5.165 global_cardinality_low_up_no_loop

DESCRIPTION LINKS GRAPH

Origin Derived from `global_cardinality_low_up` and `tree`.

Constraint $\texttt{global\_cardinality\_low\_up\_no\_loop}\left(\begin{array}{l}\texttt{MINLOOP},\\ \texttt{MAXLOOP},\\ \texttt{VARIABLES},\\ \texttt{VALUES}\end{array}\right)$

Synonym `gcc_low_up_no_loop`.

Arguments

MINLOOP	:	int
MAXLOOP	:	int
VARIABLES	:	collection(var−dvar)
VALUES	:	collection(val−int, omin−int, omax−int)

Restrictions

$\texttt{MINLOOP} \geq 0$
$\texttt{MINLOOP} \leq \texttt{MAXLOOP}$
$\texttt{MAXLOOP} \leq |\texttt{VARIABLES}|$
$\texttt{required}(\texttt{VARIABLES}, \texttt{var})$
$|\texttt{VALUES}| > 0$
$\texttt{required}(\texttt{VALUES}, [\texttt{val}, \texttt{omin}, \texttt{omax}])$
$\texttt{distinct}(\texttt{VALUES}, \texttt{val})$
$\texttt{VALUES.omin} \geq 0$
$\texttt{VALUES.omax} \leq |\texttt{VARIABLES}|$
$\texttt{VALUES.omin} \leq \texttt{VALUES.omax}$

Purpose

$\texttt{VALUES}[i].\texttt{omin}$ $(1 \leq i \leq |\texttt{VALUES}|)$ is less than or equal to the number of variables $\texttt{VARIABLES}[j].\texttt{var}$ $(j \neq i, 1 \leq j \leq |\texttt{VARIABLES}|)$ that are assigned value $\texttt{VALUES}[i].\texttt{val}$.
$\texttt{VALUES}[i].\texttt{omax}$ $(1 \leq i \leq |\texttt{VALUES}|)$ is greater than or equal to the number of variables $\texttt{VARIABLES}[j].\texttt{var}$ $(j \neq i, 1 \leq j \leq |\texttt{VARIABLES}|)$ that are assigned value $\texttt{VALUES}[i].\texttt{val}$.
The number of assignments of the form $\texttt{VARIABLES}[i].\texttt{var} = i$ $(i \in [1, |\texttt{VARIABLES}|])$ is greater than or equal to `MINLOOP` and less than or equal to `MAXLOOP`.

Example

$$\left(\begin{array}{l} 1, 1, \langle 1, 1, 8, 6\rangle, \\ \left\langle\begin{array}{lll} \texttt{val}-1 & \texttt{omin}-1 & \texttt{omax}-1, \\ \texttt{val}-5 & \texttt{omin}-0 & \texttt{omax}-0, \\ \texttt{val}-6 & \texttt{omin}-1 & \texttt{omax}-2 \end{array}\right\rangle \end{array}\right)$$

The `global_cardinality_low_up_no_loop` constraint holds since:

- Values 1, 5 and 6 are respectively assigned to the set of variables $\{\texttt{VARIABLES}[2].\texttt{var}\}$ (i.e., $\texttt{omin} = 1 \leq 1 \leq \texttt{omax} = 1$), $\{\}$ (i.e., $\texttt{omin} = 0 \leq$

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

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

Typical

$|\mathtt{VARIABLES}| > 1$
$\mathtt{range}(\mathtt{VARIABLES.var}) > 1$
$|\mathtt{VALUES}| > 1$
$\mathtt{VALUES.omin} \leq |\mathtt{VARIABLES}|$
$\mathtt{VALUES.omax} > 0$
$\mathtt{VALUES.omax} < |\mathtt{VARIABLES}|$
$|\mathtt{VARIABLES}| > |\mathtt{VALUES}|$

Symmetries

- Items of VALUES are permutable.
- VALUES.omin can be decreased to any value ≥ 0.
- VALUES.omax can be increased to any value $\leq |\mathtt{VARIABLES}|$.

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)	• `variables.var` $=$ `VALUES.val` • `variables.key` $\neq$ `VALUES.val`
Graph property(ies)	• **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)	• **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 5.373 shows the initial graphs associated with each value 1, 5 and 6 of the `VALUES` collection of the **Example** slot. Part (B) of Figure 5.373 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.

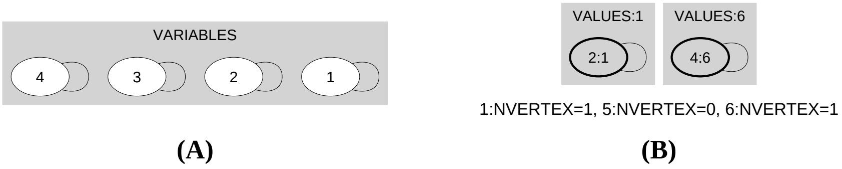

Figure 5.373: Initial and final graph of the `global_cardinality_low_up_no_loop` constraint