

5.314 sort_permutation

	DESCRIPTION	LINKS	GRAPH
Origin	[409]		
Constraint	sort_permutation(FROM, PERMUTATION, TO)		
Usual name	sort		
Synonyms	extended_sortedness, sorted, sorting.		
Arguments	FROM : collection(var-dvar) PERMUTATION : collection(var-dvar) TO : collection(var-dvar)		
Restrictions	$ PERMUTATION = FROM $ $ PERMUTATION = TO $ $PERMUTATION.var \geq 1$ $PERMUTATION.var \leq PERMUTATION $ alldifferent(PERMUTATION) required(FROM, var) required(PERMUTATION, var) required(TO, var)		
Purpose	The variables of collection FROM correspond to the variables of collection TO according to the permutation PERMUTATION (i.e., $FROM[i].var = TO[PERMUTATION[i].var].var$). The variables of collection TO are also sorted in increasing order.		

Example

$$\left(\begin{array}{c}
 \text{var - 1,} \\
 \text{var - 9,} \\
 \langle \text{var - 1,} \rangle, \\
 \langle \text{var - 5,} \rangle, \\
 \text{var - 2,} \\
 \text{var - 1} \\
 \text{var - 1,} \\
 \text{var - 6,} \\
 \langle \text{var - 3,} \rangle, \\
 \langle \text{var - 5,} \rangle, \\
 \text{var - 4,} \\
 \text{var - 2} \\
 \text{var - 1,} \\
 \text{var - 1,} \\
 \langle \text{var - 1,} \rangle, \\
 \langle \text{var - 2,} \rangle, \\
 \text{var - 5,} \\
 \text{var - 9}
 \end{array} \right)$$

The sort_permutation constraint holds since:

- The first item $\text{FROM}[1].\text{var} = 1$ of collection FROM corresponds to the $\text{PERMUTATION}[1].\text{var} = 1^{\text{th}}$ item of collection TO.
 - The second item $\text{FROM}[2].\text{var} = 9$ of collection FROM corresponds to the $\text{PERMUTATION}[2].\text{var} = 6^{\text{th}}$ item of collection TO.
 - The third item $\text{FROM}[3].\text{var} = 1$ of collection FROM corresponds to the $\text{PERMUTATION}[3].\text{var} = 3^{\text{th}}$ item of collection TO.
 - The fourth item $\text{FROM}[4].\text{var} = 5$ of collection FROM corresponds to the $\text{PERMUTATION}[4].\text{var} = 5^{\text{th}}$ item of collection TO.
 - The fifth item $\text{FROM}[5].\text{var} = 2$ of collection FROM corresponds to the $\text{PERMUTATION}[5].\text{var} = 4^{\text{th}}$ item of collection TO.
 - The sixth item $\text{FROM}[6].\text{var} = 1$ of collection FROM corresponds to the $\text{PERMUTATION}[6].\text{var} = 2^{\text{th}}$ item of collection TO.
- The items of collection TO = $\langle 1, 1, 1, 2, 5, 9 \rangle$ are sorted in increasing order.

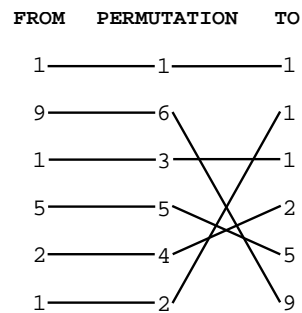


Figure 5.559: Illustration of the correspondence between the items of the FROM and the TO collections according to the permutation defined by the items of the PERMUTATION collection

Symmetry	One and the same constant can be added to the <code>var</code> attributes of all items of FROM and TO.
Remark	This constraint is referenced under the name <code>sorting</code> in SICStus Prolog .
Algorithm	[409] .
Systems	<code>sorted</code> in Gecode , <code>sorting</code> in SICStus .
See also	implies: correspondence . specialisation: sort (PERMUTATION parameter removed).
Keywords	characteristic of a constraint: sort , derived collection . combinatorial object: permutation . constraint arguments: constraint between three collections of variables .

