

Universidad de La Habana  
Facultad de Matemática y Computación

Proyecto conjunto de  
Programación Declarativa

y  
Simulación

Miguel Alejandro Asin Barthelemy C411  
**Tema:** Agentes

## Marco General

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 ambiente. 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 casilla 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.

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.

## Principales ideas

El problema se desarrolla en un ambiente discreto, dinámico, determinista y de información completa. Estas características, permiten definir un conjunto de reglas a seguir por cada agente para alcanzar el objetivo de mantener el 60 % del medio limpio. Se presentarán varios enfoques encaminados a cumplir dicho objetivo y se seleccionarán aquellos que se considera, son una mejor solución.

Se conoce que la tarea principal del Robot de la Casa es mantener suficientemente limpio al medio, para ello existe un enfoque directo: limpiar las celdas sucias accesibles por el agente y uno indirecto: llevar todos los niños hacia el corral y limpiar la mayor cantidad de celdas en el proceso. El inconveniente de la primera opción es que en el proceso de desplazamiento del robot hacia una celda sucia son altas las probabilidades de que los niños que se encuentren fuera del corral generen más suciedad, por lo que la cantidad de celdas sucias no harían más que aumentar. En el caso de la segunda opción, en el momento en el que el niño es cargado por el robot, la cantidad de basura máxima generada en cada turno disminuye respecto al valor antes de cargarlo. Teniendo en cuenta estos aspectos, se opta entonces por la opción de buscar a los niños y limpiar en el proceso.

Las acciones del agente se pueden dividir en dos grupos: cuando carga un niño y cuando no lo hace. En cada uno de estos grupos, se persiguen pequeños objetivos en concreto a fin de lograr el objetivo mayor de mantener limpio el ambiente.

En caso de que el robot esté cargando un niño, puede optar por los siguientes objetivos:

- 1.1 El robot se encuentra en una celda del corral y es el mejor lugar en el cual podría dejar al niño. Cada celda del corral tiene asignado un valor o puntuación que puede cambiar en

cada turno, la cual le brinda a los agentes la posibilidad de diferenciar un buen lugar para dejar el niño de uno malo.

**Condiciones:** Se encuentre en el corral y sea una de las celdas del corral accesible con mayor puntuación. Para la puntuación de una celda del corral, se adiciona un punto por cada lado bloqueado y uno por cada esquina (dos lados adyacentes bloqueados).

1.2 Seguir un camino hacia la celda del corral libre con mayor puntuación.

**Condiciones:** Que no se cumpla la condición de 1.1 y que exista al menos una celda del corral accesible.

1.3 Limpiar celdas sucias accesibles.

**Condiciones:** Que no se cumplan las condiciones de 1.1 – 1.2 y no exista un camino hacia el corral sin cargar niño.

1.4 Dejar al niño en la celda en la que se encuentra.

**Condiciones:** Que no se cumplan las condiciones de 1.1 – 1.3.

En caso de que el robot **no** esté cargando niños, puede optar por los siguientes objetivos:

2.1 Recoger el niño de la celda en la que se encuentra

**Condiciones:** Que haya un niño en dicha celda y que exista un camino hacia una celda del corral vacía.

2.2 Buscar y recoger al niño que se encuentra en el camino hacia una celda libre del corral.

**Condiciones:** Que no se cumpla la condición 2.1 y, que exista un camino hacia un niño y otro desde el niño hacia el corral que no pase por otro niño.

2.3 Limpiar celdas sucias.

**Condiciones:** Que no se cumplan las condiciones de 2.1 – 2.2 y que existan celdas sucias accesibles por el robot.

2.4 No hacer nada

**Condiciones:** Que no se cumplan las condiciones de 2.1 – 2.3.

Las conductas anteriores se realizarán por el robot en el orden en el que aparecen. Cada una de ellas presentan condiciones que es necesario se cumplan para poder ser efectuadas por el agente. Si una de ellas no cumple con sus condiciones, el agente pasa a evaluar la siguiente conducta. Siempre será posible elegir una por parte del agente, de ello se encarga las condiciones para realizar la condición 1.4 en el primer caso o 2.4 en el segundo.

Como puede apreciar de las conductas anteriores, aunque el agente priorice el traslado de los niños hacia el corral, siempre limpiará la mayor cantidad de celdas sucias posibles y, en caso de que no pueda trasladar niños, convertirá entonces a la limpieza de la suciedad en la conducta con mayor prioridad.

