German University in Cairo
Media Engineering and Technology
Prof. Dr. Slim Abdennadher

Concepts of Programming Languages, Spring term 2017
Project Description
"Battleship"

A fleet is placed somewhere inside a $n \times n$ grid. A fleet contains destroyers and submarines. Each destroyer is two squares in length and each submarine is one square each.

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), part of a vessel left (Left) or right (Right).

A submarine is shown by `c`. The left part of a vessel is shown by `l`. The right part is shown by `r`. A destroyer is thus represented by a left part of a vessel followed by a right part.

Your task is to implement a Prolog program to place destroyers and submarines in the grid such that all the needed properties are satisfied:

- Each row and column in a grid has to contain a specific number of vessels.
- Hints could be given to the program. A hint (e.g. Hint: 0 5 Water) means that in the cell at the given x-coordinate (0 in the example) and y-coordinate 5 (5 in the example), contains a specific vessel type (water in the example)
- Each grid has to contain a specific total number of submarines.
- Each grid has to contain a specific total number of destroyers.

Example

For a 10×10 grid with the following properties:

```
Hint: 5 0 Water
Hint: 3 5 Submarine
Hint: 9 6 Submarine

Row tallies: 0 0 0 0 0 1 1 0 0 2
Column tallies: 0 0 0 1 0 0 1 1 0 1

Total Number of Submarines: 2
Total Number of Destroyers: 1
```

The only solution is given in the below table. The extra row and column shown in the figure show the specific number in the corresponding row/column that are occupied by vessels.

Used Structure

The grid $n \times n$ should be represented using a list.

w	w	w	w	w	w	w	w	w	w	0
w	w	w	w	w	w	w	w	w	w	0
w	w	w	w	w	w	w	w	w	w	0
w	w	w	w	w	w	w	w	w	w	0
w	w	w	w	w	w	w	w	w	w	0
w	w	w	c	w	w	w	w	w	w	1
w	w	w	w	w	w	w	w	w	c	1
w	w	w	w	w	w	w	w	w	w	0
w	w	w	w	w	w	w	w	w	w	0
w	w	w	w	w	w	l	r	w	w	2
0	0	0	1	0	0	1	1	0	1	

Tabelle 1:

Predicates to be added

The rest of the description will guide you through the predicates that you need to implement to be able to solve the problem.

Hints

Hints are given as facts of the following form: at(5,0,w). at(3,5,c). at(9,6,c).

mycheck/2

The predicate `mycheck(L)` should succeed only if `L` represents a grid/part of a grid such that if `L` starts with a left part of a vessel (`l`), it is followed by the right part (`r`).

list_to_llists/2

The predicate `list_to_llists(L,W,LLists)` should succeed only if `LLists` is a list of lists containing the elements of `L`. Each list in `LLists` has a length equal to `W`.

getZeroth/2

The predicate `getZeroth(L,El)` holds if `E1` is the first element of the list `L`.

rest/2

The predicate `rest(L,R)` holds if `R` contains all the elements of `L` except the first one.

subList/4

The predicate `subList(I1,I2,L,Sub)` holds if `Sub` is a sublist of the list `L`. `Sub` contains the elements occuring in `L` from index: `I1` till `I2`.

collect_hints/1

The predicate `collect_hints(H)` holds if `H` is a list of the provided hints.

ensure_hints/4

The predicate `ensure_hints(L,Hints,W,H)` holds if in the grid represented by `L` whose width is `H` and height is `H`, all the hints given in the list `Hints` are satisfied.

random_assignment/1

The predicate `random_assignment(L)` holds if the grid `L` is filled with any of the possible values (`w, c, l or r`).

check_rows/3

The predicate `check_rows(L,W,H,Totals)` holds if `L` represents a grid/partial grid with a width `W` and height `H` such that the count of vessels in each row corresponds to the counts given in `Totals`.

check_columns/3

The predicate `check_columns(L,W,H,Totals)` holds if `L` represents a grid/partial grid with a width `W` and height `H` such that the count of vessels in each column corresponds to the counts given in `Totals`.

check_destroyer/3

The predicate `check_columns(L,W,H,TotalDestroyer)` holds if `L` represents a grid/partial grid with a width `W` and height `H` such that the count of destryoers in `L` corresponds to `TotalDestroyer`. If `L` represents the full grid, then the count should be exactly equal to `TotalDestroyer`.

check_submarines/3

The predicate `check_submarines(L,W,H,TotalSub)` holds if `L` represents a grid/partial grid with a width `W` and height `H` such that the count of submarines in `L` corresponds to `TotalSub`. Similarly, if `L` represents the full grid, then the count should be exactly equal to `TotalSub`.

battleship/4

The predicate `battleship(L,W,H,TotalSub,TotalDes,TotalRows,TotalColumns)` holds if `L` represents a battleship grid with a width `W` and height `H` such that the count of submarines in `L` corresponds to `TotalSub` and the count of destroyers is equal to `TotalDes`. The sum of each row and each column of the grid is represented in `TotalRows,TotalColumns` correspondingly.

Submission Guidelines

- The deadline of the project will be on **Saturday the 7th of April**.
- You should submit the following form: `https://goo.gl/forms/okeNQcTCAob5O8g53` with your team's information maximum by **Saturday 25/3/2016**. Each team should be of **2-3** members.
- Each team should submit a single *.pl* file, via the MET website submission link, containing the team's full project implementation. The submitted file **must** abide by the following rules:
  a) The file should be named according to your team's *Team Name* (from the to-be-posted teams list).
  b) You should include a clear documentation per implemented predicate. You should write each predicate's documentation above the predicate's implementation.
- It is the team's full responsibility to successfully submit a valid .pl file before the project's deadline.