0 Accumulator-Style Program Design

Portfolio Problems

For each of the following problems work out the solution in four different ways:

- using the design recipe
- modifying the previous solution by using an acumulator
- implementing the solution using the Scheme loop *foldl*
- implementing the solution using the Scheme loop *foldr*

Problems:

1. Problem 31.3.3 in HtDP
2. Problem 31.3.4 in HtDP
3. Compute the distance traveled along the given nonempty list of *Posn*s
4. Determine the minimum distance between two consecutive *Posn*s in a list of *Posn*s that has at least two *Posn*s
5. Concatenate all *String*s in the given list of *String*s

Pair Programming Assignment

0.1 Problem

Your database records the names of students and for each student a list of courses student is taking. It is sufficient to represent each student just by the name (a *String*) and to represent each course by the course number (again just a *String*).

Design the program that produces a roster for a given course from the given list of students. The roster is just a list of student names.

A. Write down the necessary data definitions - with examples of data

B. Design the solution in four different ways:

- using the design recipe
- modifying the previous solution by using an acumulator
- implementing the solution using the Scheme loop *foldl*
- implementing the solution using the Scheme loop *foldr*

0.2 Problem

A list of *Posn*s represents a polygon that we can draw on a canvas by drawing a line between every two consecutive *Posn*s and finally adding a line from the last point to the first one. We assume that the polygon consists of at least three *Posn*s.

A. Design the data to represent a polygon and make examples of data.

B. Design the function that will draw ths polygon on the canvas. Use the world teachpack. Consult the *H*elp Desk if necessary.

Now work on a design of two polygons that will morph from the original image to the final one as the time goes on. For examples, a hexagon may morph into a triangle by *p*ulling in three of its points until they lie on the line between the other two adjacent points.

C. Design two polygons, *start* and *finish*, with the same number of *Posn*s.

D. Design the function *morph-poly* that for the given *start* and *finish* polygon and the given *factor* that is a value between 0 and 1 produces a new polygon where each point lies between the corresponding two points of the original polygons.

   That means that for the *factor* with the value `0.5` the resulting polygon will be exactly half-way between the *start* and the *finish*.

E. Design a *MorphWorld* that will consist of the *start* and *finish* polygons, and a timer value *t* that will on each tick draw the morphed polygon with the *factor* value *t* / `1000` and increase the value of *t* by one.

F. **Extra Credit** Design a function that will generate the values for the factor from the timer value in such way that the polygon will keep morphing from the *start* image to the *finish* image and back again to the *start* image in a continous loop.