5.186 increasing_global_cardinality

DESCRIPTION LINKS GRAPH AUTOMATON

Origin Conjoin `global_cardinality_low_up` and `increasing`.

Constraint increasing_global_cardinality(VARIABLES, VALUES)

Synonyms increasing_global_cardinality_low_up, increasing_gcc, increasing_gcc_low_up.

Arguments

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

Restrictions

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

Purpose The variables of the collection VARIABLES are increasing. In addition, each value VALUES$[i]$.val $(1 \leq i \leq |\texttt{VALUES}|)$ should be taken by at least VALUES$[i]$.omin and at most VALUES$[i]$.omax variables of the VARIABLES collection.

Example

$$\left(\begin{array}{l}\langle 3,3,6,8\rangle,\\ \left\langle\begin{array}{lll}\texttt{val}-3 & \texttt{omin}-2 & \texttt{omax}-3,\\ \texttt{val}-5 & \texttt{omin}-0 & \texttt{omax}-1,\\ \texttt{val}-6 & \texttt{omin}-1 & \texttt{omax}-2\end{array}\right\rangle\end{array}\right)$$

The increasing_global_cardinality constraint holds since:

- The values of the collection $\langle 3,3,6,8\rangle$ are sorted in increasing order.
- Values 3, 5 and 6 are respectively used 2 $(2 \leq 2 \leq 3)$, 0 $(0 \leq 0 \leq 1)$ and 1 $(1 \leq 1 \leq 2)$ times within the collection $\langle 3,3,6,8\rangle$ and since no constraint was specified for value 8.

Typical

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

Symmetry Items of `VALUES` are permutable.

Usage This constraint can be used in order to break symmetry in the context of the following pattern. We have a matrix $\mathcal{M}$ of variables with the same constraint on each row and a `global_cardinality_low_up` constraint on each column. Beside lexicographically ordering the rows of $\mathcal{M}$ with a `lex_chain_lesseq` constraint, one can also state a `increasing_global_cardinality` on the first column of $\mathcal{M}$ in order to improve propagation on the corresponding variables.

Reformulation The `increasing_global_cardinality` constraint can be expressed in term of a conjunction of a `global_cardinality_low_up` and an `increasing` constraints. Even if we achieve arc-consistency on these two constraints this hinders propagation as shown by the following small example.

We have two variables X and Y $(X \leq Y)$, which both take their values in the set $\{2, 3\}$. In addition, assume that the minimum number of occurrences of values 0, 1 and 2 are respectively equal to 0, 1 and 1. Similarly assume that, the maximum number of occurrences of values 0, 1 and 2 are respectively equal to 1, 1 and 2. The reformulation does not reduce the domain of variables X, Y in any way, while the automaton described in the **Automaton** slot fixes X to 2 and Y to 3.

See also **implies:** `global_cardinality_low_up`, `increasing`.

related: `ordered_global_cardinality`.

Keywords **application area:** assignment.

characteristic of a constraint: automaton, automaton without counters, reified automaton constraint.

constraint network structure: Berge-acyclic constraint network.

constraint type: value constraint, order constraint.

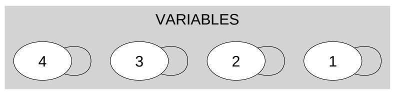
filtering: arc-consistency.

symmetry: symmetry, matrix symmetry.

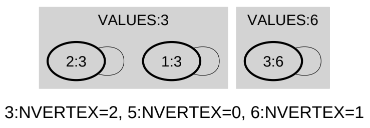
For all items of VALUES:

Arc input(s)	VARIABLES
Arc generator	$SELF \mapsto$ collection(variables)
Arc arity	1
Arc constraint(s)	variables.var = VALUES.val
Graph property(ies)	• **NVERTEX** $\geq$ VALUES.omin • **NVERTEX** $\leq$ VALUES.omax

Graph model Since we want to express one unary constraint for each value we use the "For all items of VALUES" iterator. Part (A) of Figure 5.413 shows the initial graphs associated with each value 3, 5 and 6 of the VALUES collection of the **Example** slot. Part (B) of Figure 5.413 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.



(A)



(B)

Figure 5.413: Initial and final graph of the increasing_global_cardinality constraint

Automaton

A first systematic approach for creating an automaton that only recognises the solutions to the `increasing_global_cardinality` constraint could be to:

- First, create an automaton that recognises the solutions to the `increasing` constraint.
- Second, create an automaton that recognises the solutions to the `global_cardinality_low_up` constraint.
- Third, make the product of the two previous automata and minimise the resulting automaton.

