5.408 two_orth_are_in_contact

DESCRIPTION LINKS GRAPH AUTOMATON

Origin [358], used for defining orths_are_connected.

Constraint two_orth_are_in_contact(ORTHOTOPE1, ORTHOTOPE2)

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

Arguments

ORTHOTOPE1 : ORTHOTOPE
ORTHOTOPE2 : ORTHOTOPE

Restrictions

|ORTHOTOPE| > 0
require_at_least(2, ORTHOTOPE, [ori, siz, end])
ORTHOTOPE.siz > 0
ORTHOTOPE.ori ≤ ORTHOTOPE.end
|ORTHOTOPE1| = |ORTHOTOPE2|
orth_link_ori_siz_end(ORTHOTOPE1)
orth_link_ori_siz_end(ORTHOTOPE2)

Purpose

Enforce the following conditions on two orthotopes O_1 and O_2:

- For all dimensions i, except one dimension, the projections of O_1 and O_2 onto i have a non-empty intersection.
- For all dimensions i, the distance between the projections of O_1 and O_2 onto i is equal to 0.

Example

$$\left(\begin{array}{l}\langle \texttt{ori}-1\ \texttt{siz}-3\ \texttt{end}-4, \texttt{ori}-5\ \texttt{siz}-2\ \texttt{end}-7\rangle, \\ \langle \texttt{ori}-3\ \texttt{siz}-2\ \texttt{end}-5, \texttt{ori}-2\ \texttt{siz}-3\ \texttt{end}-5\rangle\end{array}\right)$$

Figure 5.772 shows the two rectangles of the example. The two_orth_are_in_contact constraint holds since the two rectangles are in contact: the contact is depicted by a pink line-segment.

Typical |ORTHOTOPE| > 1

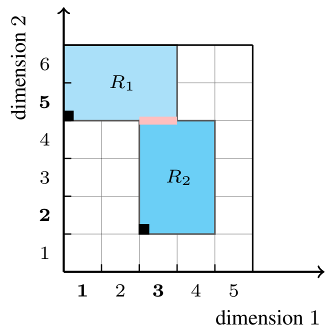
Symmetries

- Arguments are permutable w.r.t. permutation (ORTHOTOPE1, ORTHOTOPE2).
- Items of ORTHOTOPE1 and ORTHOTOPE2 are permutable (*same permutation used*).

Used in orths_are_connected.

See also **implies:** two_orth_do_not_overlap.

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



R_1: $\langle \texttt{ori}-1\ \texttt{siz}-3\ \texttt{end}-4, \texttt{ori}-5\ \texttt{siz}-2\ \texttt{end}-7\rangle$
R_2: $\langle \texttt{ori}-3\ \texttt{siz}-2\ \texttt{end}-5, \texttt{ori}-2\ \texttt{siz}-3\ \texttt{end}-5\rangle$

`ORTHOTOPES` (rectangles)

Figure 5.772: The two rectangles that are in contact of the **Example** slot where the contact is shown in pink

constraint network structure: Berge-acyclic constraint network.

constraint type: logic.

filtering: arc-consistency.

geometry: geometrical constraint, touch, contact, non-overlapping, orthotope.

Arc input(s)	ORTHOTOPE1 ORTHOTOPE2
Arc generator	$PRODUCT(=) \mapsto$ collection(orthotope1, orthotope2)
Arc arity	2
Arc constraint(s)	• orthotope1.end > orthotope2.ori • orthotope2.end > orthotope1.ori
Graph property(ies)	$\mathbf{NARC} = \lvert\texttt{ORTHOTOPE1}\rvert - 1$

Arc input(s)	ORTHOTOPE1 ORTHOTOPE2
Arc generator	$PRODUCT(=) \mapsto$ collection(orthotope1, orthotope2)
Arc arity	2
Arc constraint(s)	$\texttt{max}\left(0, \begin{array}{l}\texttt{max(orthotope1.ori, orthotope2.ori)}-\\ \texttt{min(orthotope1.end, orthotope2.end)}\end{array}\right) = 0$
Graph property(ies)	$\mathbf{NARC} = \lvert\texttt{ORTHOTOPE1}\rvert$

Graph model Parts (A) and (B) of Figure 5.773 respectively show the initial and final graph associated with the first graph constraint of 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 onto dimension 1 of the two rectangles of the example overlap.

