

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

Solitaire Battleship

A fleet is placed somewhere inside a 10x10 grid. A fleet consists of one battleship (four grid squares in length), two cruisers (each three grid squares long), three destroyers (each two squares long) and four submarines (one square each). The ships may be oriented horizontally or vertically, and no two ships will occupy adjacent grid squares, not even diagonally. The digits along the right side of and below the grid indicate the number of grid squares in the corresponding rows and columns that are occupied by vessels.

In each specific puzzle, one or more 'shots' have been taken to start off. These may show water (indicated by `Water`), a complete submarine (`Circle`), or the middle (`Middle`), top (`Top`), bottom (`Bottom`), left (`Left`) or right (`Right`) of a longer vessel.

The task is to write a Constraint Logic Program in Eclipse-Prolog that solves arbitrary instances of these puzzles. A description of the corresponding input file format can be found via the link on the course webpage. 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")`, some string representation of the solved grid should be printed.

Example: In the following you find an example instance (the file can also be found on the course webpage) as well as a solution to it. The Instance is given in the format mentioned above:

```
Hint: 0 5 Water
Hint: 5 3 Circle
Hint: 6 9 Circle
Row tallies:      2  4  3  3  2  4  1  1  0  0
Column tallies:  0  5  0  2  2  3  1  3  2  2
```

The single solution to this problem instance, where we encode `Water` by `.` and `Circle` (`Middle`, `Top`, `Bottom`, `Left`, `Right`) by `c` (`m`, `t`, `b`, `l`, `r`), is:

```
.....lr. 2
.t.lmr.... 4
.m.....lr. 3
.m..lr.... 3
.b.....c 2
...c.lmr.. 4
.....c 1
.c..... 1
..... 0
..... 0
0502231322
```

A number of difficult instances can be found on www.csplib.org (problem number 014).