However this approach is not going to scale well in practice since the automaton associated with the `global_cardinality_low_up` constraint may have a too big size. Therefore we propose an approach where we directly construct in a single step the automaton that only recognises the solutions to the `increasing_global_cardinality` constraint. Note that we do not have any formal proof that the resulting automaton is always minimum.

Without loss of generality, we assume that:

- All items of the `VALUES` collection are sorted in increasing value on the attribute `val`.
- All the potential values of the variables of the `VARIABLES` collection are included within the set of values of the collection `VALUES` (i.e., the `val` attribute).[8]
- All values of the `VALUES` collection for which the attribute `omax` is set to 0 cannot be assigned to the variables of the `VARIABLES` collection.[9]

Before defining the states of the automaton, we first need to introduce the following notion. A value $\texttt{VALUES}[v].\texttt{val}$ is *constrained by its maximum number of occurrences* if and only if $\texttt{VALUES}[v].\texttt{omax} \leq 1 \vee \texttt{VALUES}[v].\texttt{omax} < |\texttt{VARIABLES}| - \sum_{u=1,u\neq v}^{|\texttt{VALUES}|} \texttt{VALUES}[u].\texttt{omin}$.[10] Let $\mathcal{V}$ denote the set of constrained values (i.e., their indexes within the collection `VALUES`) by their respective maximum number of occurrences.

After determining the set $\mathcal{V}$, the `omax` attribute of each potential value is normalised in the following way:

- For an unconstrained value $\texttt{VALUES}[v].\texttt{val}$ we reset $\texttt{VALUES}[v].\texttt{omax}$ to $\max(1, \texttt{VALUES}[v].\texttt{omin})$.
- For a constrained value $\texttt{VALUES}[v].\texttt{val}$ we reset $\texttt{VALUES}[v].\texttt{omax}$ to 1 if its current value is smaller than 1.

We are now in position to introduce the states of the automaton.

The $1 + \sum_{v=1,v\in\mathcal{V}}^{|\texttt{VALUES}|} \texttt{VALUES}[v].\texttt{omax} + \sum_{v=1,v\notin\mathcal{V}}^{|\texttt{VALUES}|} \texttt{VALUES}[v].\texttt{omin}$ states of the automaton that only accepts solutions to the `increasing_global_cardinality` constraint are defined in the following way:

- For the v^{th} item of the collection `VALUES` we have:
  - If $v \in \mathcal{V}$, $\texttt{VALUES}[v].\texttt{omax}$ states labelled by s_{vo} $(1 \leq o \leq \texttt{VALUES}[v].\texttt{omax})$.

[8] If this is not the case, we can include these values within the `VALUES` collection and set their minimum and maximum number of occurrences to 0 and $|\texttt{VARIABLES}| - \sum_{v=1}^{|\texttt{VALUES}|} \texttt{VALUES}[v].\texttt{omin}$.

[9] We initially remove such values from all variables of the `VARIABLES` collection.

[10] When $\texttt{VALUES}[v].\texttt{omax} \leq 1$ we cannot reduce the number of states related to value $\texttt{VALUES}[v].\texttt{val}$ and we therefore consider that we are in the constrained case.

  - If $v \notin \mathcal{V}$, $\texttt{VALUES}[v].\texttt{omin}$ states labelled by s_{vo} $(1 \leq o \leq \texttt{VALUES}[v].\texttt{omin})$.
- We have an initial state labelled by s_{00}.

Terminal states correspond to those states s_{vo} such that, both (1) o is greater than or equal to $\texttt{VALUES}[v].\texttt{omin}$, and (2) there is no value item $\texttt{VALUES}[w]$ $(w > v)$ such that $\texttt{VALUES}[w].\texttt{omin} > 0$. Transitions are defined in the following way:

