

Gramáticas evolutivas

Cargamos la gramática en memoria desde un archivo

Como parámetros de entrada dejamos que el usuario elija el tamaño del cromosoma y la cantidad de wraps a considerar.

Generamos la población inicial de forma aleatoria con cromosomas con valores entre 0 y 255 y después de generar cada individuo calculamos su fitness.

Lo primero a mirar es que el individuo sea válido, es decir, que terminemos de generar la lista de operaciones antes de terminar todos los wraps permitidos, al tiempo que calculamos si un individuo es válido también calculamos la lista con las operaciones. En esta lista solo vamos a tener terminales (AO, A1, D0, D1, D2, D3) y operación IF.

Clase Gramatica

```
public class BNFGramatica {

    //Mapa con la gramática
    private Map<String, List<List<String>>>> gramatica;
    public BNFGramatica(String s) {

        gramatica = new HashMap<String, List<List<String>>>>();

        //Separamos por líneas la gramatica
        String r[] = s.split("\n");

        for(int i = 0; i < r.length; i++) {
            //Separamos las reglas de sus variables
            String regla[] = r[i].split("::=");

            //Separamos cada una de las variables
            String reglas_sig[] = regla[1].split("\\|");

            //Creamos la lista para las listas de reglas
            List<List<String>> list1 = new ArrayList<List<String>>();
            for(int j = 0; j < reglas_sig.length; j++) {
                //Hacemos un split de las reglas por los espacios
                String reglas_juntas[] = reglas_sig[j].split(" ");

                . . . . .
            }
        }
    }
}
```

Clase Individuo:

```
public class Individuo implements Comparable<Individuo> {

    private double fitness;

    private int wrap;
    private int[] individuo;
    . . .
    private BNFGramatica gramatica;
    private List <Reglas> r;

    . . .
}
```