

5.124 disjoint_sboxes

	DESCRIPTION	LINKS	LOGIC
Origin	Geometry, derived from [338]		
Constraint	<code>disjoint_sboxes(K, DIMS, OBJECTS, SBOXES)</code>		
Synonym	<code>disjoint.</code>		
Types	VARIABLES : <code>collection(v-dvar)</code> INTEGERS : <code>collection(v-int)</code> POSITIVES : <code>collection(v-int)</code>		
Arguments	K : <code>int</code> DIMS : <code>sint</code> OBJECTS : <code>collection(oid-int, sid-dvar, x - VARIABLES)</code> SBOXES : <code>collection(sid-int, t - INTEGERS, l - POSITIVES)</code>		
Restrictions	$ VARIABLES \geq 1$ $ INTEGERS \geq 1$ $ POSITIVES \geq 1$ <code>required(VARIABLES, v)</code> $ VARIABLES = K$ <code>required(INTEGERS, v)</code> $ INTEGERS = K$ <code>required(POSITIVES, v)</code> $ POSITIVES = K$ $POSITIVES.v > 0$ $K > 0$ $DIMS \geq 0$ $DIMS < K$ <code>increasing_seq(OBJECTS, [oid])</code> <code>required(OBJECTS, [oid, sid, x])</code> $OBJECTS.oid \geq 1$ $OBJECTS.oid \leq OBJECTS $ $OBJECTS.sid \geq 1$ $OBJECTS.sid \leq SBOXES $ $ SBOXES \geq 1$ <code>required(SBOXES, [sid, t, l])</code> $SBOXES.sid \geq 1$ $SBOXES.sid \leq SBOXES $ <code>do_not_overlap(SBOXES)</code>		

Purpose

Holds if, for each pair of objects (O_i, O_j) , $i \neq j$, O_i and O_j are disjoint with respect to a set of dimensions depicted by DIMS. O_i and O_j are objects that take a shape among a set of shapes. Each *shape* is defined as a finite set of shifted boxes, where each shifted box is described by a box in a K-dimensional space at a given offset (from the origin of the shape) with given sizes. More precisely, a *shifted box* is an entity defined by its shape id *sid*, shift offset *t*, and sizes *l*. Then, a shape is defined as the union of shifted boxes sharing the same shape id. An *object* is an entity defined by its unique object identifier *oid*, shape id *sid* and origin *x*.

Two objects O_i and object O_j are *disjoint* with respect to a set of dimensions depicted by DIMS if and only if for all shifted box s_i associated with O_i and for all shifted box s_j associated with O_j there exists at least one dimension $d \in \text{DIMS}$ such that (1) the origin of s_i in dimension d is strictly greater than the end of s_j in dimension d , or (2) the origin of s_j in dimension d is strictly greater than the end of s_i in dimension d .

Example

$$\left(\begin{array}{l} 2, \{0, 1\}, \\ \left\langle \begin{array}{lll} \text{oid} - 1 & \text{sid} - 1 & \mathbf{x} - \langle 1, 1 \rangle, \\ \text{oid} - 2 & \text{sid} - 2 & \mathbf{x} - \langle 4, 1 \rangle, \\ \text{oid} - 3 & \text{sid} - 4 & \mathbf{x} - \langle 2, 4 \rangle \end{array} \right\rangle, \\ \begin{array}{lll} \text{sid} - 1 & \mathbf{t} - \langle 0, 0 \rangle & \mathbf{l} - \langle 1, 2 \rangle, \\ \text{sid} - 2 & \mathbf{t} - \langle 0, 0 \rangle & \mathbf{l} - \langle 1, 1 \rangle, \\ \text{sid} - 2 & \mathbf{t} - \langle 1, 0 \rangle & \mathbf{l} - \langle 1, 3 \rangle, \end{array} \\ \left\langle \begin{array}{lll} \text{sid} - 2 & \mathbf{t} - \langle 0, 2 \rangle & \mathbf{l} - \langle 1, 1 \rangle, \\ \text{sid} - 3 & \mathbf{t} - \langle 0, 0 \rangle & \mathbf{l} - \langle 3, 1 \rangle, \\ \text{sid} - 3 & \mathbf{t} - \langle 0, 1 \rangle & \mathbf{l} - \langle 1, 1 \rangle, \\ \text{sid} - 3 & \mathbf{t} - \langle 2, 1 \rangle & \mathbf{l} - \langle 1, 1 \rangle, \\ \text{sid} - 4 & \mathbf{t} - \langle 0, 0 \rangle & \mathbf{l} - \langle 1, 1 \rangle \end{array} \right\rangle \end{array} \right)$$

Figure 5.295 shows the objects of the example. Since these objects are pairwise disjoint the `disjoint_sboxes` constraint holds.