- There is an arc, labelled by $\texttt{VALUES}[v].\texttt{val}$, from the initial state s_{00} to every state s_{v1} where $\texttt{VALUES}[v]$ is an item for which all values $\texttt{VALUES}[u].\texttt{val}$ strictly less than $\texttt{VALUES}[v].\texttt{val}$ verify the condition $\texttt{VALUES}[u].\texttt{omin} = 0$.
- For each value $\texttt{VALUES}[v].\texttt{val}$ constrained by its maximum number of occurrences (i.e., $v \in \mathcal{V}$), there is an arc, labelled by $\texttt{VALUES}[v].\texttt{val}$, from the state s_{vk} to the state s_{vk+1} for all k in $[1, \texttt{VALUES}[v].\texttt{omax} - 1]$.
- For each value $\texttt{VALUES}[v].\texttt{val}$ unconstrained by its maximum number of occurrences (i.e., $v \notin \mathcal{V}$), there is an arc, labelled by $\texttt{VALUES}[v].\texttt{val}$, from the state s_{vk} to the state s_{vk+1} for all k in $[1, \texttt{VALUES}[v].\texttt{omin} - 1]$. There is also a loop, labelled by $\texttt{VALUES}[v].\texttt{val}$, from state s_{vk} to the state s_{vk} for $k = \texttt{VALUES}[v].\texttt{omin}$.
- For each value $\texttt{VALUES}[v].\texttt{val}$ constrained by its maximum number of occurrences (i.e., $v \in \mathcal{V}$), there is an arc, labelled by $\texttt{VALUES}[w].\texttt{val}$, from state s_{vk} to state s_{w1} $(v < w)$ for all k in $[\texttt{VALUES}[v].\texttt{omin}, \texttt{VALUES}[v].\texttt{omax}]$ and for all w such that $\forall u \in [v+1, w-1] : \texttt{VALUES}[u].\texttt{omin} = 0$.
- For each value $\texttt{VALUES}[v].\texttt{val}$ unconstrained by its maximum number of occurrences (i.e., $v \notin \mathcal{V}$), there is an arc, labelled by $\texttt{VALUES}[w].\texttt{val}$, from state s_{vk} to state s_{w1} $(v < w)$ for $k = \texttt{VALUES}[v].\texttt{omin}$ and for all w such that $\forall u \in [v+1, w-1] :$ $\texttt{VALUES}[u].\texttt{omin} = 0$.

Figure 5.414 depicts the automaton associated with the `increasing_global_cardinality` constraint of the **Example** slot. For this purpose we assume without loss of generality that we have four decision variables that all take their potential values within interval $[3, 8]$. Consequently, values 4, 7 and 8 are first added to the items of the `VALUES` collection. Both values 3 and 6 are unconstrained by their respective maximum number of occurrences. Therefore their `omax` attributes are respectively reduced to 2 and 1. All other values, namely values 4, 5, 7 and 8, are constrained values. The `increasing_global_cardinality` constraint holds since the corresponding sequence of visited states, s_{00} s_{11} s_{12} s_{41} s_{61}, ends up in an accepting state (i.e., accepting states are denoted graphically by a double circle in the figure). Note that non initial states are first indexed by the position of an item within the `VALUES` collection, and not by the value itself (e.g., within s_{12} the 1 designates value 3). For instance state s_{11} depicts the fact that the automaton has already recognised a single occurrence of value 3, while s_{12} corresponds to the fact that the automaton has already seen at least two occurrences of value 3.[11]

[11]The at least comes from the loop on state s_{12}.

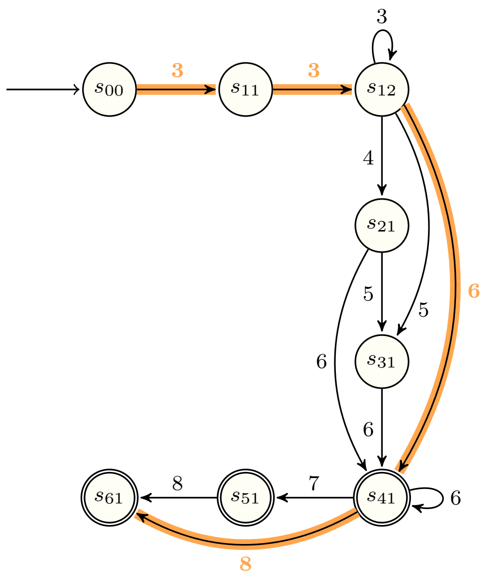


Figure 5.414: Automaton of the `increasing_global_cardinality` constraint of the **Example** slot: the path corresponding to the solution $\langle \mathbf{3}, \mathbf{3}, \mathbf{6}, \mathbf{8} \rangle$ is depicted by thick orange arcs