Proyecto de Simulación y Programación Declarativa Agentes

Reinaldo Barrera Travieso C411

Orientaciones Generales

Cada estudiante debe realizar la implementación del problema presentado (seleccionado por el estudiante y conociendo que los proyectos son individuales y que las soluciones deben ser diferentes en cada caso).
Esta implementación se debe realizar en Haskell y la solución debe estar en un repositorio de Github. La URL del proyecto debe entregarse por correo electrónico al profesor de conferencia de Simulación (yudy@matcom.uh.cu) con copia a la profesora de conferencia de Programación Declarativa (dafne@matcom.uh.cu), antes del viernes 28 de enero a las 12:00 de la noche. Junto a la implementación, en el proyecto de Github, debe estar presente el informe del trabajo (un documento en formato pdf).

El informe de trabajo debe contener los siguientes elementos:

- Generales del Estudiante
- Orden del Problema Asignado
- Principales Ideas seguidas para la solución del problema
- Modelos de Agentes considerados (por cada agente se deben presentar dos modelos diferentes)
- Ideas seguidas para la implementación
- Consideraciones obtenidas a partir de la ejecución de las simulaciones del problema (determinar a partir de la experimentación cuáles fueron los modelos de agentes que mejor funcionaron)

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 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 cuadricula de 3 por 3 hay un solo niño, entonces, luego de que él se mueva aleatoriamente, una de las casillas de la cuadricula anterior que esté vaca 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 pude 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 vacia, 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 vaciás no están sucias.

Principales Ideas seguidas para la solución del problema

Para este problema se inicia con un tablero de NxM, donde sus dimensiones son múltiplo de tres. Cada elemento del tablero representa un posiciona de la casa, el cual almacena 6 valores booleanos distribuidos de la siguiente manera:

- Dirty : Es verdadero si la casilla posee suciedad.
- Yard : Es verdadero si las casilla constituye un corral.
- Boy: Es verdadero si en la casilla se encuentra un niño.

- Robot: Es verdadero si en la casilla se encuentra un robot.
- Object: Es verdadero si en la casilla se encuentra un objecto
- Load: Es verdadero si un robot esta cargando a un niño.

Inicialmente y de forma aleatoria se pasa a general los elementos del estado inicial de la simulación. Primero de generan los corrales que van a ser igual a la cantidad de niños de la simulación. Para este caso primero se coloca un primero elementos de forma random y las restante cumpliendo la propiedad que sean adyacente a una casilla que ya sea corral. Seguido de esto, se pasa a general los restantes elementos de forma aleatoria en casillas totalmente vacías.

Una vez que sean colocadas todos los elementos iniciales, se pasa a general de forma aleatoria toda la suciedad que se tiene como estado inicial. Para esto se generan un valor aleatoria entre los el 40% y el 90% de las casillas que hasta el momento se encuentran vacías.

Después de completar el estado inicial se comienza con la simulación, la cual se realiza por tiempos, donde en cada tic del reloj se realiza una pasada por todo el tablero donde cada Agente realiza una opción o opta por no hacer nada ya sea porque no quiere o porque no puede general movimientos validos. Para este paso se manejan dos tablero, un tablero que representa el estado anterior y un tablero nuevo completamente vació, donde se va a ir actualizando los nuevos movimientos.

Comencemos por los los nos agentes, los cuales no realizan ninguna acción durante la simulación, a no ser que un agente lo modifique. En este caso encontramos a los objectos, la suciedad, y las casillas que representan a los corrales.

Entre los agentes encontramos a los niños, los cuales operaran de la siguiente forma:

- Si el niño se encuentra atrapado (esta dentro de un corral o un robo lo tienen cargado) no realiza ninguna acción.
- Por otro lado de poder moverse se escoge de forma aleatoria un valor que se encuentra entre 1 y 10. En caso de ser el valor mino el niño decide no moverse. En este caso existe una probabilidad de 0.1 de que el niño decida no moverse.
- Para el caso de que se decida continuar, se busca de todas las casillas adyacentes cuales son validas para futuros movimientos, y de todas las posibilidades existente se escoge una de forma aleatoria.
- Para el caso que se decida empujar un objecto se chequee si el desplazamiento es valido para luego proceder.
- Si ninguna de las opciones anteriores son validas, se siguen generando posiciones aleatorias hasta llegar a una correcta, a no ser que ya se halla verificado que no es posible ningún movimiento. En este ultimo caso se pasa para otro agente.

