5.314 overlap_sboxes

DESCRIPTION LINKS LOGIC

Origin Geometry, derived from [338]

Constraint overlap_sboxes(K, DIMS, OBJECTS, SBOXES)

Synonym overlap.

Types

```
VARIABLES  : collection(v−dvar)
INTEGERS   : collection(v−int)
POSITIVES  : collection(v−int)
```

Arguments

```
K        : int
DIMS     : sint
OBJECTS  : collection(oid−int, sid−dvar, x − VARIABLES)
SBOXES   : collection(sid−int, t − INTEGERS, l − POSITIVES)
```

Restrictions

```
|VARIABLES| ≥ 1
|INTEGERS| ≥ 1
|POSITIVES| ≥ 1
required(VARIABLES, v)
|VARIABLES| = K
required(INTEGERS, v)
|INTEGERS| = K
required(POSITIVES, v)
|POSITIVES| = K
POSITIVES.v > 0
K > 0
DIMS ≥ 0
DIMS < K
increasing_seq(OBJECTS, [oid])
required(OBJECTS, [oid, sid, x])
OBJECTS.oid ≥ 1
OBJECTS.oid ≤ |OBJECTS|
OBJECTS.sid ≥ 1
OBJECTS.sid ≤ |SBOXES|
|SBOXES| ≥ 1
required(SBOXES, [sid, t, l])
SBOXES.sid ≥ 1
SBOXES.sid ≤ |SBOXES|
do_not_overlap(SBOXES)
```

Purpose

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

An object O_i *overlaps* an object O_j with respect to a set of dimensions depicted by DIMS if and only if, there exists a shifted box s_i associated with O_i and there exists a shifted box s_j associated with O_j, such that (1) there exists a dimension $d \in$ DIMS where the end of O_i in dimension d is strictly greater than the start of O_j in dimension d, and (2) the end of O_j in dimension d is strictly greater than the start of O_i in dimension d.

Example

$$\left(\begin{array}{l} 2, \{0,1\}, \\ \left\langle\begin{array}{lll} \texttt{oid}-1 & \texttt{sid}-1 & \texttt{x}-\langle 1,1\rangle, \\ \texttt{oid}-2 & \texttt{sid}-2 & \texttt{x}-\langle 3,2\rangle, \\ \texttt{oid}-3 & \texttt{sid}-3 & \texttt{x}-\langle 2,4\rangle \end{array}\right\rangle, \\ \left\langle\begin{array}{lll} \texttt{sid}-1 & \texttt{t}-\langle 0,0\rangle & \texttt{l}-\langle 4,5\rangle, \\ \texttt{sid}-2 & \texttt{t}-\langle 0,0\rangle & \texttt{l}-\langle 3,3\rangle, \\ \texttt{sid}-3 & \texttt{t}-\langle 0,0\rangle & \texttt{l}-\langle 2,1\rangle \end{array}\right\rangle \end{array}\right)$$

Figure 5.637 shows the objects of the example. Since O_1 overlaps both O_2 and O_3, and since O_2 overlaps O_3, the overlap_sboxes constraint holds.

Typical

$|\texttt{OBJECTS}| > 1$

Symmetries

- Items of OBJECTS are permutable.
- Items of SBOXES are permutable.
- Items of OBJECTS.x, SBOXES.t and SBOXES.l are permutable (*same permutation used*).
- SBOXES.l.v can be increased.

Arg. properties

Suffix-contractible wrt. OBJECTS.

Remark

One of the eight relations of the *Region Connection Calculus* [338].

See also

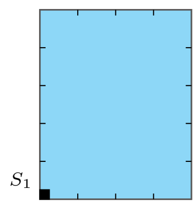
common keyword: contains_sboxes, coveredby_sboxes, covers_sboxes, disjoint_sboxes, equal_sboxes, inside_sboxes, meet_sboxes (*rcc8*), non_overlap_sboxes (*geometrical constraint*,*logic*).

Keywords

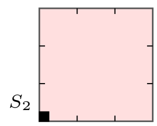
constraint type: logic.

geometry: geometrical constraint, rcc8.

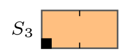
miscellaneous: obscure.



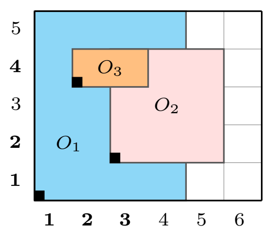
(A) Shape of the first object



(B) Shape of the second object



(C) Shape of the third object



OBJECTS

O_1:	oid − 1	sid − 1	x − $\langle 1, 1\rangle$
O_2:	oid − 2	sid − 2	x − $\langle 3, 2\rangle$
O_3:	oid − 3	sid − 3	x − $\langle 2, 4\rangle$

(D) Three objects where O_1 both overlaps O_2 and O_3 and where O_2 overlaps O_3

Figure 5.637: (D) the three mutually overlapping objects O_1, O_2, O_3 of the **Example** slot respectively assigned shapes S_1, S_2, S_3; (A), (B), (C) shapes S_1, S_2 and S_3 are made up from a single shifted box.

Logic

- $\texttt{origin}(\texttt{O1},\texttt{S1},\texttt{D}) \stackrel{\mathrm{def}}{=} \texttt{O1.x}(\texttt{D}) + \texttt{S1.t}(\texttt{D})$
- $\texttt{end}(\texttt{O1},\texttt{S1},\texttt{D}) \stackrel{\mathrm{def}}{=} \texttt{O1.x}(\texttt{D}) + \texttt{S1.t}(\texttt{D}) + \texttt{S1.l}(\texttt{D})$
- $\texttt{overlap\_sboxes}(\texttt{Dims},\texttt{O1},\texttt{S1},\texttt{O2},\texttt{S2}) \stackrel{\mathrm{def}}{=}$
  $\forall \texttt{D} \in \texttt{Dims}$
  $$\bigwedge \left(\begin{array}{l} \texttt{end}(\texttt{O1},\texttt{S1},\texttt{D}) > \\ \texttt{origin}(\texttt{O2},\texttt{S2},\texttt{D}) \end{array}, \begin{array}{l} \texttt{end}(\texttt{O2},\texttt{S2},\texttt{D}) > \\ \texttt{origin}(\texttt{O1},\texttt{S1},\texttt{D}) \end{array} \right)$$
- $\texttt{overlap\_objects}(\texttt{Dims},\texttt{O1},\texttt{O2}) \stackrel{\mathrm{def}}{=}$
  $\forall \texttt{S1} \in \texttt{sboxes}([\texttt{O1.sid}])$
  $\exists \texttt{S2} \in \texttt{sboxes}\left(\left[\texttt{O2.sid} \right] \right)$
  $$\texttt{overlap\_sboxes}\left(\begin{array}{l} \texttt{Dims}, \\ \texttt{O1}, \\ \texttt{S1}, \\ \texttt{O2}, \\ \texttt{S2} \end{array} \right)$$
- $\texttt{all\_overlap}(\texttt{Dims},\texttt{OIDS}) \stackrel{\mathrm{def}}{=}$
  $\forall \texttt{O1} \in \texttt{objects}(\texttt{OIDS})$
  $\forall \texttt{O2} \in \texttt{objects}(\texttt{OIDS})$
  $$\begin{array}{l} \texttt{O1.oid} < \\ \texttt{O2.oid} \end{array} \Rightarrow \texttt{overlap\_objects}\left(\begin{array}{l} \texttt{Dims}, \\ \texttt{O1}, \\ \texttt{O2} \end{array} \right)$$
- $\texttt{all\_overlap}(\texttt{DIMENSIONS},\texttt{OIDS})$