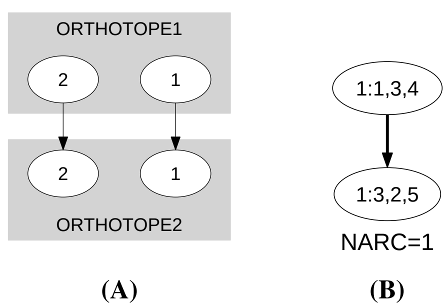


Figure 5.773: Initial and final graph of the two_orth_are_in_contact constraint

Signature Consider the second graph constraint. Since we use the arc generator $PRODUCT(=)$ on the collections ORTHOTOPE1 and ORTHOTOPE2, and because of the restriction $\lvert\texttt{ORTHOTOPE1}\rvert = \lvert\texttt{ORTHOTOPE2}\rvert$, the maximum number of arcs of the corresponding final graph is equal to $\lvert\texttt{ORTHOTOPE1}\rvert$. Therefore we can rewrite the graph property $\mathbf{NARC} = \lvert\texttt{ORTHOTOPE1}\rvert$ to $\mathbf{NARC} \geq \lvert\texttt{ORTHOTOPE1}\rvert$ and simplify $\overline{\mathbf{NARC}}$ to $\overline{\mathbf{NARC}}$.

Automaton Figure 5.774 depicts the automaton associated with the `two_orth_are_in_contact` 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 signature variable S_i, which takes its value in $\{0, 1, 2\}$, as well as the following signature constraint:

$$((\texttt{SIZ1}_i > 0) \wedge (\texttt{SIZ2}_i > 0) \wedge (\texttt{END1}_i > \texttt{ORI2}_i) \wedge (\texttt{END2}_i > \texttt{ORI1}_i)) \Leftrightarrow S_i = 0$$

$$((\texttt{SIZ1}_i > 0) \wedge (\texttt{SIZ2}_i > 0) \wedge (\texttt{END1}_i = \texttt{ORI2}_i \vee \texttt{END2}_i = \texttt{ORI1}_i)) \Leftrightarrow S_i = 1.$$

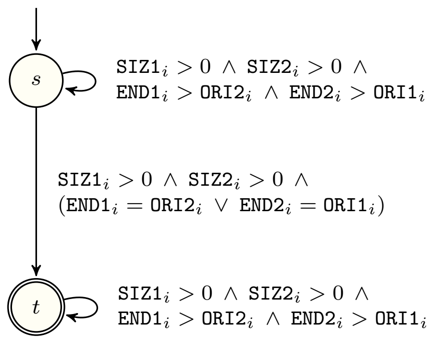


Figure 5.774: Automaton of the `two_orth_are_in_contact` constraint

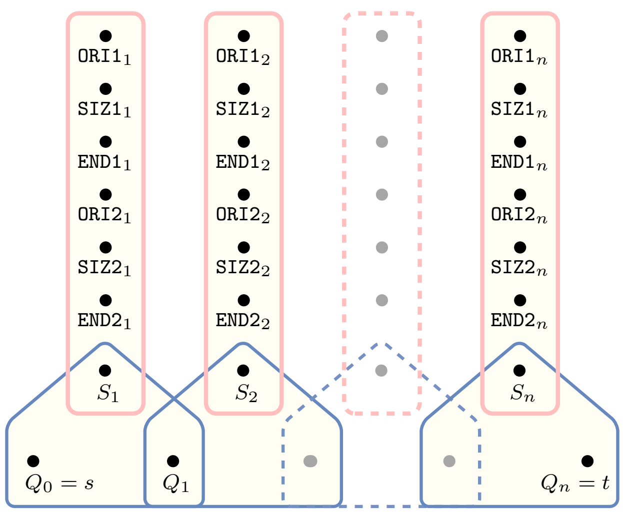


Figure 5.775: Hypergraph of the reformulation corresponding to the automaton of the `two_orth_are_in_contact` constraint