Problem 132: A layout problem

Matthew Grum
Dept. Computer Science, The University of York
July 2004

This problem involes laying out a series of shapes within a larger shape so that no space is wasted and no shapes overlap. In this problem shapes are described as bitmaps, a set of unit squares (given by thier x and y co-ordinates) that representing solid area of the shape.

`given`	x_dim, y_dim, n_shapes : `nat`
`letting`	X `be` $1..x\_dim, Y$ `be` $1..y\_dim, SHAPE$ `be` $1..n\_shapes$
`given`	$Space : X \times Y, Form : SHAPE \rightarrow$ `set of` $(X \times Y)$
`find`	$Layout : Space \rightarrow SHAPE$
`such that`	$\forall_{s \in SHAPE} . \forall_{c \in Form(s)} .$
	$Layout(c[1] + \min(\text{domain}(Layout(\_, \_, s))), c[2] + \min(\text{range}(Layout(\_, \_, s)))) = s$ (1)

As layout is a function, rather than a relation, each square in the layout can only belong to one shape, ensuring that no overlapping occurs. A constraint is required to check that the squares allocated to s in the layout match the shape of s, given by $Form(s)$.

Last updated October 1, 2004
Reference:

[1] Pedro Meseguer, Carme Torras. Exploiting Symmetries Within Constraint Satisfaction Search
Artificial Intelligence: an International Journal, Volume 129, Issues 1-2 (June 2001) pages 133–163