6.005 Elements of Software Construction  
Problem Set 2b: The Symbolic Paradigm  

Solutions

Abstract Datatypes

Note: There are many possible solutions to each of these questions.

1. producers:
   // Takes an integer and returns a constant IntPoly, eg. "3"
   constant: int -> IntPoly

   // Takes a variable as a char and returns a variable IntPoly, eg. "x"
   variable: char -> IntPoly

   // Adds two IntPolys, eg. add("3x + 2", "x^2") -> "x^2 + 3x + 2"
   add: IntPoly, IntPoly -> IntPoly

   // Subtracts two IntPolys
   subtract: IntPoly, IntPoly -> IntPoly

   // Multiplies two IntPolys, eg. multiply("3", "x") -> "3x"
   multiply: IntPoly, IntPoly -> IntPoly

2. producers:
   // Returns the empty string
   empty: void -> String

   // Append a character to a String, eg. "ab" + 'c' -> "abc"
   append: String, char -> String
// Concatenate two strings, eg. concatenate("ab", "cd") -> "abcd"
concatenate: String, String -> String

observers:
// Returns the character at a given index
charAt: String, int -> char

3.

BillCoin = Bill(denomination : int) + Coin(value : float)

Cash

producers:
// Creates an empty collection of Cash
none: void -> Cash

// Operations that add to your collection, eg. earn(none, 3.00) = "$3.00"
earn: Cash, BillCoin -> Cash
earn: Cash, Cash -> Cash
earn: Cash, float -> Cash

// Operations to spend your collection, eg. spend(3.00, 1.00) = "$2.00"
spend: Cash, BillCoin -> Cash
spend: Cash, Cash -> Cash
spend: Cash, float -> Cash

observers:
// Check how much money you have, eg.
// value(earn(Coin(.25), earn(Bill(1), none))) = 1.25
value: Cash -> float

4.

Point = Point(latitude : float, longitude : float)
Route

producers:
    // Create a route segment between two Points
    route: Point, Point -> Route

    // Insert a point into the route at a given index
    insert: Route, int, Point -> Route

    // Remove a point from the route
    remove: Route, Point -> Route
    remove: Route, int -> Route

observers:
    // Get the nth point in the route
    get: int -> Point

    // Get the distance from point-to-point of the entire route
    distance: Route -> float

5.
Grid = EmptyGrid + Cons(row : Row, rest : Grid)
Row = EmptyRow + Cons(tile : Tile, rest : Row)
Tile = Blank + X + O

Invariant:
- No null Grid, Row, or Tile
- Size of Grid is 3
- All three Rows are size 3
- # Xs equal to or one more than # Os (X plays first)
- # Xs + # Os + # Blanks == 9 (redundant; implicit from the first 3)

Implementation

6.
The implementation part is available under the Published directory of the SVN repository.