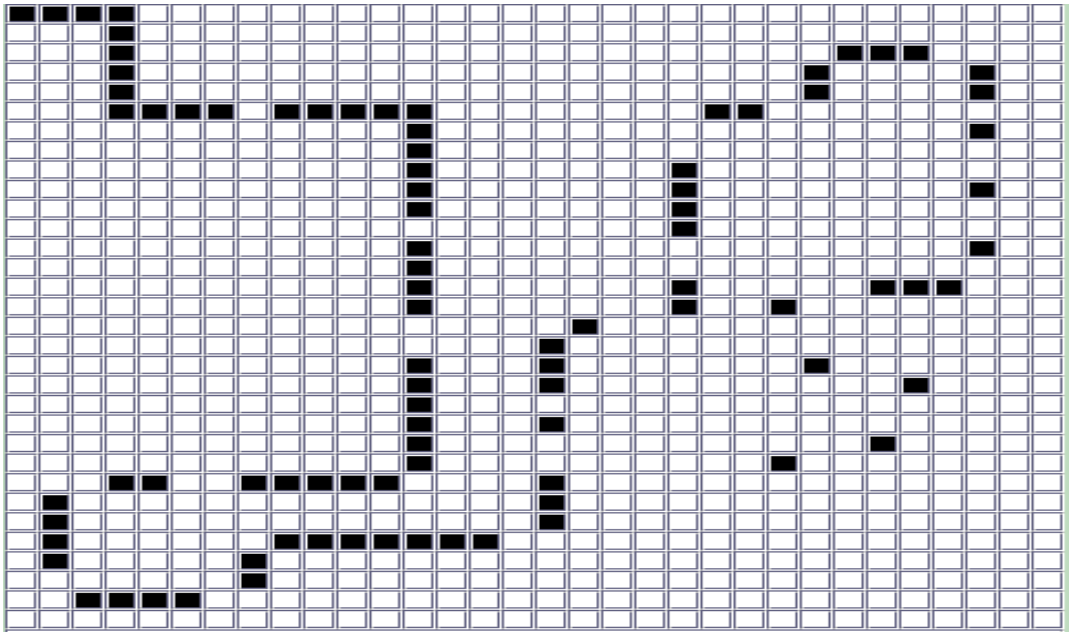
Programación evolutiva
Facultad de Informática
Curso 2018/2019

Práctica 3: Programación genética

La práctica consiste en resolver mediante programación genética el problema de la hormiga que se mueve por un tablero buscando comida. La hormiga debe ser capaz de encontrar toda la comida situada a lo largo de un rastro irregular. Dicha hormiga se mueve en un tablero (toroidal) de 32 x 32 celdas. La hormiga comienza en la esquina superior izquierda del tablero, identificada por las coordenadas (0,0), y mirando en dirección este. El modelo de rastro propuesto, con el que se han realizado diversos estudios, se conoce como "rastro de Santa Fe" , y tiene una forma irregular compuesta de 89 "bocados" de comida. El rastro no es recto y continuo, sino que presenta huecos de una o dos posiciones, que también pueden darse en los ángulos.



Nuestro problema requiere disponer operaciones que permitan a la hormiga moverse hacia delante, girar a uno u otro lado y detectar comida a lo largo de rastro irregular. El conjunto de operandos (terminales) para este problema es:

Terminales = {**AVANZA, DERECHA, IZQUIERDA**}

donde **AVANZA** mueve la hormiga hacia delante en la dirección a la que mira en ese momento, **DERECHA** gira la hormiga 90° a la derecha, e **IZQUIERDA** la gira 90° a la izquierda.
Las funciones disponibles son:

Funciones = { **SIComida(a,b), PROGN2 (a,b), PROGN3 (a,b,c)** }

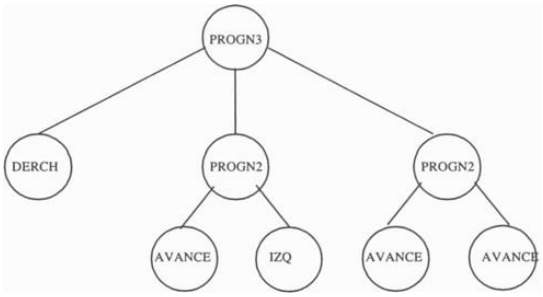
La información que queremos procesar es la que la hormiga obtiene del mundo exterior a través de un sencillo sensor que detecta si hay comida en frente. Por ello incluiremos en el conjunto de operadores uno de selección condicional dependiente de la información del sensor.

En el conjunto de operadores incluiremos también conectivas que fuerzan la ejecución de sus argumentos en un determinado orden. Tomando el nombre de una conectiva de Lisp, PROGN, que causa este efecto, incluimos en el conjunto de operadores las conectivas PROGN2 y PROGN3, que toman dos y tres argumentos respectivamente. Por lo tanto disponemos de las siguientes funciones:

- **`SIComida (a,b)`** *:* el operador **SIComida** toma dos argumentos y ejecuta **a** si se detecta comida delante y **b** en otro caso.
- **`PROGN2(a,b)`** : evalúa **a**, luego **b**, y devuelve el valor de **b**.
- **`PROGN3(a,b,c)`** : evalúa **a**, **b** luego **c**, devolviendo el valor de **c**

Un ejemplo de programa:

```
(PROGN3 (DERECHA)
        (PROGN2  (AVANZA) (IZQUIERDA))
        (PROGN2 (AVANZA) (AVANZA))
```



PASOS:

- Se genera un población inicial de programas aleatorios (árboles) usando el conjunto de funciones y terminales posibles. Estos árboles deben ser sintácticamente correctos. Limitamos la profundidad. Técnica Ramped and half.
- La función de fitness será la cantidad de alimento comido por la hormiga dentro de un espacio de tiempo razonable al ejecutar el programa a evaluar. Se considera que cada operación de movimiento o giro consume una unidad de tiempo. Limitaremos el tiempo a 400 pasos. El tablero se va actualizando a medida que desaparece la comida.
- Operador de Cruce: Intercambiar dos subárboles (elegidos aleatoriamente) entre los dos árboles padres.
- Operador de Mutación: de terminal, de función, de inicialización.
- Alguna forma de controlar el bloating para mejorar la adaptación y evitar programas muy largos
- Se mostrarán las gráficas de evolución y la visualización de la ejecución de la solución: representar gráficamente la evolución de un juego en el tablero al aplicar la estrategia que codifica la solución.

ENTREGA

- **Plazo de entrega: 14 de mayo**. Debes entregar por el campus un archivo comprimido con el código java de la aplicación (**proyecto en Eclipse** o **NetBeans)** que incluya una breve memoria que contenga el estudio de las gráficas y resultados obtenidos. Aquí se valorarán las conclusiones y observaciones que se consideren interesantes respecto al resultado obtenido. Nombre del proyecto-archivo: **G01P2** (por ejemplo, para el grupo 01)

- La **corrección** será en la sesión de Laboratorio del **17 de mayo** y deberán estar presentes los dos miembros del grupo, a los que se evaluará por igual mediante una serie de preguntas. Es importante conocer bien la práctica y los aspectos teóricos en los que se basa, pues es lo que determina la calificación. El orden de corrección de cada grupo será el mismo que el orden de entrega por el campus.