5.276 next_greater_element

DESCRIPTION LINKS GRAPH

Origin M. Carlsson

Constraint next_greater_element(VAR1, VAR2, VARIABLES)

Arguments

VAR1	:	dvar
VAR2	:	dvar
VARIABLES	:	collection(var−dvar)

Restrictions

VAR1 $<$ VAR2
$|$VARIABLES$| > 0$
required(VARIABLES, var)

Purpose VAR2 is the value strictly greater than VAR1 located at the smallest possible entry of the table TABLE. In addition, the variables of the collection VARIABLES are sorted in strictly increasing order.

Example $(7, 8, \langle 3, 5, 8, 9\rangle)$

The next_greater_element constraint holds since:

- VAR2 is fixed to the first value 8 strictly greater than VAR1 $= 7$,
- The var attributes of the items of the collection VARIABLES are sorted in strictly increasing order.

Typical

$|$VARIABLES$| > 1$
range(VARIABLES.var) > 1

Usage Originally introduced in [97] for modelling the fact that a nucleotide has to be consumed as soon as possible at cycle VAR2 after a given cycle VAR1.

Remark Similar to the minimum_greater_than constraint, except for the fact that the var attributes are sorted.

Reformulation Let $V_1, V_2, \ldots, V_{|\mathtt{VARIABLES}|}$ denote the variables of the collection of variables VARIABLES. By creating the extra variables M and $U_1, U_2, \ldots, U_{|\mathtt{VARIABLES}|}$, the next_greater_element constraint can be expressed in term of the following constraints:

1. $V_1 < V_2 < \cdots < V_{|\mathtt{VARIABLES}|}$
2. maximum(M, VARIABLES),
3. VAR2 $>$ VAR1,
4. VAR2 $\leq M$,
5. $V_i \leq \mathtt{VAR1} \Rightarrow U_i = M \quad (i \in [1, |\mathtt{VARIABLES}|])$,

6. $V_i > \texttt{VAR1} \Rightarrow U_i = V_i \ \ (i \in [1, |\texttt{VARIABLES}|])$,
7. minimum$(\texttt{VAR2}, \langle U_1, U_2, \ldots, U_{|\texttt{VARIABLES}|} \rangle)$.

See also

common keyword: minimum_greater_than *(order constraint)*.

implies: minimum_greater_than.

related: next_element *(allow to iterate over the values of a table)*.

Keywords

characteristic of a constraint: minimum, derived collection.

constraint type: order constraint, data constraint.

modelling: table.

Derived Collection $\mathtt{col}(\mathtt{V}-\mathtt{collection}(\mathtt{var}-\mathtt{dvar}), [\mathtt{item}(\mathtt{var}-\mathtt{VAR1})])$

Arc input(s)	VARIABLES
Arc generator	$PATH \mapsto \mathtt{collection}(\mathtt{variables1}, \mathtt{variables2})$
Arc arity	2
Arc constraint(s)	$\mathtt{variables1.var} < \mathtt{variables2.var}$
Graph property(ies)	$\mathbf{NARC} = \lvert\mathtt{VARIABLES}\rvert - 1$

Arc input(s)	V VARIABLES
Arc generator	$PRODUCT \mapsto \mathtt{collection}(\mathtt{v}, \mathtt{variables})$
Arc arity	2
Arc constraint(s)	$\mathtt{v.var} < \mathtt{variables.var}$
Graph property(ies)	$\mathbf{NARC} > 0$
Sets	$\mathsf{SUCC} \mapsto [\mathtt{source}, \mathtt{variables}]$
Constraint(s) on sets	$\mathtt{minimum}(\mathtt{VAR2}, \mathtt{variables})$

Graph model Parts (A) and (B) of Figure 5.578 respectively show the initial and final graph associated with the second graph constraint of the **Example** slot. Since we use the **NARC** graph property, the arcs of the final graph are stressed in bold.

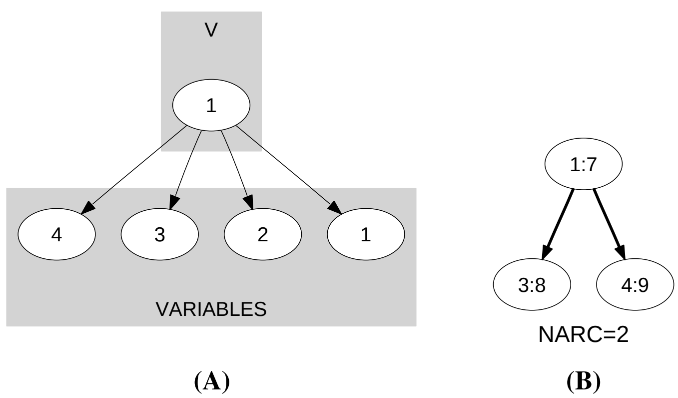

Figure 5.578: Initial and final graph of the `next_greater_element` constraint

Signature Since the first graph constraint uses the $PATH$ arc generator on the VARIABLES collection, the number of arcs of the corresponding initial graph is equal to $\lvert\mathtt{VARIABLES}\rvert - 1$. Therefore the maximum number of arcs of the final graph is equal to $\lvert\mathtt{VARIABLES}\rvert - 1$. For this reason we can rewrite $\mathbf{NARC} = \lvert\mathtt{VARIABLES}\rvert - 1$ to $\mathbf{NARC} \geq \lvert\mathtt{VARIABLES}\rvert - 1$ and simplify $\underline{\overline{\mathbf{NARC}}}$ to $\overline{\mathbf{NARC}}$.