

# Home Robot

---

Karla Olivera Hernández

C411

## 1. 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 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 la 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.

#### **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.

## **2. Principales ideas seguidas para la solución del problema**

En cada turno de la simulación primero se va iterando por cada robot dentro del ambiente y se manda a ejecutar una acción. Luego si es el momento del cambio aleatorio, se itera de igual manera por cada niño y se ejecuta su acción generada.

Un niño primero verifica qué posiciones tiene disponibles a las que se pueda mover, si no existe ninguna, permanece en su lugar ese turno. En otro caso, escoge de forma aleatoria una de las posiciones disponibles, se mueve y genera suciedad en alguna de las casillas que quedaban a su alrededor en la posición anterior. Si la posición a la que se va a mover, tiene un obstáculo, primero verifica que lo puede mover, sino también se queda en la misma posición.

Para el caso de los robots primero se verifica que en la posición en que se encuentre haya suciedad y no este cargando un niño, si ese es el caso, entonces la limpia y termina el turno. En otro caso, si tiene un niño cargado busca el corral más cercano y se mueve dos posiciones en esa dirección, o una si ya es suficiente para llegar. Si no puede llegar a un corral, entonces deja al niño y pasa a buscar la suciedad más cercana y se mueve una posición en esa dirección.

## **3. Modelos de agentes considerados**

El problema solo cuenta con un solo tipo de agente: los robots y se proponen dos tipos de modelos, ambos reactivos, pero cambiando el orden de prioridad de cada categoría.

#### **Modelo 1:**

**if** *hay suciedad en la casilla actual y no está cargando un niño*

**then** *limpiar suciedad*

**if** *no está cargando un niño y hay un niño accesible*

**then** *moverse a la casilla más cercana a un niño*

**if** *no está cargando un niño y no hay un niño accesible y hay suciedad accesible*

**then** *moverse a la casilla más cercana a una suciedad*

**if** *está cargando un niño y no está en el corral y hay un corral accesible*

**then** moverse a la casilla más cercana a un corral

**if** está cargando un niño y no está en el corral y no hay un corral accesible y hay suciedad accesible

**then** dejar al niño y moverse a la casilla más cercana a una suciedad

**else** moverse de manera random

## Modelo 2:

**if** hay suciedad en la casilla actual y no está cargando un niño

**then** limpiar suciedad

**if** hay suciedad accesible y está cargando un niño

**then** dejar el niño y moverse a la casilla más cercana a una suciedad

**if** hay suciedad accesible y no está cargando un niño

**then** moverse a la casilla más cercana a una suciedad

**if** no está cargando un niño y hay un niño accesible

**then** moverse a la casilla más cercana a un niño

**if** está cargando un niño y no está en el corral y hay un corral accesible

**then** moverse a la casilla más cercana a un corral

**else** moverse de manera random

## 4. Ideas seguidas para la implementación

El repositorio está estructurado en 5 módulos diferentes:

- Input: Este módulo está destinado a que el usuario introduzca los datos del tablero inicial, para que a partir de este se ejecute la simulación.
- Main: Se encuentra la función inicial del programa, `main` la cual se encarga de llamar en un primer momento a la función `initEnvironment` para inicializar el entorno, luego inicia la simulación y, por último, una vez que esta termina, comprueba si fue exitosa o no. De esta ultima operación se encarga la función `houseIsClean` que comprueba que la cantidad de casillas vacías sea mayor o igual que el 60 % del total de las casillas.

La función que se encarga de llevar el ciclo de la simulación es `simulateTurn` la cual primero llama a que se realicen los cambios naturales (los realizados por los robots) y luego los aleatorios (los realizados por los niños).

- Types: Se encuentran definidas las dos principales estructuras del programa:
  - `Square` que representa cada casilla del entorno, con sus atributos `row` y `column`.
  - `Environment` que se encarga de guardar el estado del ambiente en todo momento: `nRows`, `nColumns`, `currentTime`, `finalTime`, `randomTime`, `stdGen` (se va actualizando cada vez que se genera un número random nuevo, para que cada vez que se calcule uno no se obtenga el mismo). Además tiene cuatro listas para guardar los elementos que se encuentran dentro del ambiente: `obstacles`, `dirt`, `corral`, `children` y `robots`.

Están implementadas además algunas funciones como: `initEnviroment` mencionada en el módulo anterior que se usa para generar el ambiente; entre otras destinadas a actualizar los atributos del `Enviroment`.

- Utils: En este módulo tenemos algunas funciones de ayuda, como las encargadas de imprimir el ambiente en consola y otras de verificar que se cumplan algunas condiciones que son de ayuda en otros módulos.
- Child: La función principal es `childrenTurn` que se encarga de recorrer la lista de children del enviroment y luego ejecutar `childAction` por cada uno de ellos. Esta última función primero calcula los posibles lugares a los que un niño puede moverse, que son aquellas casillas que no tienen suciedad, otro niño, un robot o un corral. En caso de no existir posibles movimientos, el niño permanece ese turno sin moverse; sino se escoge de manera aleatoria una de esas posibles posiciones. Si la posición escogida es una casilla vacía, entonces se mueve y genera suciedad en una de las 9 casillas que se encontraban a su alrededor en la posición anterior. Si la posición escogida es un obstáculo, entonces verifica que pueda moverlo, en caso de que no pueda entonces permanece en el mismo lugar.
- Robot: Al igual que en el módulo anterior, en este tenemos una función `robotsTurn` que recorre la lista de robots del ambiente y por cada uno ejecuta `robotAction`. Esta función primeramente pregunta si en la posición en que está parado el robot hay también una suciedad. En caso de ser cierto, la limpia y termina su turno; sino pregunta si hay un niño y si no hay un corral, lo cual significa que el robot tiene cargado al niño y se va a dirigir a la casilla más cercana a un corral. Si ninguna de las opciones anteriores se cumple, entonces el robot se dirige al niño más cercano. Si el robot trata de buscar un corral y no encuentra ninguno, entonces deja al niño en la posición en que está y se mueve buscando la suciedad más cercana; lo mismo sucede si trata de buscar a un niño y no lo encuentra.

Para hacer los cálculos de cual es la casilla mas cercana que cumpla un objetivo, se implemento un BFS que va a retornar las dos primeras casillas, en caso de existir, que estén en el camino más corto. Se devuelven dos, ya que se utilizan para que si el robot va a cambiar de posición y trae un niño cargado pueda moverse dos casillas.

## 5. Consideraciones obtenidas a partir de la ejecución de las simulaciones del problema

El modelo del agente robot que mejor funcionó en la mayoría de los casos fue el primero, ya que prioriza poner a los niños en el corral para que de esta forma no generen más suciedad.