

# Proyecto conjunto de Programación Declarativa y Simulación. Andy Sánchez Sierra C-411

## 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 adyacente, 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.

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.

## 2- Principales Ideas

El ambiente en el cual intervienen los agentes es discreto, determinista y de información completa, por tanto, todos los agentes conocen toda la información sobre el Ambiente. Estas características, permiten definir un conjunto de reglas a seguir por cada agente para alcanzar el objetivo. Se presentarán 2 enfoques encaminados a cumplir dicho objetivo.

- 1- Solo se enfocará en limpiar la basura, sin tener en cuenta los niños ni nada más en el ambiente.
- 2- En primer lugar, se enfocará en llevar niños al corral, una vez no queden niños fuera del corral, se enfocará en limpiar la basura sin tener en cuenta nada más en el ambiente.

Los agentes, en cada turno, deciden que acción realizar, dicha acción depende de que tipo de enfoque este realizando.

Acciones que realizan los agentes del primer enfoque:

- 1- Si se encuentra encima de una basura, la limpia.
- 2- Si quedan basura en el ambiente, se mueve hacia la más cercana.
- 3- Si no quedan basura, se queda en el lugar.

Acciones que realizan los agentes del segundo enfoque:

- 1- Si tiene un niño cargado y se encuentra en una celda del corral, deja el niño.
- 2- Si tiene un niño cargado y está fuera del corral, se mueve hacia la casilla del corral más cercana.
- 3- Si no tiene un niño cargado y quedan niños en el ambiente, se mueve hacia el más cercano.
- 4- Si no tiene un niño cargado, no quedan mas niños en el ambiente y está parado encima de una basura, la limpia.
- 5- Si no tiene un niño cargado, no quedan más niños en el ambiente y quedan basura en el ambiente, se mueve hacia la más cercana.
- 6- Si no tiene un niño cargado, no quedan más niños en el ambiente y no quedan basura, se queda en el lugar.

### 3- Modelo de Agentes considerado

#### **Agente reactivo**

La principal característica de estos agentes es la capacidad para responder de un modo oportuno a los cambios que ocurren en el medio. Si se tiene en cuenta que el ambiente definido cambia constantemente de forma natural y/o aleatoria en cada turno, el hecho de que el robot sea capaz de responder ante ello, demuestra la propiedad reactiva del mismo.

### 4- Simulación

#### **Simulacion.hs**

Es en donde se simula el comportamiento del ambiente.

En primer lugar, usando el paquete Environment.hs, inicia el ambiente usando el método `initEnvironment` pasándole `n` y `m`, el tamaño que se desea crear el ambiente.

Una vez creado el ambiente, Se llama al método `simulate` pasándole el ambiente ya creado y el numero de simulaciones que se desean realizar, este de manera recursiva realiza una `k` cantidad de simulaciones, realizando las siguientes acciones:

- 1- Cada `t` simulaciones, llama al método `changeEnvironment` del paquete Environment.hs, el cual se encarga de modificar el ambiente generando basuras nuevas de manera aleatoria.
- 2- Llama al método `simulateChild` del paquete Childs.hs, el cual se encarga de simular las acciones que realizan cada niño del ambiente.
- 3- Llama al método `simulateRobots` del paquete Bots.hs, el cual se encarga de simular las acciones que realizan cada agente del ambiente.

## **Types.hs**

En este paquete de encuentran todos los tipos que se usan para facilitar el desarrollo de la simulación.

```
data Direction = North | South | East | West
data Accion = Move | Drop | Clean | Pass
data Cell = Obstacle | Child | Dirt | Playpen | Bot | Empty
```

`Direction` se usa para definir una dirección, esta es usada para realizar los movimientos de los niños y los agentes en el ambiente.

`Accion` se usa para definir las acciones que realizan los agentes en el ambiente.

`Cell` se usa para definir cada uno de las casillas del ambiente.

## **Environment.hs**

En este paquete se encuentra toda la funcionalidad del ambiente en si, como es crear el ambiente, modificarlo, insertar una nueva casilla, eliminarla, entre otras.

## **Childs.hs**

En este paquete se encuentra toda la funcionalidad que realizan los niños, como es moverse, ensuciar el ambiente, mover obstáculos, entre otras

Además, cuenta con dos métodos que son los encargados de realizar la simulación de los niños.

`simulateChild`: realiza en movimiento aleatorio de un niño y decide si ensuciar la parte del ambiente donde se encuentra.

`simulateChilds`: de forma recursiva recorre todos los niños del ambiente y ejecuta en cada uno el método `simulateChild`.

## **Bots.hs**

En este paquete se encuentra toda la funcionalidad que realizan los agentes, como es moverse limpiar, soltar los niños, entre otras.

Además, cuenta con dos métodos que son los encargados de realizar la simulación de los agentes.

`simulateRobot`: este busca entre todas las acciones que puede realizar el agente, cual es la mejor a realizar y la realiza.

`simulateRobots`: de forma recursiva recorre todos los agentes del ambiente y ejecuta en cada uno el método `simulateRobot`.

## **BFS.hs**

En este paquete se encuentran los métodos que se encargan de encontrar cual es la mejor dirección a donde se debe mover un agente en un momento dado, Para encontrar dicha dirección, se realiza un bfs y se devuelve la dirección donde se encuentra la primera casilla de uno de los caminos mínimos desde el agente hasta el objeto que desea llegar.

## **Utils.hs**

En este paquete se encuentran métodos auxiliares que se utilizan en los demás paquetes, entre los que se encuentran imprimir el ambiente, eliminar elementos de una lista, devolver el elemento de una posición en una lista, devolver la longitud del ambiente, devolver las casillas adyacentes a una casilla, entre otros

## **5- Consideraciones**

Una vez ejecutadas varias simulaciones, se llega a la conclusión de que el segundo enfoque es más efectivo, dado que el primer enfoque decide nada más limpiar la basura, como no se preocupa por los niños, estos siguen ensuciando el ambiente. El segundo enfoque, como primero decide llevar los niños al corral, cuando se ejecutan varias simulaciones, una vez todos los niños en el corral, estos no ensucian mas y entonces limpian mas basura que el primer enfoque.