

Proyecto de Simulación y Programación Declarativa AGENTES-HASKELL



UNIVERSIDAD DE LA HABANA
FACULTAD DE MATEMÁTICA Y
COMPUTACIÓN

Sheyla Cruz Castro
Grupo C-412

Índice general

Problema	2
Solución	4
2.1. Lógica del ambiente:.....	4
2.2. Ideas de la implementación y Consideraciones:	5
2.3. Aspectos a tener en cuenta:	6
2.4. Repositorio en Github:.....	7

1. Problema

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.

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. Solución

2.1 Lógica del ambiente: Se tiene una habitación de n filas y m columnas, las cuales son introducidas por el usuario, y un valor t_0 , el cual indica que a cada t_0 intervalos de tiempo, la habitación varía aleatoriamente. Los objetos que coexisten en el ambiente son los robots de casa, los niños, la suciedad, el corral y los obstáculos, los mismos van a tener una cantidad aleatoria una vez que comience el programa. Para que su visualización sea un poco más dinámica se procede a pintar la habitación con símbolos coloreados, una vez que se inicia sale la descripción de estos.

Se asume que la cantidad de niños que habitan en la habitación es igual a la cantidad de columnas de la misma puesto que se fija el corral en la última fila del tablero. Los niños en cada turno pueden o no realizar una acción seguido de las acciones de los agentes. Se asume que es totalmente distinto el cambio aleatorio del ambiente con el desplazamiento del niño incluyendo o no suciedad.

El cambio aleatorio del ambiente viene dado a generar un nuevo tablero prefijando la cantidad de objetos actual. También se debe tener en cuenta que los agentes realizan alguna actividad si y solo si el ambiente se encuentra sucio. El ambiente se encuentra sucio si existe más de un 40 % de casillas sucias.

Se tienen cuatro tipos de robots: aquellos que no tienen niño cargado y se encuentra fuera del corral(R) o dentro del mismo(RC), y los que tienen niño cargado y están fuera del corral(RW) o dentro(RWC). Se separan de esta forma porque en el subsistema de control se establecen las siguientes prioridades:

$$RC < RWC < RW < R$$

Se tiene de esta manera ya que se establecen tareas fijas para cada uno de ellos, y en el proyecto se considera que tiene más importancia sacar a un robot del corral que ya cumplió su tarea para darle espacio a otro que de ser posible, le está ocupando la casilla, al igual que los robots que tienen niño y están dirigiéndose al corral, porque mientras más rápido esos robots estén libres más tareas pueden realizar. Por último se analizan los robots solos fuera del corral ya que estos mediante una búsqueda en profundidad se les asigna una función y se dirigen hacia el gradiente más cercano.

2.2 Ideas de la implementación y Consideraciones: Se implementa agentes de razonamiento práctico, para ello se separan a los robots en grupo de preferencia, ya que cada robot va a tener una tarea específica. Particularmente se centra en la Arquitectura BDI (Believes-Desires-Intentions). Se tienen dos cuestiones por robot: qué objetivo se quiere lograr y cómo lo va a lograr. Está claro que las intenciones están relacionadas con las creencias futuras, ya que mientras el robot no cambie de estado siempre va a buscar lograr el mismo objetivo, o sea, el robot que tiene un niño cargado desea dejar al niño en el corral, de tal manera que ese niño no moleste ni ensucie más, mientras no esté en el corral (RW) su objetivo es llegar a él, una vez que esté dentro automáticamente cambia su estado para RWC. Lo mismo ocurre con RWC, hasta que el robot no llegue a una casilla del corral habitable no abandona el corral y por lo tanto no suelta al niño para cambiar al estado RC. El robot que se encuentre en el estado RC se desplaza para cualquier casilla externa al corral para cambiar a R, donde finalmente su intención es desplazarse a limpiar o cargar a un niño, depende de la casilla más cercana que encuentre será su labor.

Se escogió esta arquitectura ya que las intenciones dirigen al planeamiento, limitan las futuras deliberaciones, son persistentes e influyen en las creencias sobre las cuales está basado el razonamiento práctico en el futuro. En este caso, esta arquitectura no está propensa a fallas porque se garantiza que siempre el robot puede cumplir con su intención ya que por ejemplo siempre podrá dejar a un niño en el corral ya que existe camino y existen tantas casillas en el corral como niños.

ASPECTOS IMPORTANTES:

- *Las creencias actuales:* información completa del ambiente.
- *Revisión de creencias (brf):*
 - Entrada perceptual: estado el robot.
 - Conjunto de creencias: la posición actual dada por el estado del robot.
 - Nuevo conjunto de creencias: la nueva posición generada para acercarse a su objetivo.
- *Función de generación de opciones (options):*
 - R: buscar suciedad o niño más cercano, sino existe caminar aleatoriamente.
 - RW: entrar al corral.
 - RWC: depositar al niño en una casilla del corral posible.
 - RC: salir del corral.

-
- *Una función de filtrado (filter):*
 - R: dirigirse a nueva tarea < desplazar aleatoriamente.
 - RW: entrar al corral < acercarse al corral.
 - RWC: depositar niño < dirigirse a una posición correcta para depositarlo.
 - RC: salir del corral < acercarse a los bordes.
 -
 - *Función de selección (execute):* Se seleccionan sobre la base de filter, de tal manera que cada estado cambia a:
 - $R \rightarrow RW$
 - $RW \rightarrow RWC$
 - $RWC \rightarrow RC$
 - $RC \rightarrow R$

Para establecer la coordinación y planificación en la interacción de los agentes, se implementa la arquitectura de tres capas horizontales de producción activa para agentes de TOURING MACHINES.

En cada turno cada agente sigue teniendo previsto su objetivo, pero eso no significa que siempre realizará la acción más óptima sin contar con los otros agentes, y así se evitan conflictos de intenciones, además que se controlan las nuevas posiciones tomadas. Ejemplo, en la capa de modelación se prevén estos conflictos ya que los agentes negocian sus posiciones para de esta forma todos tener futuras acciones lo más óptimas posibles.

En resumen, cada agente tiene un conjunto de opciones a ejecutar, en cada estado se organizan estas opciones y se otorgan a cada robot una sin crear conflictos con los robots de su mismo estado. Estas nuevas posiciones se conocen por los robots que aún no han ejecutado sus opciones y trabaja como una máscara, no puede existir nueva posición que ya haya sido tomada, por lo tanto, en cada estado se resuelven las opciones considerando las acciones ya realizadas en el turno actual.

2.3 Aspectos a tener en cuenta:

- ✓ Se asume que las entradas al programa sean correctas, o sea, que se introduzcan números naturales.
- ✓ Se fija un contador “c” para que el programa itere c veces, si se desea realizar una mayor visualización del comportamiento del ambiente se debe aumentar su número. El mismo se encuentra como primer parámetro en el método “agentStart” (*Main.hs*). Actualmente su valor es 10.

2.4 Repositorio en Github: <https://github.com/sheycc/Proyecto-Agentes-Haskell>