

Till Mossakowski

Formal Methods for Software Development

Exercise sheet 4: first “big” exercise

Suppose that we want to implement a simple graphics package. For example, a graphics screen could be represented as

```
type Coord = (Int,Int)
type Screen = [Coord]
```

(but you might choose your own representation).

Specify (in P-logic) and implement (in Haskell) the following functions for drawing a pixel, a line, and a circle, and for uniting and intersecting two screens:

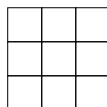
```
pixel :: Coord -> Screen
line  :: Coord -> Coord -> Screen
circle :: Coord -> Int -> Screen
union  :: Screen -> Screen -> Screen
intersection :: Screen -> Screen -> Screen
neighbourhood :: Coord -> Screen
```

A line is given by its start and end point, a circle by its centre and its radius. The function `union` should unite two screens, such that you can write e.g.

```
test = line (4,4) (14,14) 'union'
      line (4,14) (14,4) 'union'
      circle (9,9) 8
```

Similarly, `intersection` should intersect two screens.

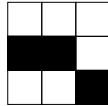
The function `neighbourhood` should return a square of 9 pixels around the given coordinate:



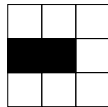
The specification should express that

- all the pixels on the drawn line are within distance 1 from the ideal line,

- all the pixels on the drawn circle are within distance 1 from the ideal circle,
- for any pixel on a line, the intersection of its neighbourhood with the line has cardinality 3:



except for the endpoints, where it has cardinality 2:



- for any pixel on a circle, the intersection of its neighbourhood with the circle has cardinality 3.

Use Programatica and its QuickCheck certificates to validate the assertions of your specification.

It will also be convenient to have a function that outputs a screen on your computer:

```
draw :: Screen -> IO()
```

This function need not be specified.

Line and circle drawing algorithms can be found e.g. under

<http://www.cs.unc.edu/~mcmillan/comp136/Lecture6/Lines.html>.

<http://www.cfxweb.net/modules.php?name=News&file=article&sid=998>

Please return the solutions to Till Mossakowski (till@tzi.de) before

5th January 2005