Derived Collection

$$\text{col} \left(\begin{array}{l} \text{FROM_PERMUTATION} - \text{collection}(\text{var} - \text{dvar}, \text{ind} - \text{dvar}), \\ [\text{item}(\text{var} - \text{FROM.var}, \text{ind} - \text{PERMUTATION.var})] \end{array} \right)$$

Arc input(s)

FROM_PERMUTATION TO

Arc generator $\text{PRODUCT} \mapsto \text{collection}(\text{from_permutation}, \text{to})$ **Arc arity**

2

Arc constraint(s)

- $\text{from_permutation.var} = \text{to.var}$
- $\text{from_permutation.ind} = \text{to.key}$

Graph property(ies) $\overline{\text{NARC}} = |\text{PERMUTATION}|$ **Arc input(s)**

TO

Arc generator $\text{PATH} \mapsto \text{collection}(\text{to1}, \text{to2})$ **Arc arity**

2

Arc constraint(s) $\text{to1.var} \leq \text{to2.var}$ **Graph property(ies)** $\overline{\text{NARC}} = |\text{TO}| - 1$ **Graph model**

Parts (A) and (B) of Figure 5.560 respectively show the initial and final graph associated with the first graph constraint of the **Example** slot. In both graphs the source vertices correspond to the items of the derived collection FROM_PERMUTATION, while the sink vertices correspond to the items of the TO collection. Since the first graph constraint uses the $\overline{\text{NARC}}$ graph property, the arcs of its final graph are stressed in bold. The sort_permutation constraint holds since:

- The first graph constraint holds since its final graph contains exactly PERMUTATION arcs.
- Finally the second graph constraint holds also since its corresponding final graph contains exactly $|\text{PERMUTATION} - 1|$ arcs: all the inequalities constraints between consecutive variables of TO holds.

Signature

Consider the first graph constraint where we use the PRODUCT arc generator. Since all the key attributes of the TO collection are distinct, and because of the second condition $\text{from_permutation.ind} = \text{to.key}$ of the arc constraint, each vertex of the final graph has at most one successor. Therefore the maximum number of arcs of the final graph is equal to $|\text{PERMUTATION}|$. So we can rewrite the graph property $\overline{\text{NARC}} = |\text{PERMUTATION}|$ to $\overline{\text{NARC}} \geq |\text{PERMUTATION}|$ and simplify $\overline{\text{NARC}}$ to $\overline{\text{NARC}}$.

Consider now the second graph constraint. Since we use the PATH arc generator with an arity of 2 on the TO collection, the maximum number of arcs of the corresponding final graph is equal to $|\text{TO}| - 1$. Therefore we can rewrite $\overline{\text{NARC}} = |\text{TO}| - 1$ to $\overline{\text{NARC}} \geq |\text{TO}| - 1$ and simplify $\overline{\text{NARC}}$ to $\overline{\text{NARC}}$.

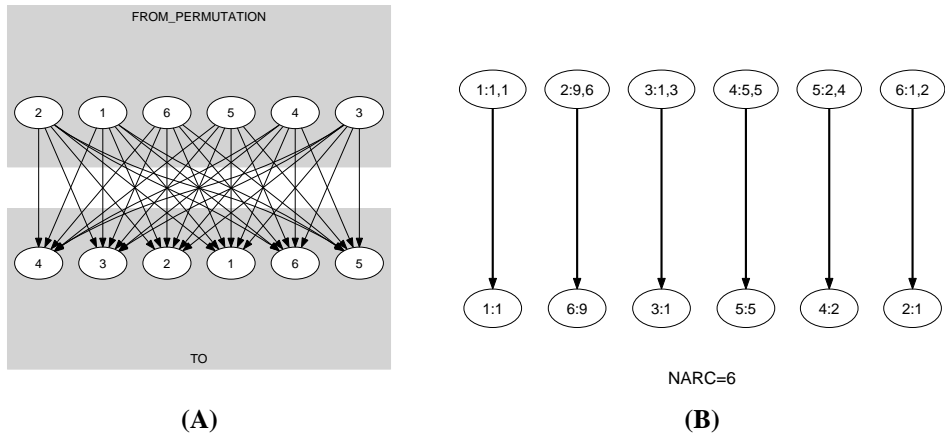


Figure 5.560: Initial and final graph of the sort_permutation constraint