MSc in Computer Science / Computational Logic

Constraint Programming

Project I – Tiling of a Rectangle

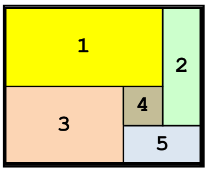
1. Specify in COMET a function **tile(x, y, w, h, r, maxW, maxH, o)** that constrains a rectangle of dimensions **maxW** and **maxH** to be tiled with a set of rectangles with widths / heights indicated in arrays **w** / **h**, respectively, and whose left bottom corners are in coordinates **x** and **y**, respectively. The rectangles may be rotated, which is indicated by the array **r**, of 0/1 FD variables. If some rectangle **i** is rotated its actual width and height are respectively h[i] and w[i], otherwise they are w[i] and height h[i]. All variables in arrays **x, y, w, h** and **r**, as well as arguments **maxW** and **maxH** are FD variables. Array **o**, is used to specify a number of options to be considered in this constraint, namely:

- **o[1] (rotations):** a Boolean indicating whether some of the *rectangles* can be rotated.
- **o[2] (redundant):** a Boolean indicating whether a redundant cumulative constraint is used to speed up the search for solutions.
- **o[3] (global):** a Boolean indicating whether the cumulative constraint is specified by means of either global (o[3] = true) or propositional (o[3] = false) constraints. Of course, if o[2] is false this option has no effect.
- **o[4] (symmetry):** a Boolean indicating whether some symmetry breaking constraints are used to speed up the search for solutions.

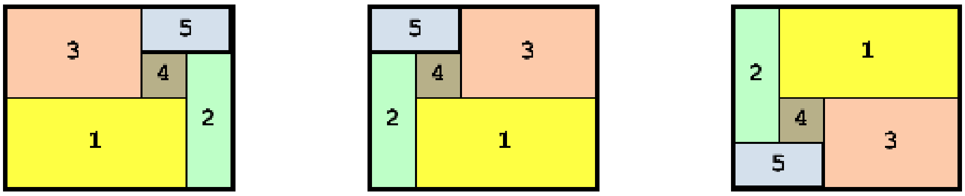
Example:

Constraint tile succeeds if called with the arguments below, corresponding to the tiling of the figure.

```
x = [1,5,1,4,4],
y = [3,2,1,2,1],
w = [4,3,3,1,2],
h = [2,1,2,1,1],
r = [F,T,F,F,F],
maxW = 5,
maxH = 4,
```



Notice that other symmetrical solutions for x and y are possible (if not prevented with o[4]).



2. Informally, two rectangles are incomparable if one does not fit inside the other. More formally, two rectangles with width/heights w_1/h_1 and w_2/h_2 are incomparable iff whenever one of the rectangles has a shorter width than it must have a longer height ($w_1 <= w_2 \Rightarrow h_1 > h_2$ && $w_2 <= w_1 \Rightarrow h_2 > h_1$), even when one of the rectangles is subject to a rotation.

   Use the above constraint to tile a 22 × 13 rectangle with 7 incomparable rectangles. Repeat the tiling for a larger 38 × 10 rectangle.

3. Write a small report, describing the results we have achieved with the different options, as well as the execution times and number of backtracks obtained.

The report and the code should be sent by email to the lecturer (**pb@di.fct.unl.pt**) no later than Sunday 15th November, at 24:00. Please use subject **Project_PR_1_by_XXXXX+YYYYY** (where XXXXX and YYYYY are the numbers of the students - max 2 per group).