## 5.343 two_orth_do_not_overlap

DESCRIPTION LINKS GRAPH AUTOMATON

**Origin** Used for defining diffn.

**Constraint** two_orth_do_not_overlap(ORTHOTOPE1, ORTHOTOPE2)

**Type**
```
ORTHOTOPE : collection(ori−dvar, siz−dvar, end−dvar)
```

**Arguments**

| | | |
|---|---|---|
| ORTHOTOPE1 | : | ORTHOTOPE |
| ORTHOTOPE2 | : | ORTHOTOPE |

**Restrictions**
$|\texttt{ORTHOTOPE}| > 0$
require_at_least$(2, \texttt{ORTHOTOPE}, [\texttt{ori}, \texttt{siz}, \texttt{end}])$
$\texttt{ORTHOTOPE.siz} \geq 0$
$\texttt{ORTHOTOPE.ori} \leq \texttt{ORTHOTOPE.end}$
$|\texttt{ORTHOTOPE1}| = |\texttt{ORTHOTOPE2}|$
orth_link_ori_siz_end(ORTHOTOPE1)
orth_link_ori_siz_end(ORTHOTOPE2)

**Purpose** For two orthotopes $O_1$ and $O_2$ enforce that there exists at least one dimension $i$ such that the projections on $i$ of $O_1$ and $O_2$ do not overlap.

**Example**
$$\left( \begin{array}{l} \langle \texttt{ori}-2\ \texttt{siz}-2\ \texttt{end}-4, \texttt{ori}-1\ \texttt{siz}-3\ \texttt{end}-4\rangle, \\ \langle \texttt{ori}-4\ \texttt{siz}-4\ \texttt{end}-8, \texttt{ori}-3\ \texttt{siz}-3\ \texttt{end}-6\rangle \end{array} \right)$$

Figure 5.616 represents the respective position of the two rectangles of the example. The coordinates of the leftmost lowest corner of each rectangle are stressed in bold. The two_orth_do_not_overlap constraint holds since the two rectangles do not overlap.

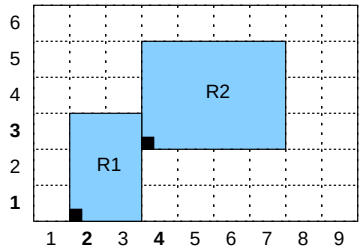


Figure 5.616: The two rectangles of the example

**Symmetries**
- Arguments are permutable w.r.t. permutation (ORTHOTOPE1, ORTHOTOPE2).
- Items of ORTHOTOPE1 and ORTHOTOPE2 are permutable (*same permutation used*).
- ORTHOTOPE1.siz can be decreased to any value $\geq 0$.
- ORTHOTOPE2.siz can be decreased to any value $\geq 0$.

**Used in** diffn.

**See also** **implied by:** two_orth_are_in_contact.

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

**constraint network structure:** Berge-acyclic constraint network.

**filtering:** arc-consistency, constructive disjunction.

**final graph structure:** bipartite, no loop.

**geometry:** geometrical constraint, non-overlapping, orthotope.

| | |
|---|---|
| **Arc input(s)** | ORTHOTOPE1 ORTHOTOPE2 |
| **Arc generator** | $SYMMETRIC\_PRODUCT(=) \mapsto$ collection(orthotope1, orthotope2) |
| **Arc arity** | 2 |
| **Arc constraint(s)** | orthotope1.end $\leq$ orthotope2.ori $\vee$ orthotope1.siz $= 0$ |
| **Graph property(ies)** | **NARC** $\geq 1$ |
| **Graph class** | • BIPARTITE<br>• NO_LOOP |

**Graph model** We build an initial graph where each arc corresponds to the fact that, either the projection of an orthotope on a given dimension is empty, either it is located before the projection in the same dimension of the other orthotope. Finally we ask that at least one arc constraint remains in the final graph.

Parts (A) and (B) of Figure 5.617 respectively show the initial and final graph associated with the **Example** slot. Since we use the **NARC** graph property, the unique arc of the final graph is stressed in bold. It corresponds to the fact that the projection in dimension 1 of the first orthotope is located before the projection in dimension 1 of the second orthotope. Therefore the two orthotopes do not overlap.

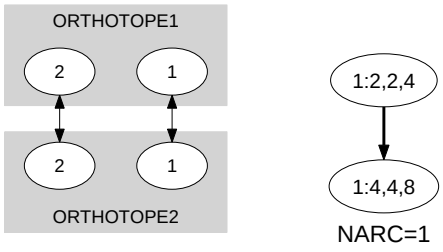


Figure 5.617: Initial and final graph of the two_orth_do_not_overlap constraint

**Automaton**

Figure 5.618 depicts the automaton associated with the `two_orth_do_not_overlap` constraint. Let $\texttt{ORI1}_i$, $\texttt{SIZ1}_i$ and $\texttt{END1}_i$ respectively be the `ori`, the `siz` and the `end` attributes of the $i^{th}$ item of the `ORTHOTOPE1` collection. Let $\texttt{ORI2}_i$, $\texttt{SIZ2}_i$ and $\texttt{END2}_i$ respectively be the `ori`, the `siz` and the `end` attributes of the $i^{th}$ item of the `ORTHOTOPE2` collection. To each sextuple $(\texttt{ORI1}_i, \texttt{SIZ1}_i, \texttt{END1}_i, \texttt{ORI2}_i, \texttt{SIZ2}_i, \texttt{END2}_i)$ corresponds a 0-1 signature variable $\texttt{S}_i$ as well as the following signature constraint: $((\texttt{SIZ1}_i > 0) \wedge (\texttt{SIZ2}_i > 0) \wedge (\texttt{END1}_i > \texttt{ORI2}_i) \wedge (\texttt{END2}_i > \texttt{ORI1}_i)) \Leftrightarrow \texttt{S}_i$.

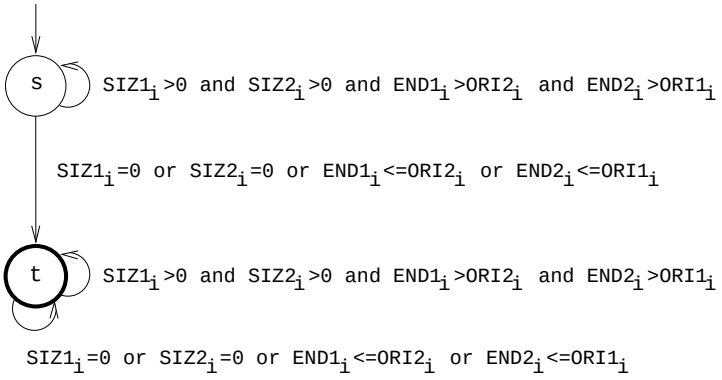


Figure 5.618: Automaton of the `two_orth_do_not_overlap` constraint

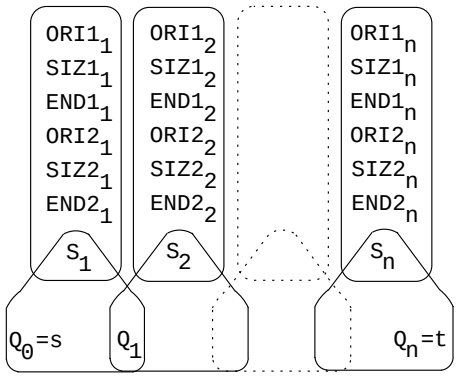


Figure 5.619: Hypergraph of the reformulation corresponding to the automaton of the `two_orth_do_not_overlap` constraint