Exercise 7

Foundations of Constraint Programming

Prof. Michael Thielscher, Sebastian Voigt

International Master Program in Computational Logic — winter term 2009/2010

Date of submission: 31.01.2010

Pallet Loading

A manufacturer wants to pack some boxes, which have the shape of cuboids, on a pallet, which is a cuboid as well. The boxes have different values and might consume more space than the pallet offers. The manufacturer now wants to find a packing arrangement that maximizes the overall value of the boxes which are packed on the pallet. As a simplification we consider a two-dimensional setting, hence no stacking on top of other objects is performed and the height of each object is smaller or equal to the height of the pallet.

The task is to write a Constraint Logic Program in Eclipse-Prolog that solves arbitrary instances of the problem which are given as a file of the following format: There is exactly one line of the form `Pallet:` $length$ `x` $width$ and one or more lines of the form `Square:` $length$ `x` $witdh$, $value$, where $length$, $width$ and $value$ are positive integers with the expected meaning. Note that squares might be rotated to fit the pallet.
Your program should provide a predicate `solve/1` which takes the filename of a problem instance as string. As a result of calling `solve("filename")`, the maximal value as well as some string representation of the corresponding optimal solution should be printed.

Examples: In the following you find two instances and solutions to them. The files can also be found on the course webpage.

1. ```
   Pallet: 5x6
   Square: 1x4, 4
   Square: 2x3, 5
   Square: 2x4, 4
   Square: 2x5, 6
   Square: 3x1, 3
   Square: 3x3, 7
   Square: 4x3, 9
   ```

   The optimal value is 24 . The followig shows a possible solution, where the colors are used to separate different rectangles and the numbers indicate their respective values:

5	5	5	3	3	3
5	5	5	9	9	9
7	7	7	9	9	9
7	7	7	9	9	9
7	7	7	9	9	9
   ```

2. `Pallet: 8x8`
   `Square: 2x4, 9`
   `Square: 2x2, 3`
   `Square: 2x3, 5`
   `Square: 2x4, 7`
   `Square: 2x4, 9`
   `Square: 2x5, 6`
   `Square: 2x7, 15`
   `Square: 3x4, 13`

   The optimal value is 61 and can be reached for example with:

15	15	3	3	9	9	7	7
15	15	3	3	9	9	7	7
15	15	9	9	9	9	7	7
15	15	9	9	9	9	7	7
15	15	9	9		13	13	13
15	15	9	9		13	13	13
15	15	5	5	5	13	13	13
		5	5	5	13	13	13