

## **Proyecto final de Simulación y Programación Declarativa**

### **Amalia Ibarra Rodríguez - C412**

#### 1. Orden del problema

El ambiente en el cual intervienen los agentes es discreto y tiene la forma de un rectángulo de  $N \times M$ . El ambiente es de información completa, por tanto todos los agentes conocen toda la información sobre el agente. El ambiente puede variar aleatoriamente cada  $t$  unidades de tiempo. El valor de  $t$  es conocido. Las acciones que realizan los agentes ocurren por turnos. En un turno, los agentes realizan sus acciones, una sola por cada agente, y modifican el medio sin que este varíe a no ser que cambie por una acción de los agentes. En el siguiente, el ambiente puede variar. Si es el momento de cambio del ambiente, ocurre primero el cambio natural del ambiente y luego la variación aleatoria. En una unidad de tiempo ocurren el turno del agente y el turno de cambio del ambiente. Los elementos que pueden existir en el ambiente son obstáculos, suciedad, niños, el corral y los agentes que son llamados Robots de Casa. A continuación se precisan las características de los elementos del ambiente:

**Obstáculos:** estos ocupan una única casilla en el ambiente. Ellos pueden ser movidos, empujándolos, por los niños, una única casilla. El Robot de Casa sin embargo no puede moverlo.

No pueden ser movidos ninguna de las casillas ocupadas por cualquier otro elemento del ambiente.

**Suciedad:** la suciedad es por cada casilla del ambiente. Solo puede aparecer en casillas que previamente estuvieron vacías. Esta, o aparece en el estado inicial o es creada por los niños.

**Corral:** el corral ocupa casillas adyacentes en número igual al del total de niños presentes en el ambiente. El corral no puede moverse. En una casilla del corral solo puede coexistir un niño. En una casilla del corral, que esté vacía, puede entrar un robot. En una misma casilla del corral pueden coexistir un niño y un robot solo si el robot lo carga, o si acaba de dejar al niño.

**Niño:** los niños ocupan solo una casilla. Ellos en el turno del ambiente se mueven, si es posible (si la casilla no está ocupada: no tiene suciedad, no está el corral, no hay un Robot de Casa, y aleatoriamente (puede que no ocurra movimiento), a una de las casillas adyacentes. Si esa casilla está ocupada por un obstáculo este es empujado por el niño, si en la dirección hay más de un obstáculo, entonces se desplazan todos. Si el obstáculo está en una posición donde no puede ser empujado y el niño lo intenta, entonces el obstáculo no se mueve y el niño ocupa la misma posición.

Los niños son los responsables de que aparezca suciedad. Si en una cuadrícula de 3 por 3 hay un solo niño, entonces, luego de que él se mueva aleatoriamente, una de las casillas de la cuadrícula anterior que esté vacía puede haber sido ensuciada. Si hay dos niños se pueden ensuciar hasta 3. Si hay tres niños o más pueden resultar sucias hasta 6.

Los niños cuando están en una casilla del corral, ni se mueven ni ensucian. Si un niño es capturado por un Robot de Casa tampoco se mueve ni ensucia.

Robot de Casa: El Robot de Casa se encarga de limpiar y de controlar a los niños. El Robot se mueve a una de las casillas adyacentes, las que decida. Solo se mueve una casilla sino carga un niño. Si carga un niño puede moverse hasta dos casillas consecutivas.

También puede realizar las acciones de limpiar y cargar niños. Si se mueve a una casilla con suciedad, en el próximo turno puede decidir limpiar o moverse. Si se mueve a una casilla donde está un niño, inmediatamente lo carga. En ese momento, coexisten en la casilla Robot y niño.

Si se mueve a una casilla del corral que está vacía, y carga un niño, puede decidir si lo deja esta casilla o se sigue moviendo. El Robot puede dejar al niño que carga en cualquier casilla. En ese momento cesa el movimiento del Robot en el turno, y coexisten hasta el próximo turno, en la misma casilla, Robot y niño.

## 2. Objetivos

El objetivo del Robot de Casa es mantener la casa limpia. Se considera la casa limpia si el 60 % de las casillas vacías no están sucias.

### **Ideas generales**

Se inicializa el ambiente con una cantidad específica de niños, robots, suciedad, obstáculos y un corral con tamaño de la cantidad de niños. A partir de aquí la simulación avanza por turnos, primero se mueve el agente, luego los niños y si ya ha transcurrido el tiempo  $t$  ocurre la variación aleatoria del ambiente generando suciedad en las cuadrículas de 3x3 donde antes hubo un niño.

Cada vez que se completa un ciclo se aumenta el tiempo y se calcula el porcentaje de suciedad, si este es mayor que 40 se detiene la simulación y significa que el robot no pudo cumplir su objetivo. También se puede definir un tiempo límite para la simulación.

## **Modelos de Agentes**

Se implementó el modelo de agente pro-activo cuyo objetivo es que los niños no generen suciedad, para esto se prioriza llevarlos hasta el corral o cargarlos, cuando todos los niños estén controlados entonces se pone a limpiar.

## **Ideas seguidas para la implementación**

El ambiente se representa con una lista del datatype Element que contiene data constructors simulando los elementos. Cada elemento tiene su coordenada en el ambiente y los robots y los niños cuentan además con un Boolean que indica si están cargando a un niño o siendo cargados por un robot respectivamente.

Para los movimientos del agente se simula una máquina de estado auxiliándonos del pattern matching sobre los elementos que hay en la celda del mismo. Si está cargando un niño y todavía quedan niños libres avanza hacia el corral más cercano, si ya está parado en el corral suelta al niño y si no quedan niños libres entonces comienza a limpiar con el niño cargado, no lo lleva hasta el corral. En cualquier otro caso el robot avanza hacia el niño más cercano y sólo cuando estén todos controlados (cargados o en un corral) es que comienza a limpiar.

Una vez el agente decide hacia qué elemento se quiere aproximar el próximo paso a dar se encuentra realizando un bfs para encontrar el elemento y recorriendo el camino hacia atrás auxiliándonos de la matriz de distancia que devuelve el bfs.

Para ejecutar el proyecto correr:

```
cabal run Main.hs
```

Es necesario tener los módulos Data.Matrix y Data.Random que se instalan con los comandos:

```
cabal install matrix y cabal install random
```

## **Resultados**

Se realizaron pruebas en distintos escenarios variando la cantidad de niños, robots, suciedad inicial, obstáculos, el tamaño del tablero y el robot logra mantener la limpieza de la casa durante el tiempo de la simulación en la gran mayoría de los casos.