Advanced Programming
Assignment 0

Nikolaj Dybdahl Rathcke (rfq695)
Victor Petrén Bach Hansen (grn762)

September 6, 2015

Implemented Functions

The functions implemented in the assignment is briefly explained though we omitted some trivial functions and type declarations.

`connect`: Connect takes the first point from the first curve and uses this as the new first point for the new connected curve. The remaining points are then appended on each other in the correct order, namely the order [curve1 tail, curve2 head, curve2 tail], creating a the new connected curve.

`rotate`: For the function rotate, the rotation matrix for the cartesian plane were used. The rotation matrix R is

$$R = \begin{bmatrix} cos\theta & -sin\theta \\ sin\theta & cos\theta \end{bmatrix}$$

For a point (x, y) and an angle θ, the new coordinates (x', y') become

$$x' = xcos\theta - ysin\theta$$
$$y' = xsin\theta + ycos\theta$$

Since `sin` and `cos` takes the angle v in radians, it is converted from degrees to radians as: $\frac{v \cdot \pi}{180}$.

`translate`: When translating the curve, the distance between the given point and the first point of the curve was first calculated (the difference in x and y). When these values were found, they were added to all the point in the curve.

`reflect`: When reflecting a point over a custom horizontal/vertical axis that has the function $y = d$ or $x = d$, the new points can be calcualted as $(2 \cdot d - x, y)$ when the line is vertical and $(x, 2 \cdot d - y)$ when it is horizontal.

`bbox`: The function finds the lower left corner and the upper right corner of the bounding box for a curve c. This is achieved by finding minimum and maximum values of x and y in the curve as $(min(x), min(y))$ and $(max(x), max(y))$ corresponds to the corners. In the code, we used the high order function `foldl` on the list and the functions `min` and `max` to find values for x and y.

`width`: The width can easily be computed as it corresponds to the difference between $min(x)$ and $max(x)$ found with the `bbox` function. It is simply $max(x) - min(x)$.

`height`: Likewise, the height corresponds to the difference between $min(y)$ and $max(y)$ which is also found with `bbox` function.

`toSVG`: As the the header of the svg file is fixed, whereas the number of lines would have to be generated from the number of points in the curve, this part of the svg-file is filled in with the height and width of the file (rounded up using ceiling). The line elements of the curve was then recursively generated from the list of points in the curve, appending these on one another using the seperate function `svgLines`.

`toFile`: The function `writeFile` is used to create a new file with a given name, containing the generated svg content (using the function `toSVG`) from the given curve.

Code Assessment

The code works entirely as intended. We used the test framework given in the assignment as an indication for quality. It passed all tests without any error and no comments. Using `hlint` on `Curves.hs` produced no suggestions either.
Generally, the approach when writing the code, was to make it as clear as possible, instead of writing convoluted one-liners that solved the problem.