Typical

`|OBJECTS| > 1`

Symmetries

- Items of OBJECTS are [permutable](#).
- Items of SBOXES are [permutable](#).
- SBOXES.l.v can be [decreased](#) to any value ≥ 1 .

Arg. properties

[Suffix-contractible](#) wrt. OBJECTS.

Remark

One of the eight relations of the [Region Connection Calculus](#) [338]. Unlike the `non_overlap_sboxes` constraint, which just prevents objects from overlapping, the `disjoint_sboxes` constraint in addition enforces that borders and corners of objects are not directly in contact.

See also

common keyword: [contains_sboxes](#), [coveredby_sboxes](#), [covers_sboxes](#), [equal_sboxes](#), [inside_sboxes](#), [meet_sboxes](#) (*rcc8*), [non_overlap_sboxes](#) (*geometrical constraint, logic*), [overlap_sboxes](#) (*rcc8*).
implies: `non_overlap_sboxes`.

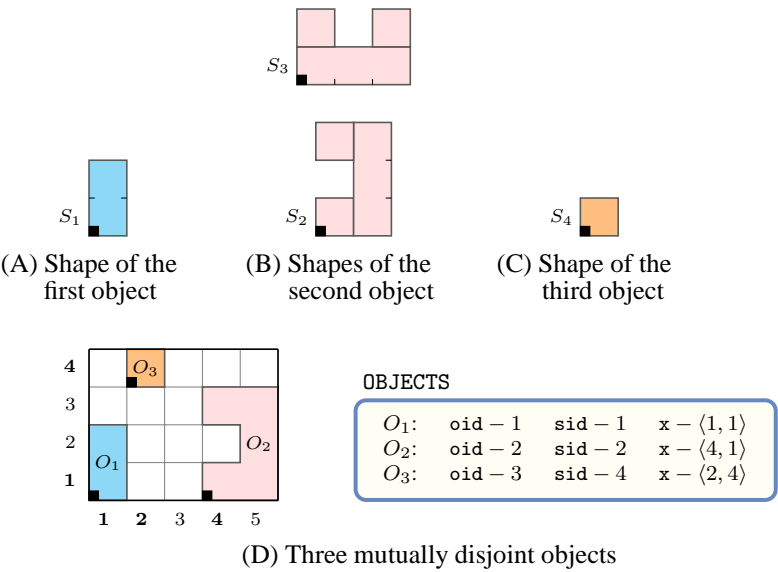


Figure 5.295: (D) the three mutually disjoint objects O_1 , O_2 , O_3 of the **Example** slot respectively assigned shapes S_1 , S_2 , S_4 ; (A), (B), (C) shapes S_1 , S_2 , S_3 and S_4 are respectively made up from 1, 3, 3 and 1 disjoint shifted box.

Keywords

constraint type: logic.

geometry: geometrical constraint, rcc8.

miscellaneous: obscure.

Logic

- $\text{origin}(O1, S1, D) \stackrel{\text{def}}{=} O1.x(D) + S1.t(D)$
- $\text{end}(O1, S1, D) \stackrel{\text{def}}{=} O1.x(D) + S1.t(D) + S1.l(D)$
- $\text{disjoint_sboxes}(\text{Dims}, O1, S1, O2, S2) \stackrel{\text{def}}{=} \begin{aligned} &\exists D \in \text{Dims} \\ &\bigvee \left(\begin{array}{l} \text{origin}(O1, S1, D) > \\ \text{end}(O2, S2, D) \\ \text{origin}(O2, S2, D) > \\ \text{end}(O1, S1, D) \end{array} \right) \end{aligned}$
- $\text{disjoint_objects}(\text{Dims}, O1, O2) \stackrel{\text{def}}{=} \begin{aligned} &\forall S1 \in \text{sboxes}([O1.\text{sid}]) \\ &\forall S2 \in \text{sboxes}([O2.\text{sid}]) \\ &\text{disjoint_sboxes} \left(\begin{array}{l} \text{Dims}, \\ O1, \\ S1, \\ O2, \\ S2 \end{array} \right) \end{aligned}$
- $\text{all_disjoint}(\text{Dims}, \text{OIDS}) \stackrel{\text{def}}{=} \begin{aligned} &\forall O1 \in \text{objects}(\text{OIDS}) \\ &\forall O2 \in \text{objects}(\text{OIDS}) \\ &O1.\text{oid} < \Rightarrow \\ &O2.\text{oid} \\ &\text{disjoint_objects} \left(\begin{array}{l} \text{Dims}, \\ O1, \\ O2 \end{array} \right) \end{aligned}$
- $\text{all_disjoint}(\text{DIMENSIONS}, \text{OIDS})$