

5.75 contains_sboxes

	DESCRIPTION	LINKS	LOGIC
Origin	Geometry, derived from [305]		
Constraint	contains_sboxes(K, DIMS, OBJECTS, SBOXES)		
Synonym	contains.		
Types	VARIABLES : collection(v-dvar) INTEGERS : collection(v-int) POSITIVES : collection(v-int)		
Arguments	K : int DIMS : sint OBJECTS : collection(oid-int, sid-int, x - VARIABLES) SBOXES : collection(sid-int, t - INTEGERS, l - POSITIVES)		
Restrictions	<pre> required(VARIABLES, v) VARIABLES = K required(INTEGERS, v) INTEGERS = K required(POSITIVES, v) POSITIVES = K POSITIVES.v > 0 K > 0 DIMS ≥ 0 DIMS < K required(OBJECTS, [oid, sid, x]) OBJECTS.oid ≥ 1 OBJECTS.oid ≤ OBJECTS OBJECTS.sid ≥ 1 OBJECTS.sid ≤ SBOXES required(SBOXES, [sid, t, l]) SBOXES.sid ≥ 1 SBOXES.sid ≤ SBOXES </pre>		

Purpose

Holds if, for each pair of objects (O_i, O_j) , $i < j$, O_i contains O_j 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 τ , 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 .

An object O_i contains an object O_j with respect to a set of dimensions depicted by DIMS if and only if, for all shifted boxes s_j associated with O_j , there exists a shifted box s_i of O_i such that s_i contains s_j . A shifted box s_i contains a shifted box s_j if and only if, for all dimensions $d \in \text{DIMS}$, (1) the start of s_i in dimension d is strictly less than the start of s_j in dimension d and (2) the end of s_j in dimension d is strictly less than the end of s_i in dimension d .

Example

$$\left(\begin{array}{l} 2, \{0, 1\}, \\ \left\langle \begin{array}{lll} \text{oid} - 1 & \text{sid} - 1 & x - \langle 1, 1 \rangle, \\ \text{oid} - 2 & \text{sid} - 2 & x - \langle 2, 2 \rangle, \\ \text{oid} - 3 & \text{sid} - 3 & x - \langle 3, 3 \rangle \end{array} \right\rangle, \\ \left\langle \begin{array}{lll} \text{sid} - 1 & \tau - \langle 0, 0 \rangle & l - \langle 5, 5 \rangle, \\ \text{sid} - 2 & \tau - \langle 0, 0 \rangle & l - \langle 3, 3 \rangle, \\ \text{sid} - 3 & \tau - \langle 0, 0 \rangle & l - \langle 1, 1 \rangle \end{array} \right\rangle \end{array} \right)$$

Figure 5.152 shows the objects of the example. Since O_1 contains both O_2 and O_3 , and since O_2 contains O_3 , the contains_sboxes constraint holds.

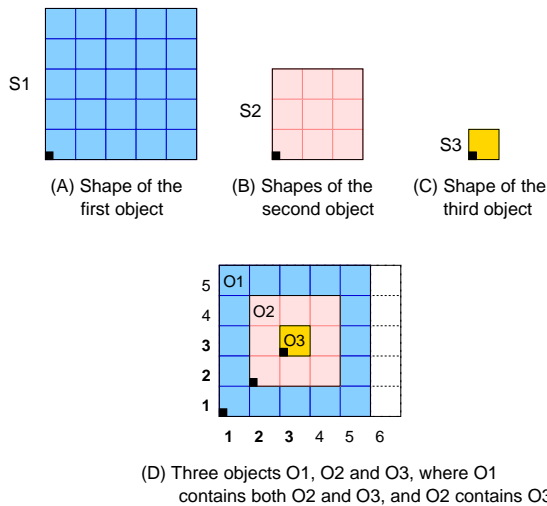


Figure 5.152: The three objects of the example

Typical

$|\text{OBJECTS}| > 1$

Symmetries

- Items of SBOXES are [permutable](#).
- Items of OBJECTS.x, SBOXES.t and SBOXES.l are [permutable](#) (*same permutation used*).

Remark

One of the eight relations of the *Region Connection Calculus* [305]. The constraint `contains_sboxes` is a restriction of the original relation since it requires that each shifted box of an object is contained by one shifted box of the other object.

See also

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

Keywords

constraint type: [logic](#).
geometry: [geometrical constraint](#), [rcc8](#).

Logic

- $\text{origin}(O1, S1, D) \stackrel{\text{def}}{=} O1.x(D) + S1.t(D)$
- $\text{end}(O1, S1, D) \stackrel{\text{def}}{=} O1.x(D) + S1.t(D) + S1.l(D)$
- $\text{contains_sboxes}(\text{Dims}, O1, S1, O2, S2) \stackrel{\text{def}}{=} \forall D \in \text{Dims} \wedge \left(\begin{array}{l} \text{origin}(O1, S1, D) < \\ \text{origin}(O2, S2, D) \\ \text{end}(O2, S2, D) < \\ \text{end}(O1, S1, D) \end{array} \right)$
- $\text{contains_objects}(\text{Dims}, O1, O2) \stackrel{\text{def}}{=} \forall S1 \in \text{sboxes}([O1.\text{sid}]) \exists S2 \in \text{sboxes}([O2.\text{sid}]) \text{contains_sboxes} \left(\begin{array}{l} \text{Dims}, \\ O1, \\ S1, \\ O2, \\ S2 \end{array} \right)$
- $\text{all_contains}(\text{Dims}, \text{OIDS}) \stackrel{\text{def}}{=} \forall O1 \in \text{objects}(\text{OIDS}) \forall O2 \in \text{objects}(\text{OIDS}) O1.\text{oid} < \Rightarrow O2.\text{oid} \text{contains_objects} \left(\begin{array}{l} \text{Dims}, \\ O1, \\ O2 \end{array} \right)$
- $\text{all_contains}(\text{DIMENSIONS}, \text{OIDS})$