NOTAS ACERCA DEL EJERCICIO 5.

SERGIO GALÁN MARTIN - RAFAEL SÁNCHEZ SÁNCHEZ - G2202

En esencia hemos tratado de seguir en la medida de lo posible las interfaces sugeridas. Sin embargo, si hemos introducido algunos cambios en ellas que nos parecían más eficientes, como, por ejemplo:

-Añadir un parámetro al constructor del DominioAritmetico (tolerancia), en el que especificaremos con que tolerancia aceptaremos los valores obtenidos comparados con la respuesta correcta.

-No hemos hecho ningún control de las excepciones ArgsDistintosFuncionesException e IllegalArgumentException, ya que si se produce alguno de esos dos fallos son errores irrecuperables, por lo que simplemente hacemos throw de ellas y saldrá el mensaje de error por la consola. Sin embargo, con CruceNuloException, al ser un estado del que nos podemos recuperar, la hacemos catch y reintentamos el cruce. El resto de las excepciones (IOException, FileNotFoundException...) tampoco las hemos controlado ya que también son irrecuperables.

-Hemos añadido a INodo las funciones setPadre y detachPadre, que nos permiten, principalmente al hacer el cruce, llevar el control de quien es el nodo padre del nodo que tenemos que cruzar, para poder eliminar dicho nodo de la lista de descendientes. También hemos añadido algunos getters y setters adicionales.

-Hemos añadido a IIndividuo la función getNodo, que nos permitirá buscar un nodo con cierta etiqueta en la expresión del Individuo. Por supuesto, también lo hemos tenido que añadir en INodo.

-Hemos añadido en IDominio la función getVp, que nos permitirá acceder al mapa de valores de prueba del Dominio.

-Hemos añadido un método privado en la implementación del Algoritmo que es un método auxiliar para la creación de la población inicial por recursión.

-Para permitir una gran flexibilidad en el Algoritmo, en el constructor le pasamos una gran cantidad de parámetros que permitirán adaptarlo a diferentes situaciones.

En nuestro test del Algoritmo hemos fijado los siguientes parámetros, con lo que obtenemos un individuo con máximo fitness en una media de 300 generaciones:

Profundidad máxima: 3 Individuos torneo: 6

Población inicial: 100 individuos Objetivo: 1

Máximas generaciones: 7000

El parámetro Objetivo es el porcentaje en decimal sobre el fitness máximo que, si un individuo lo alcanza, lo consideraremos exitoso. En este caso no tiene mucho sentido, ya que una vez que un individuo llega a fitness 4, el siguiente es 21. Con otro algoritmo u otro fitness quizá tendría más sentido buscar una precisión más sesgada, como un acierto de 0.9 sobre 1.