Los niños son unos de los agentes que pueden realizar cambios en el estado del ambiente, pues ademas de pode mover objectos son los encargados de hacer aparecer la suciedad en la casa sin contar la que había inicialmente. La cantidad de suciedad es escogida y colocada al final del que todos los agentes realicen sus operaciones. Esta depende de la cantidad de niños que existen en las cuadriculas de 3x3, por cual motivo se escoge un tablero inicial con dimensiones que sean múltiplo de 3, pues de no ser así, un mismo agente pudiera pertenecer hasta 9 cuadriculas.

El otra agente son los robot los cuales tienen como tarea arreglar los desastres provocados por los niños. Por esta razón los reboses tienen el siguiente esquema:

- Si el robo se encuentra cargando un niño y se encuentra en un corral, procede a dejarlo, para que así el niño no pueda seguir provocado suciedad.
- Si no se encuentra en un corral pero posee alguno en los adyacentes, incluyendo hasta dos casilla pues opta por desplazarse hacia ahí, para así poder dejarlo en el siguiente estado.
- En el caso que exista basura el robo procede a limpiar
- Si ninguna de las anteriores ocurre, el robot analiza todas las casillas adyacente para ver todas las posibilidades que tiene para moverse. Se escoge esta opción porque como se encuentra cargando un niño, el desplazamiento es la mejor opción para acercase un corral y de paso el niño no provoca suciedad.

Esta última opción fue escogida ya que como el robot tenia un niño cargado puede desplazarse hasta dos casilla, lo que le facilita un mayor desplazamiento.

- Por el orden a seguir la otra opción que tiene en cuenta el robot es analizar sus vecino en busca de la existencia de algún niño, así puede realizar una captura de el para posteriormente colocarlo en un corral.
- De no se así, pasaría a buscar algún adyacente con basura, si no pues de las casillas adyacente disponibles para su desplazamiento escoge alguna aleatoriamente.

Si después de todo estos punto el robo no encuentra ninguna opción pues significa que el robo de puede desplazare.

Resultado Obtenidos

Para la muestra de los resultado se tomara varios experimentaciones y analicemos el tiempo que toma en completar el 60%.

Intento	Time
1	18
2	10
3	38
4	0
5	81

6x6 – (5, 5, 5)

Intento	Time
1	5
2	0
3	1
4	0
5	2

6x6 – (5, 10, 5)

Intento	Time
1	5
2	28
3	48
4	15
5	56

15x24 – (15, 25, 30)

Intento	Time
1	388
2	1288...
3	385
4	1123...
5	684

15x24 – (50, 25, 30)

En las tablas se puede ver una distribución desigual, pues para los diferentes caso probados el tiempo de finalización varias , por lo que no se puede llegar a afirmar que existe algún patrón fijo. Solo podemos destacar que mientras mayor sea la diferencia entre los niños con respecto a los robot mas tiempo se toma la simulación para finalizar .

El programa

El código esta escrito en haskell , los parámetros se están definidos a principio del fichero main, para ejecutar se pueden usarlas siguientes vías:

- runhaskell Mian.hs - para ejecutarlo directo en la consola
- ghc Mian.hs | ./Mian – para compilar y ejecutar el programa

El resultado se muestra en consola usando caracteres ASCII en forma de tablero a continuación se describe el significado de cada simbologías:

- "XB " = Niño + Basura
- " X " = Basura
- "XbR" = Robot(Niño) + Basura
- "XBR" = Robot + Niño + Basura
- "XR " = Robot + Basura
- "bR " = Robot(Niño)
- "BR " = Robot + Niño
- " R " = Robot
- " B " = Niño

- " O " = Objeto
- "[B]" = Niño en Corral
- "[ ]" = Corral
- "[Z]" = Robot + Niño en Corral
- "[R]" = Robot en Corral
- "[r]" = Robot(Niño) en Corral

El símbolo + significa que ambos elementos estan en la misma casilla. Para el caso () significa que el robo esta cargando a un niño y los [] que se trata de un corral.