

5.165 global_cardinality_low_up_no_loop

	DESCRIPTION	LINKS	GRAPH
Origin	Derived from <code>global_cardinality_low_up</code> and <code>tree</code> .		
Constraint	$\text{global_cardinality_low_up_no_loop} \left(\begin{array}{l} \text{MINLOOP,} \\ \text{MAXLOOP,} \\ \text{VARIABLES,} \\ \text{VALUES} \end{array} \right)$		
Synonym	<code>gcc_low_up_no_loop</code> .		
Arguments	<pre> MINLOOP : int MAXLOOP : int VARIABLES : collection(var-dvar) VALUES : collection(val-int, omin-int, omax-int) </pre>		
Restrictions	<pre> MINLOOP ≥ 0 MINLOOP ≤ MAXLOOP MAXLOOP ≤ VARIABLES required(VARIABLES, var) VALUES > 0 required(VALUES, [val, omin, omax]) distinct(VALUES, val) VALUES.omin ≥ 0 VALUES.omax ≤ VARIABLES VALUES.omin ≤ VALUES.omax </pre>		
Purpose	<p>VALUES[i].omin ($1 \leq i \leq \text{VALUES}$) is less than or equal to the number of variables VARIABLE[j].var ($j \neq i, 1 \leq j \leq \text{VARIABLES}$) that are assigned value VALUES[i].val.</p> <p>VALUES[i].omax ($1 \leq i \leq \text{VALUES}$) is greater than or equal to the number of variables VARIABLE[j].var ($j \neq i, 1 \leq j \leq \text{VARIABLES}$) that are assigned value VALUES[i].val.</p> <p>The number of assignments of the form VARIABLE[i].var = i ($i \in [1, \text{VARIABLES}]$) is greater than or equal to MINLOOP and less than or equal to MAXLOOP.</p>		
Example	$\left(\begin{array}{l} 1, 1, \langle 1, 1, 8, 6 \rangle, \\ \left\langle \begin{array}{lll} \text{val} - 1 & \text{omin} - 1 & \text{omax} - 1, \\ \text{val} - 5 & \text{omin} - 0 & \text{omax} - 0, \\ \text{val} - 6 & \text{omin} - 1 & \text{omax} - 2 \end{array} \right\rangle \end{array} \right)$		
	<p>The <code>global_cardinality_low_up_no_loop</code> constraint holds since:</p> <ul style="list-style-type: none"> • Values 1, 5 and 6 are respectively assigned to the set of variables {VARIABLES[2].var} (i.e., $\text{omin} = 1 \leq 1 \leq \text{omax} = 1$), {} (i.e., $\text{omin} = 0 \leq$ 		

$0 \leq \text{omax} = 0$) and $\{\text{VARIABLES}[4].\text{var}\}$ (i.e., $\text{omin} = 1 \leq 1 \leq \text{omax} = 2$). Note that, due to the definition of the constraint, the fact that $\text{VARIABLES}[1].\text{var}$ is assigned to 1 is not counted.

- In addition the number of assignments of the form $\text{VARIABLES}[i].\text{var} = i$ ($i \in [1, 4]$) is greater than or equal to $\text{MINLOOP} = 1$ and less than or equal to $\text{MAXLOOP} = 1$.

Typical

```
|VARIABLES| > 1
range(VARIABLES.var) > 1
|VALUES| > 1
VALUES.omin ≤ |VARIABLES|
VALUES.omax > 0
VALUES.omax < |VARIABLES|
|VARIABLES| > |VALUES|
```

Symmetries

- Items of VALUES are [permutable](#).
- VALUES.omin can be [decreased](#) to any value ≥ 0 .
- VALUES.omax can be [increased](#) to any value $\leq |VARIABLES|$.

Usage

Within the context of the [tree](#) constraint the `global_cardinality_low_up_no_loop` constraint allows to model a minimum and maximum degree constraint on each vertex of our trees.

Algorithm

The [flow](#) algorithm that handles the original `global_cardinality` constraint [342] can be adapted to the context of the `global_cardinality_low_up_no_loop` constraint. This is done by creating an extra *value* node representing the loops corresponding to the roots of the trees. The rightmost part of Figure 3.30 illustrates the corresponding flow model for the `global_cardinality_low_up_no_loop` constraint where there is a one-to-one correspondence between feasible flows in the flow model and solutions of the `global_cardinality_low_up_no_loop` constraint.

See also

generalisation: `global_cardinality_no_loop` (fixed interval replaced by variable).

implied by: `same_and_global_cardinality_low_up`.

related: `tree` (graph partitioning by a set of trees with degree restrictions).

root concept: `global_cardinality_low_up` (assignment of a variable to its position is ignored).

Keywords

constraint type: value constraint.

filtering: flow.

	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 • variables.key \neq VALUES.val
Graph property(ies)	<ul style="list-style-type: none"> • NVERTEX \geq VALUES.omin • NVERTEX \leq VALUES.omax
Arc input(s)	VARIABLES
Arc generator	SELF \mapsto collection(variables)
Arc arity	1
Arc constraint(s)	variables.var = variables.key
Graph property(ies)	<ul style="list-style-type: none"> • NARC \geq MINLOOP • NARC \leq MAXLOOP

Graph model

Since, within the context of the first graph constraint, we want to express one unary constraint for each value we use the “For all items of VALUES” iterator. Part (A) of Figure 5.356 shows the initial graphs associated with each value 1, 5 and 6 of the VALUES collection of the **Example** slot. Part (B) of Figure 5.356 shows the two corresponding final graphs respectively associated with values 1 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.

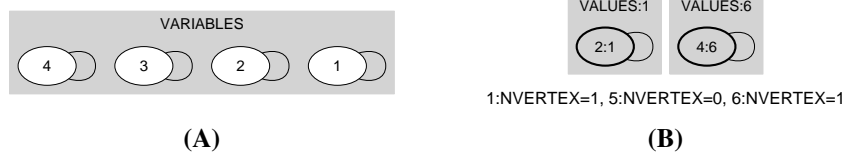


Figure 5.356: Initial and final graph of the `global_cardinality_low_up_no_loop` constraint

20051218

1261