## Modelos de Agentes considerados

Todos los agentes son sistemas capaces de realizar acciones autónomas con el fin de cumplir un objetivo dado. Dichas acciones autónomas las podemos definir como un conjunto de reglas que ha de seguir el robot y, si se tiene en cuenta la propiedad dinámica del medio en el cual se encuentra y la capacidad del mismo para responder a los cambios del medio, se obtendría entonces un agente reactivo y proactivo.

## 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.

En cada turno el robot de la casa debe ser capaz de determinar cual es el objetivo inmediato a partir de la percepción que obtiene del medio. Anteriormente se describieron las condiciones que son necesarias para que el agente determine dicho objetivo. Si dichas condiciones se cumplen, el agente establece su objetivo y la cadena de acciones encaminadas a cumplirlo (de ahí que el robot de la casa sea pro-activo). Por ejemplo, suponga que el robot se encuentra en una celda cargando un niño. Primeramente se selecciona una ruta hacia una celda del corral libre (asuma que existe la ruta) y se mueve por un camino óptimo hasta llegar a ella; en el próximo turno el ambiente podrá cambiar ya sea de forma natural o aleatoria, en cualquier caso, el robot reevaluará nuevamente las conductas para determinar cual es el mejor objetivo del nuevo ambiente y la cadena de acciones para alcanzarlo.

Sin embargo este comportamiento tiene sus desventajas. Una vez el robot establece su objetivo no lo modifica, por lo que en un medio en el cual intervengan varios agentes sería ineficiente que varios robots se encaminen a cumplir exactamente el mismo objetivo. Para evitarlo se decidió implementar entonces otro modelo de agente.

## Agente Inteligente

Hasta el momento se han dado las características de un agente reactivo y proactivo, sin embargo, nunca llega a mostrar un comportamiento sociable con otros agentes del medio. Para convertir al agente reactivo-proactivo anterior en un agente inteligente es necesario incluir en el mismo la capacidad de comunicación con otros robots de la casa.

El objetivo de la comunicación entre los agentes es lograr una mayor eficiencia en el proceso de limpieza del medio, por ello debe ser capaz de modificar la conducta de varios agentes para distribuir las pequeñas tareas a realizar en un instante de tiempo dado. Para ello se establece entonces la condición de que todos los agentes deben tener un objetivo inmediato distinto (que las celdas a las cuales se dirijan no coincidan) y, para lograrlo, cada agente debe comunicarse con el resto de los agentes para determinar que celdas se encuentran en el objetivo de otro agente y de esta forma optar por otra distinta.

Cumpliendo con las características anteriores, es posible considerar entonces a todos los robots de la casa como agentes inteligentes.

## Algoritmo de la Simulación

El código de la simulación se desarrolló en el lenguaje de programación funcional Haskell. No se requirieron de librerías ni paquetes adicionales salvo las proporcionadas por el lenguaje. Este se compone por seis paquetes *.hs*:

*Enviroments.hs*: Es en donde se simula el comportamiento del ambiente. Primeramente se crea el ambiente inicial con *init\_enviroment*. Para ello se generan las celdas que formarán el corral con *generate\_corral*, los robots que se encargarán de mantener la casa limpia, los niños con *generate\_children*, los obstáculos con *generate\_obstacles* y la suciedad con *generate\_dirt*. En el transcurso de cada unidad de tiempo ocurren los cambios naturales

del ambiente, los cuales están dados por las acciones de los niños. Cada niño tiene una probabilidad de 0,5 de moverse, sin embargo solo lo hará si en la dirección en la que se moverá, está la celda vacía o existe un obstáculo que puede ser desplazado en caso de ser empujado. En ese mismo turno se genera de forma aleatoria la variable que indica si el niño generará suciedad o no, de hacerlo, se toma una celda al azar del tablero y si está limpia entonces se genera la nueva suciedad en la celda. Tras cumplirse el tiempo  $t$  preestablecido para el cambio del ambiente, se genera un nuevo estado del medio (*randomize\_enviroment*), con una nueva distribución de los elementos del mismo a excepción de los agentes, los cuales mantendrán la última posición que registraron.

