5.94 covers_sboxes

DESCRIPTION LINKS LOGIC

Origin Geometry, derived from [338]

Constraint covers_sboxes(K, DIMS, OBJECTS, SBOXES)

Synonym covers.

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 covers 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 *covers* an object O_j with respect to a set of dimensions depicted by `DIMS` if and only if, for all shifted box s_j of O_j, there exists a shifted box s_i of O_i such that:

- For all dimensions $d \in$ `DIMS`, (1) the start of s_i in dimension d is less than or equal to the start of s_j in dimension d, and (2) the end of s_j in dimension d is less than or equal to the end of s_i in dimension d.
- There exists a dimension d where, (1) the start of s_i in dimension d coincide with the start of s_j in dimension d, or (2) the end of s_i in dimension d coincide with the end of s_j in dimension d.

Example

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

Figure 5.210 shows the objects of the example. Since O_1 covers both O_2 and O_3, and since O_2 covers O_3, the `covers_sboxes` constraint holds.

Typical

$|$`OBJECTS`$| > 1$

Symmetries

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

Arg. properties

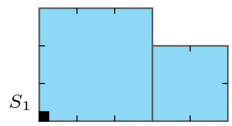
Suffix-contractible wrt. `OBJECTS`.

Remark

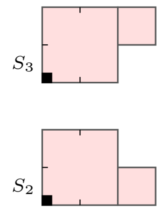
One of the eight relations of the *Region Connection Calculus* [338]. The constraint `covers_sboxes` is a relaxation of the original relation since it requires that each shifted box of an object is covered by one shifted box of the other object.

See also

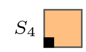
common keyword: `contains_sboxes`, `coveredby_sboxes`, `disjoint_sboxes`, `equal_sboxes`, `inside_sboxes`, `meet_sboxes` (*rcc8*), `non_overlap_sboxes` (*geometrical constraint*, *logic*), `overlap_sboxes` (*rcc8*).



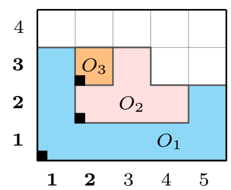
(A) Shape of the first object



(B) Shapes 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 − $\langle 2, 2\rangle$
O_3:	oid − 3	sid − 4	x − $\langle 2, 3\rangle$

(D) Three objects O_1, O_2, O_3, where O_1 covers both O_2 and O_3 and where O_2 covers O_3

Figure 5.210: (D) the three 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 2, 2, 2 and 1 single shifted box.

Keywords

constraint type: logic.

geometry: geometrical constraint, rcc8.

miscellaneous: obscure.

Logic

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