

Universidad de La Habana
Facultad de Matemática y Computación
Proyecto conjunto de
Programación Declarativa
y
Simulación

Tema:

Agentes

Autor:

Yandy Sánchez Orosa – C411

https://github.com/tingui96/Proyecto_Sim_Decl

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 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.

Principales Ideas

La tarea principal del Robot de la Casa es mantener suficientemente limpia la casa, al menos el 60% de las casillas, pero tener niños fuera de los corrales hace que constantemente se este generando basura debido a que la probabilidad a generar basura en el proyecto es de $\frac{2}{3}$ (bastante elevada, aunque esa probabilidad se puede cambiar manualmente) por lo que muchas veces es mejor para mi que el robot guarde los niños primero y por el camino ir recogiendo basura porque de no ser así es mas probable que los niños ensucien mas casillas en lo que el robot va recogiendo basura lo que puede crear un ciclo infinito (los niños ensuciando – el robot limpiando) y de la manera antes mencionada reduce las probabilidades que se genere más basura en la casa, debido a que existe un niño menos solo sin ser atendido por un robot. Aunque las 2 simulaciones están implementadas y se pueden utilizar cada una solo con un cambio en los parámetros

de la entrada, la mejor, en este caso es guardar primero a los niños para reducir esas probabilidades y luego recoger toda la basura.

Por lo tanto uno de los casos de paradas del algoritmo es cuando todos los niños estén dentro del corral y cuando al menos el 60% de la casa este limpia. En el otro caso los robots le dan prioridad a recoger basura si la basura sobrepasa el 60% de las casillas totales de la casa, de otra forma van a recoger a los niños y su caso de parada es el mismo pero hay un máximo de 100000 turnos que simula el algoritmo para finalizar el ciclo y no pase lo antes planteado que aunque tenga poca probabilidad puede pasar.

Modelos de Agentes considerados

Niños: Se representa por una N en el tablero. Siempre van a existir la misma cantidad de corrales que niños y estos se mantienen en el lugar. Los niños se mueven de manera aleatoria por el tablero teniendo una probabilidad de $\frac{1}{2}$ de generar basura a su alrededor, esto se logra generando un random con modulo 4 que serian los 4 movimientos:

- norte
- este
- sur
- oeste

Siguiendo las manecillas del reloj y comenzando desde 0 hasta 3.

Cuando un niño es cargado por un robot o cuando esta en un corral, este esta bloqueado, no puede generar basura ni moverse.

Agente reactivo con cierto grado de pro-actividad

Robot: Se representa por una R en el tablero. Un robot tiene que tomar varias decisiones antes de moverse o hacer algo en su turno.

1-Lo primero que hace es da igual hacia donde se dirija ,si esta en una casilla sucia, la limpia.

Sabiendo esto podemos pasar al otro punto.

2-Buscar si existe algún niño libre en el tablero, luego haciendo un BFS busca el camino mas corto para llegar hasta él. Todos los turnos realiza la misma operación debido a que si el niño esta en movimiento tiene que trazar otro camino.

3-Un robot va a buscar donde existe basura si y solo si no hay niños libres.

Esto se hace en ese orden para aprovechar la búsqueda de los niños y la ida hacia los corrales para ir limpiando un poco el ambiente.

Agente reactivo con cierto grado de pro-actividad

Robot:

- 1-Lo primero que hace es ir igual hacia donde se dirija ,si se encuentra un niño en el camino lo carga.
- 2- Revisa si las casillas limpias en la casa es más del 60%, en caso de no serlo realiza un BFS para buscar el camino mas corto hacia alguna basura y se dirige hacia la misma a limpiarla.
- 3-Un robot va a buscar directamente a un niño si y solo si el 60% de la casa esta limpia.
- 4-Si tiene un niño cargado y la casa se considera limpia realiza un BFS para buscar el corral mas cercano y dirigirse hacia él.
- 5-Si no tiene niño cargado y la casa se considera limpia realiza BFS para buscar el niño mas cercano.

No agentes pero ocupan posiciones en la casa

Obstáculos: Se representa por una O en el tablero, estos solo pueden ser movidos por los niños

Basura: Se representa por una B en el tablero. Solo desaparece cuando un robot esta en la misma casilla.

Corral: Se representa por una C en el tablero

Cada cierto tiempo t todas las cosas de la casa pueden cambiar de lugar, por lo que si en algún momento por generar aleatoriamente las posiciones, quedó algún corral bloqueado o algún robot bloqueado ya sea por obstáculos o cualquier otra cosa que pueda pasar, solo falta que pasen t turnos para continuar con la simulación cuando cambie el ambiente y llegar al caso de parada antes mencionado.

Algoritmo de la Simulación

El código de la simulación se desarrolló en el lenguaje de programación funcional Haskell.

Enviroment.hs Es en donde se simula el proyecto

Tools.hs Un conjunto de funciones auxiliares.

Robot.hs Donde se lleva a cabo todos los movimientos y las decisiones del robot para cada turno.

House.hs Donde se crea el modelo para simular la casa que no es mas que una matriz de fila por columnas y tiene todas las funciones para ver los adyacentes, generar tablero vacío, buscar celdas que contengan algún objeto en especifico, etc.

Constant.hs Contiene las constantes a usar en el proyecto para mayor facilidad.

Niño.hs Contiene todas las funciones relacionadas con los niños, como generar basura, si esta bloqueado, los movimientos, etc.

El proyecto se corre usando

-> \$ ghci Enviroment.hs

Enviroments > main turn row column obstacles kids trash robots ia
donde:

turn es cantidad de turnos para que cambie el ambiente

row y **column** cantidad de filas y columnas de la casa a generar

obstacles cantidad de obstáculos a generar en la casa

kids cantidad de niños a generar en la casa (misma cantidad de corrales)

trash cantidad de basura a generar en la casa, igual se puede pasar 0 y los niños tienen alta probabilidad de generarla

robot cantidad de robots a generar en la casa

ia 1 para priorizar los niños o 2 para priorizar la basura

Consideraciones obtenidas a partir de las simulaciones

Simularemos y tomaremos en cuenta 3 casos, cuando la cantidad de niños es mayor que la de robots, cuando son iguales, y cuando la cantidad de robots es mayor a la de niños y probaremos como se comportan ambos modelos.

Teniendo en cuenta una casa de 20 filas con 20 columnas serian 400 casillas, con 20 obstáculos y 10 basuras.

Misma cantidad de niños y robots (5):

Priorizando los niños	Priorizando las basuras
89 turnos	212 turnos
104 turnos	404 turnos
69 turnos	233 turnos
59 turnos	126 turnos
62 turnos	245 turnos

Más robots (10) que niños (5):

67 turnos	72 turnos
101 turnos	59 turnos
71 turnos	88 turnos
25 turnos	69 turnos
26 turnos	116 turnos

Más niños (10) que robots (5):

173 turnos	283 turnos
82 turnos	307 turnos
115 turnos	352 turnos
110 turnos	549 turnos
111 turnos	471 turnos

Podemos decir que la mejor estrategia es buscar a los niños primero porque optimiza mejor el tiempo a medida que existan mas niños que generen más basuras lo que puede provocar que el segundo modelo, que , aunque para estos ejemplos no sucedió, cuando se añaden muchos mas niños se puede quedar simulando muchísimo tiempo sin llegar a tener la habitación limpia. Es verdad también que para la simulación force un aumento de la probabilidad de generar basura para observar mejor los casos.