5.99 cumulative_two_d

DESCRIPTION LINKS

Origin

Inspired by cumulative and diffn.

Constraint

cumulative_two_d(RECTANGLES, LIMIT)

Arguments

$$\begin{array}{lll} \texttt{RECTANGLES} & : & \texttt{collection}\left(\begin{array}{l} \texttt{start1}-\texttt{dvar}, \\ \texttt{size1}-\texttt{dvar}, \\ \texttt{last1}-\texttt{dvar}, \\ \texttt{start2}-\texttt{dvar}, \\ \texttt{size2}-\texttt{dvar}, \\ \texttt{last2}-\texttt{dvar}, \\ \texttt{height}-\texttt{dvar} \end{array}\right) \\ \texttt{LIMIT} & : & \texttt{int} \end{array}$$

Restrictions

require_at_least(2, RECTANGLES, [start1, size1, last1])
require_at_least(2, RECTANGLES, [start2, size2, last2])
required(RECTANGLES, height)
RECTANGLES.size1 ≥ 0
RECTANGLES.size2 ≥ 0
RECTANGLES.height ≥ 0
LIMIT ≥ 0

Purpose

Consider a set $\mathcal{R}$ of rectangles described by the RECTANGLES collection. Enforces that at each point of the plane, the cumulated height of the set of rectangles that overlap that point, does not exceed a given limit.

Example

$$\left(\left\langle\begin{array}{lllllll} \texttt{start1}-1 & \texttt{size1}-4 & \texttt{last1}-4 & \texttt{start2}-3 & \texttt{size2}-3 & \texttt{last2}-5 & \texttt{height}-4, \\ \texttt{start1}-3 & \texttt{size1}-2 & \texttt{last1}-4 & \texttt{start2}-1 & \texttt{size2}-2 & \texttt{last2}-2 & \texttt{height}-2, \\ \texttt{start1}-1 & \texttt{size1}-2 & \texttt{last1}-2 & \texttt{start2}-1 & \texttt{size2}-2 & \texttt{last2}-2 & \texttt{height}-3, \\ \texttt{start1}-4 & \texttt{size1}-1 & \texttt{last1}-4 & \texttt{start2}-1 & \texttt{size2}-1 & \texttt{last2}-1 & \texttt{height}-1 \end{array}\right\rangle, 4\right)$$

Part (A) of Figure 5.239 shows the 4 parallelepipeds of height 4, 2, 3 and 1 associated with the items of the RECTANGLES collection (parallelepipeds since each rectangle also has a height). Part (B) gives the corresponding cumulated 2-dimensional profile, where each number is the cumulated height of all the rectangles that contain the corresponding region. The cumulative_two_d constraint holds since the highest peak of the cumulated 2-dimensional profile does not exceed the upper limit 4 imposed by the last argument of the cumulative_two_d constraint.

Typical

$|\texttt{RECTANGLES}| > 1$
RECTANGLES.size1 > 0
RECTANGLES.size2 > 0
RECTANGLES.height > 0
LIMIT $<$ sum(RECTANGLES.height)

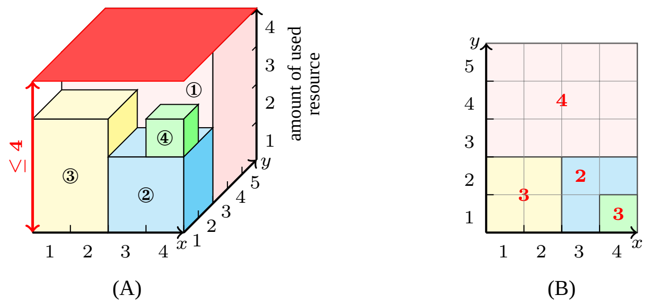

Figure 5.239: Two representations of a 2-dimensional cumulative profile of the **Example** slot (where the profile provides for each point of coordinates (c_x, c_y) the corresponding sum of the heights of the items intersecting that point): (A) a three dimensional representation and (B) a two dimensional representation from above with the height of the profile in red; as for the `cumulative` constraint the position of an item on the z axis does not matter, i.e. only its height matters.

Symmetries

- Items of `RECTANGLES` are permutable.
- Attributes of `RECTANGLES` are permutable w.r.t. permutation (`start1`, `start2`) (`size1`, `size2`) (`last1`, `last2`) (`height`) (*permutation applied to all items*).
- `RECTANGLES.height` can be decreased to any value ≥ 0.
- One and the same constant can be added to the `start1` and `last1` attributes of all items of `RECTANGLES`.
- One and the same constant can be added to the `start2` and `last2` attributes of all items of `RECTANGLES`.
- `LIMIT` can be increased.

Arg. properties

Contractible wrt. `RECTANGLES`.

Usage

The `cumulative_two_d` constraint is a necessary condition for the `diffn` constraint in 3 dimensions (i.e., the placement of parallelepipeds in such a way that they do not pairwise overlap and that each parallelepiped has his sides parallel to the sides of the placement space).

Algorithm

A first natural way to handle this constraint would be to accumulate the compulsory part [250] of the different rectangles in a quadtree [367]. To each leave of the quadtree we associate the cumulated height of the rectangles containing the corresponding region.

Systems

`geost` in **Choco**.

See also

related: `diffn` (*`cumulative_two_d` is a necessary condition for `diffn`: forget one dimension when the number of dimensions is equal to* 3).

specialisation: `bin_packing` (`square` *of size* 1 *with a* `height` *replaced by* `task` *of* `duration` 1), `cumulative` (`rectangle` *with a* `height` *replaced by* `task` *with same* `height`).

Keywords

characteristic of a constraint: derived collection.

constraint type: predefined constraint.

filtering: quadtree, compulsory part.

geometry: geometrical constraint.