Proyecto 1

Eddy Ramírez
Lenguajes de programación
Instituto Tecnológico de Costa Rica

II Semestre 2014

1. Programación genética

Esta es la especificación del segundo proyecto de Lenguajes de programación.

2. Especificación

Su tarea consiste en realizar en *Scheme* utilizando programación genética un programa que genere programas (funciones) que cumplan con ciertas restricciones.

2.1. Programación genética

La programación genética es una variedad de los algoritmos evolutivos (dentro de los cuales están los algoritmos genéticos) la cual busca hacer programas que para ciertas entradas produzcan salidas de acuerdo con una serie de restricciones.

En particular, la diferencia que existe con los algoritmos genéticos tradicionales es que los individuos en la programación genética son en realidad programas o funciones, por lo que su número de genes puede ser variable.

Por otra parte, la forma de evaluar cada individuo es en cuánto se aproximan ciertas entradas a ciertas salidas que son seleccionadas bajo algún criterio humano.

2.2. Funciones en Scheme

Dado que en Scheme, las funciones son escritas en preorden, pueden ser vistas como individuos, donde cada árbol tiene como raíz la función y como

hijos las subfunciones o parámetros de la función raíz. Por otra parte, siendo apoyados por la función ***eval***, se pueden evaluar árboles como funciones para saber su resultado para ciertas entradas.

El cruce debe de ser definido por cada grupo de trabajo.

2.3. Objetivo del programa

El programa generado debe de poder generar una función en (x, y, z) de la forma: $f(x, y) = z$.

Esta función tiene aplicaciones reales en la generación de mapas para zonas de poco acceso, se sobrevuela dejando una serie de sensores, que durante algún tiempo puedan emitir su ubicación en coordenadas (x, y) y su altura relativa con respecto al nivel del mar (z). Finalmente, para generar el mapa, basta con encontrar la función que pase por todos los puntos o lo que es lo mismo que minimice la diferencia entre los puntos reales y los puntos por los que pasa la función.

2.4. Funciones permitidas

Las funciones siempre serán binarias y serán: $\{+, -, *, /, expt, bin{-}and, bin{-}or, bin - xor\}$ donde las funciones bin-and, bin-or y bin-xor son el resultado de aplicar la función definida sobre la representación binaria de cada número que recibe como parámetro.

3. Entrada del programa

El programa recibe como entrada una serie de puntos dispuestos en triadas de números enteros, una triada por cada línea.

Un ejemplo de esa entrada sería:

```
2 3 10
6 9 30
12 -3 18
```

4. Puntos extra: GUI

Para motivar al estudiante a investigar sobre el manejo de interfaz gráfica, se dará un 10 % extra a aquel grupo de trabajo que muestre en una imagen 3D el gráfico generado por la función que se ha encontrado, dentro

de un cuadrante definido por (en el peor caso) los x, y máximos y mínimos del archivo.

5. Consideraciones generales

En esta sección aparecen las consideraciones generales a tomar en cuenta para el desarrollo de este proyectos.

5.1. Aspectos técnicos

1. Toda la programación debe de realizarse en Scheme y debe de ejecutarse correctamente en GNU/Linux.

2. Se debe de entregar únicamente un archivo .zip con los fuentes (y sólo los fuentes). En caso de utilizar bibliotecas gráficas externas, indicar una manera de instalarlas o adjuntar las bibliotecas también

3. El nombre de la función principal debe de ser: *genetica* y recibe como parámetro único la dirección del archivo que contiene los puntos.

   Ejemplo: (genetica "entrada.txt")

5.2. Aspectos administrativos

- Cualquier falta de los aspectos técnicos implicará una nota de cero
- Cualquier sospecha de fraude implicará una nota de cero
- La fecha de entrega es el 4 de octubre a los correos: proyectos.eddy@gmail.com y proyectos.eddy.cartago@gmail.com
- El proyecto puede ser realizado en parejas