*ReactiveRobot.hs*: En este se describe el comportamiento reactivo de los agentes. Está compuesto por tres funciones, dos de las cuales son las principales: *run\_robot* el cual recibe el robot al cual le toca actuar en el turno y retorna un nuevo ambiente resultado de la acción realizada por el robot y; *action* la cual determina, según el estado del robot y del ambiente, la próxima conducta del robot en correspondencia con la acción a realizar por el mismo para cumplirla.

*SmartRobot.hs*: Es quien describe el funcionamiento de los agentes inteligentes. La diferencia fundamental entre *run\_irobot* de este fichero y el *run\_robot* visto anteriormente, es la función *erase\_objectives* de *run\_irobot* la cual elimina como objetivo del robot actual, los objetivos definidos por los agentes analizados anteriormente.

*Structures.hs*: Contiene las estructuras de datos usadas en la simulación. Como se puede apreciar, los robots de la casa son definidos con el tipo de dato *Agent*, los cuales pueden ser *Robot* (agentes reactivos) o *IRobot* agentes inteligentes, la principal diferencia entre ellos es que los agentes inteligentes almacenan el objetivo que persiguen según el estado del medio, el cual pueden compartir con otros agentes para alcanzar una mayor eficiencia cumpliendo la tarea de limpieza.

*Prints.hs*: Es el encargado de brindar una representación visual aceptable al usuario que represente los estados por los que pasa el ambiente según las acciones que se van realizando.

*Utils.hs*: Un conjunto de funciones útiles. Juegan un rol secundario en la simulación.

Para entrar en el entorno se debe ejecutar el fichero *Enviroments.hs*:

```
$ ghci Enviroments.hs
```

El inicio de la simulación está en la función *main*, la cual recibe cuatro enteros ( $n$   $m$   $t$  *robot\_cant*) y un booleano (*smart\_robots*) los cuales indican la cantidad de filas del tablero, la cantidad de columnas, el tiempo definido para el cambio aleatorio del medio, la cantidad de robots de casa y si son agentes inteligentes, respectivamente. Ejemplo:

```
Enviroments> main 5 5 60 2 False
```

Una vez se inicia, cada turno se divide en la(s) acción(es) del(los) agente(s) (que son invocados desde *run\_simulation*) y el cambio natural del medio, en caso de que haya transcurrido  $t$  turnos desde el último cambio aleatorio o del inicio del medio, se genera entonces un cambio aleatorio del mismo.

Para la representación visual de los datos, se usará una matriz de  $n \times m$  en la que los elementos del medio tendrán los siguientes símbolos:

**C:** Corral

**N:** Niño

**O:** Obstáculos

**S:** Suciedad

**R:** Robot de la Casa

**CN:** Niño en corral

**CR:** Robot en Corral

**CNR:** Robot y niño en corral

**NR:** Robot y niño en la misma celda

**SNR:** Robot cargando niño en celda sucia

En la generación de los valores aleatorios se parte de una semilla inicial igual a 0, si desea modificar este valor es necesario hacerlo manualmente en el algoritmo en la función *main* (línea 282).

## Consideraciones

A continuación se muestran los resultados obtenidos tras ejecutar 5 veces el algoritmo para dos agentes reactivos y dos agentes inteligentes.

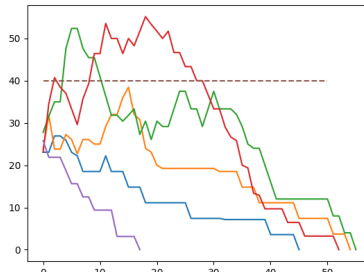


Figura 1: Por ciento de suciedad por turnos a lo largo de la simulacion (Agentes reactivos).

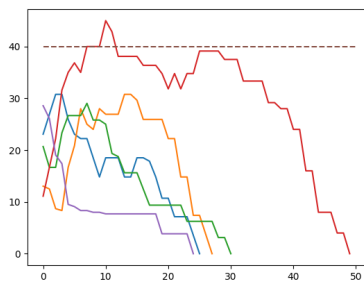


Figura 2: Por ciento de suciedad por turnos a lo largo de la simulacion (Agentes inteligentes).

Como se puede apreciar existe gran diferencia entre ambos, una mejor cooperación entre los agentes fue suficiente para marcar la diferencia en el objetivo final del robot de la casa de mantener el 60 % del ambiente limpio.