PTP, or PointToPoint minimalistic programming language

Project Proposal For COMS W4115 (Programing Languages and Translators)
Columbia University, Summer 2011

Author: Venera Varbanova (vv2200@columbia.edu)
June 8, 2011

Introduction

The purpose of this language is to express and manipulate navigation of point-sized items (aka points) in a 2-dimensional space, given a sequence of steps. Each sequence of steps is called a ‘trail’. For simplicity, the steps are always of size 1, and their direction is one of the 4 geo directions - east, west, north, or south. Each sequence of directions is applied to a point which is regarded as the starting location, and it leads to a destination location which is another point. The trails can be merged, and can be replicated, in order to produce bigger trails. The idea behind the language is to keep it simpler but more stable, and to give priority to quality over quantity, as this is an individual project and the summer semester is relatively short, making it even harder to deal with the learning curve associated with the tools that we are required to use and to still deliver what was proposed. Another idea behind the language is that it will be relatively easy to test, and the behavior is simple and pretty self-explanatory, thus reducing the possibility of undefined behavior. (If you get a point on the 2d plane and start moving it around, it will never have undefined behavior, the point will invariably be somewhere on that same 2d plane, we just have to be keep track where).

Language Features

The language will support 2 complex data types (trail and point). Arrays of complex data types will be supported, too. The only simple data type that will be used is integer, and it will be part of the complex data types. The operators + and * will be used to manipulate the trails object, + and - will be supported for integers. The programming language will also support control constructs in the form of if statements, and foreach loops. Functions will be called by value, and will always have a predefined return type.

Sample Program

Please note that the exact syntax may evolve, this is just preliminary pseudo-code trying to demonstrate the idea behind the PTP language.

```
enum dir {east,west,south,east}
object point(int x, int y)
object trail (listof dir)

start-function move (point p, dir d) returns point
        point ret(p)

        if (d == east)
          ret.x = ret.x + 1

        if (d == west)
           ret.x = ret.x - 1

        if (d == north)
           ret.y =  ret.y + 1

        if (d == south)
           ret.y = ret.y - 1

        return ret
end-function

start-function go(trail t, point p) returns point
        point ret(p)
        foreach dir d in t
             ret = move(ret, d)
        return ret
end-function

start-program

trail a [east, west, east, west, east]
trail b [east, east, east, east, east]
point origin [0,0]

point destinationA = go(a, origin)
print(destinationA)

point destinationB = go(b, origin)
print(destinationB)

trail c = a+b
```

```
point destinationAB = go(c, origin)
print(destimationAB)

trail a2 = a*2;
point destinationAx2 = go(a2, origin)
print(destinationAx2)

# optional, will be implemented if the rest is not complex enough
int d = distance(p1, p2)
print(d)

end-program
```