

Informe Escrito

Autor

Nombre y Apellidos	Grupo	Correo
Ariel Plasencia Díaz	C-412	arielplasencia00@gmail.com

Orientación del 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 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*, *corrales* y *agentes*, estos últimos son llamados robots de casa. A continuación se precisan las características de los elementos del ambiente:

Obstáculo

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 moverlos. No pueden ser movidos a 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, es decir, no tiene suciedad, no hay un 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 3×3 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 celdas. 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 en 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.

Principales ideas seguidas para la solución del problema

Para llevar a cabo las simulaciones del problema creamos una matriz de $N \times M$ donde están representados (con o con un conjunto de letras) todos los posibles elementos que nos presenta el problema.

Símbolo	Descripción
[]	casilla vacía
[R]	casilla con robot
[C]	casilla con niño
[T]	casilla con suciedad
[H]	casilla con corral
[X]	casilla con obstáculo
[RT]	casilla con robot limpiando una basura
[RC]	casilla con robot cargando un niño
[RH]	casilla con robot en el corral
[CH]	casilla con un niño en un corral
[RCH]	casilla de un robot dejando a un niño en el corral
[RCT]	casilla con robot cargando a un niño y limpiando una basura

En un inicio se generan encima de la matriz o tablero todos los parámetros del ambiente, es decir, la cantidad de turnos con que cambia el ambiente, los robots, los niños, las basuras, los corrales y los obstáculos. Todos estos datos son insertados previamente por el usuario. Cabe mencionar que las dimensiones del tablero también son personalizadas por el usuario. Además, se escoge el tipo de modelo de agente para la realización de la simulación.

También, realizamos una serie de suposiciones antes de pasar a la solución del problema, sin contradecir la esencia de la orden asignada, con el objetivo de desambiguar algunos puntos en la orden y lograr una correcta concepción del problema. A continuación se listan las principales ideas asumidas:

1. Se interpreta como variación aleatoria el movimiento de los niños y la generación de basura por ellos, los restantes elementos del ambiente no cambian.
2. En el ambiente interviene un único agente a la vez.
3. El robot, los niños y los obstáculos pueden moverse en dirección horizontal o vertical.
4. Cuando decimos que un objeto se encuentra en una cuadrícula de 3×3 asumimos que está en el centro de la misma.
5. Un robot puede pasar solo o cargando un niño por cualquier otra casilla del tablero excepto por una que contenga un obstáculo.

La simulación culmina cuando se alcanza una cantidad de turnos máximas o cuando la habitación posee el 60% de las casillas vacías limpias como indica la orientación del problema.

Modelos de agentes considerados

En el caso de los niños, modelamos su movimiento de forma aleatoria y con una probabilidad de 50% de querer moverse y de generar suciedad.

Para el caso de los robots de casa, usamos dos modelos, ambos reactivos, pero con un gran grado de pro-actividad diferenciándose únicamente en la prioridad de sus categorías. Cabe destacar que ambos tienen una arquitectura de *Brooks*. Las categorías son las siguientes:

1. Si la posición actual está sucia, se limpia.
2. Si el robot está libre y puede alcanzar algún niño, se mueve hacia el niño más cercano.
3. Si el robot puede alcanzar alguna basura, se mueve hacia la basura más cercana.
4. Si el robot carga un niño y puede alcanzar algún corral, se mueve hacia el corral más cercano.
5. Se queda en su lugar.

Prioridad de cada modelo:

- Priorizan primero los niños: $1 \rightarrow 2 \rightarrow 4 \rightarrow 3 \rightarrow 5$
- Priorizan primero las suciedades: $1 \rightarrow 3 \rightarrow 2 \rightarrow 4 \rightarrow 5$

Ideas seguidas para la implementación

La implementación fue realizada en `Haske11`. Se implementaron todos los elementos presentes en el problema, tanto el ambiente como los objetos como los modelos de agentes que participan. A continuación mostramos la tabla anterior actualizada con las constantes utilizadas:

Símbolo	Descripción	String
[]	casilla vacía	empty
[R]	casilla con robot	robot
[C]	casilla con niño	child
[T]	casilla con suciedad	trash
[H]	casilla con corral	corral
[X]	casilla con obstáculo	obstacle
[RT]	casilla con robot limpiando una basura	robot-trash
[RC]	casilla con robot cargando un niño	robot-child
[RH]	casilla con robot en el corral	robot-corral
[CH]	casilla con un niño en un corral	child-corral
[RCH]	casilla de un robot dejando a un niño en el corral	robot-child-corral
[RCT]	casilla con robot cargando a un niño y limpiando una basura	robot-child-trash

El código fuente se encuentra totalmente dentro de la carpeta `src` y consta de seis ficheros:

- *Main.hs*: Script encargado del inicio de la simulación.
- *Board.hs*: Archivo relacionado con las funciones del tablero.
- *Child.hs*: Fichero que engloba todos los métodos relacionados con los niños.
- *Robot.hs*: Script que tiene los movimientos de los robots.
- *Random.hs*: Archivo utilizado para la generación de números aleatorios.
- *Utils.hs*: Fichero que posee los métodos auxiliares.

