraint is the arc constraint where we also specify that the position of the considered variable should belong to the first argument S.

Part (A) of Figure 5.635 shows the initial graphs associated with each value 3, 5 and 6 of the VALUES collection of the **Example** slot. Part (B) of Figure 5.635 shows the two corresponding final graphs respectively associated with values 3 and 6 that are both assigned to the variables of the VARIABLES collection (since value 5 is not assigned to any variable of the VARIABLES collection the final graph associated with value 5 is empty). Since we use the NVERTEX graph property, the vertices of the final graphs are stressed in bold.



Figure 5.635: Initial and final graph of the `open_global_cardinality_low_up` constraint

20060824

1903