Para representar el ambiente se usa una matriz de $N \times M$ de tipo lista de listas donde N corresponde con la cantidad de filas y M la cantidad de columnas. Para crear el ambiente se ubica primero el corral, ya que este requiere de casillas adyacentes, de esta manera se garantiza su obtención. Luego se ubican los obstáculos, que no pueden dejar ninguna zona de ambiente inaccesible para el robot. Luego se ubican los niños, los robots y las basuras en ese orden y de manera aleatoria.

Una vez creado el tablero se desplazan los niños. Estos se mueven cada t turnos, con una probabilidad de $\frac{1}{2}$ para cada niño. Si en la dirección que este se va a mover existe un obstáculo, se busca la primera casilla libre en la dirección del movimiento a realizar para poder desplazar todos los obstáculos en esta dirección, en caso de que no exista, el niño mantiene su posición. En caso de que el movimiento del niño sea satisfactorio se procede a generar basura, esta se crea buscando todas las subcuadrículas de 3×3 a las que pertenecía el niño y por cada subcuadrícula se cuenta la cantidad de niños que hay dentro de las nueve celdas y la cantidad de basura que hay en estas. Dados estos números se sabe el número máximo de basura que se puede generar, en caso de que sea mayor que 0, se itera por encima de cada celda de la subcuadrícula y se genera basura con probabilidad $\frac{1}{2}$, una vez se acaban las celdas en la subcuadrícula o se llega al máximo de basura permitido, se pasa a la siguiente subcuadrícula.

Con los niños desplazados, se mueven los robots pertenecientes al tablero resultante. Con este fin cada estado de un robot tiene un objetivo en su movimiento. Este objetivo cambia con respecto al modelo seleccionado en un comienzo por el usuario para los estados en los que un robot está solo, está en un corral vacío o está en un corral con un niño pero ya lo dejó en un turno anterior. Para cumplir el objetivo correspondiente a cada estado en ambos modelos se realiza una búsqueda de la casilla que satisface la condición de ser la celda deseada más cercana, para esto se implementó una matriz de distancia (con el algoritmo de *Depth First Search*) que se expande desde el robot analizado en ese instante, en donde no puede atravesar los tipos de celdas que no pueden poseer un robot encima especificados en la orientación y se toma de los objetivos el que menor distancia con respecto al robot tenga. Con esta misma matriz de distancia se genera el camino que debe tomar el robot y así saber qué dirección debe tomar este en el turno. Con la trayectoria que debe seguir el robot se realiza el movimiento para acercarlo a la celda deseada. En caso de que dicha celda no exista, o no sea accesible, el robot busca otro objetivo secundario de menor importancia, en caso del modelo que prioriza la recogida de niños, su objetivo secundario es la basura, y en el modelo que su casilla deseada es la basura, al no poder llegar a esta, buscará a un niño. Si en la trayectoria un robot coincide con una basura, este la limpia, y si coincide con un niño lo carga. Si un robot tiene un niño cargado su objetivo principal en ambos modelos es dejarlo en un corral y su objetivo secundario es la basura. En caso de que un robot no pueda cumplir su objetivo primario ni secundario, este mantendrá su posición sin realizar ninguna acción.

La simulación correrá mientras la cantidad de turnos transcurridos sea menor que el máximo fijado (por defecto es 1000) y mientras la cantidad de casillas limpias sea menor al 60% de las casillas ensuciadas en el tablero.

Consideraciones obtenidas a partir de las simulaciones

A continuación los resultados de un conjunto de simulaciones partiendo de varios juegos de parámetros distintos para conformar los ambientes y se recogen los resultados por cada modelo, atendiendo a la cantidad a la cantidad de turnos demorados y al porcentaje de suciedad en el ambiente final.

Los resultados alcanzados para el primer modelo (robots que prefieren los niños) fueron:

Filas	Columnas	Suciedad inicial	Obstáculos	Niños	Robots	t	Turnos demorados	Suciedad final (%)
10	10	5	0	4	4	1	30	15
10	10	5	0	4	4	3	47	7
10	10	10	5	6	3	2	28	6
10	10	10	5	6	3	4	21	8
10	10	15	10	8	5	3	31	1

Los resultados alcanzados para el segundo modelo (robots que prefieren las basuras) fueron:

Filas	Columnas	Suciedad inicial	Obstáculos	Niños	Robots	t	Turnos demorados	Suciedad final (%)
10	10	5	0	4	4	1	148	4
10	10	5	0	4	4	3	296	3
10	10	10	5	6	3	2	81	6
10	10	10	5	6	3	4	73	10
10	10	15	10	8	5	3	578	28

Conclusiones

Podemos concluir que la mejor estrategia es la primera ya que a medida que aumentan los robots se va disminuyendo el por ciento de las celdas sucias aunque aumenta la cantidad de turnos. Por el contrario, para el segundo agente mientras mayor sea la cantidad de los parámetros aumenta los turnos demorados y la suciedad final.

Enlace a GitHub

Para acceder al enlace en GitHub pulse [aquí](#).