

Universidad de La Habana

FACULTAD DE MATEMÁTICA Y
COMPUTACIÓN

ROBOTS DE CASA

Proyecto Final de Simulación y Programación Declarativa

Estudiante:

Laura Brito Guerrero C-412

0.1. Orientación

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 de 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:

1. Obstáculos: estos ocupan una única casilla en el ambiente. Estos 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.
2. 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.
3. Corral: el corral ocupa casillas adyacentes en número igual al del total del 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.
4. 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 en el corral, no hay un Robot de Casa), y aleatoriamente (puede que no ocurra movimiento), a una de las casillas 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 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.

5. Robots 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 a 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 la 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 por ciento de las casillas no están sucias.

0.2. Idea a seguir

Se tiene un tablero con rango (h, w) donde h es la altura y w es el ancho. Una vez conocidos estos valores y a t (a cada t iteraciones los niños pueden o no moverse o dejar suciedad en casillas cercanas), se crea todo el espectro de objetos que tienen cabida en la misma : robots, niños, corral, obstáculos, suciedad(puede o no aparecer al inicio). Estos objetos tienen un número aleatorio en cada ejecución del programa. Una vez creado el tablero con todos los datos, se realizan las acciones posibles en los agentes (robots de casa). Las acciones a realizar por los agentes son las siguientes:

```

childDown(x,y) → in(x,y) and childIn(r) and insideCorral(x,y) and
downPossible(x,y,posCorral) = 0
corralIn(x,y) → in(x,y) and childIn(r) and insideCorral(x,y) = 1 + distance
((x,y) posCorral)
corralBound(x,y) → in(x,y) and childIn(r) and posCorralBound((x,y)
(bounds corral)) = 2 + distance ((x,y) posCorral)
clearRule(x,y) → in(x,y) and childIn(r) and dirt(x,y) = 3 + distance ((x,y)
posCorral)
goCorral(x,y) → in(x,y) and childIn(r) = 4 + distance ((x,y) posCorral)
upChildRule(x,y) → in(x,y) and alone(r) and child(x,y) = (h * w) + 5
clearAloneRule(x,y) → in(x,y) and alone(r) and dirt(x,y) = (h * w) + 6
goRadar(x,y) → in(x,y) and alone(r) = (h * w) + 7 + distance ((x,y)
objectRadar)
outCorral(x,y) → beforeInsideCorral(r) and in(x,y) and alone(r) and
outsideCorral(x,y) = (h * w) + 8 + distance((x,y) bounds corral)

```

Donde:

1. in(x,y) : el robot se encuentra en la posición (x, y)

2. $\text{childIn}(r)$: el robot r esta cargando un niño
3. $\text{insideCorral}(x,y)$: (x,y) está dentro del corral
4. $\text{outsideCorral}(x,y)$: (x,y) está fuera del corral
5. $\text{downPossible}(x,y,\text{posCorral})$: $(x,y) == \text{posCorral}$
6. $\text{posCorralBound}((x,y), \text{boundsCorral})$: $\text{contains}(x,y) \text{ boundsCorral}$
7. $\text{dirt}(x,y)$: la casilla (x,y) está sucia
8. $\text{beforeInsideCorral}(r)$: en la posición actual se encuentra dentro del corral
9. $\text{alone}(r)$: el robot r no esta cargando a ningún niño
10. objectRadar : niño fuera del corral o suciedad más cercana para el robot

A través de estos predicados se aplican las reglas que interpretan las acciones de los agentes otorgándoles una prioridad. De esta manera en cada paso se conoce todas las acciones óptimas de cada agente, y se toma de todas ellas la de menor prioridad posible.

Estas reglas se analizan por cada nueva posición adyacente posible de cada agente, para de esta forma tener todas las posibles acciones del mismo.

Cuando el robot está cargando a un niño se le consideran sus acciones prioritarias con respecto a los que están sin niños. Es importante señalar que a los robots que cargan niños le es posible dar dos pasos seguidos en cada turno, significa que a estos se les analiza por cada $(x0,y0)$ nueva, 8 nuevas posiciones adyacentes más a $(x0,y0)$.

Una vez que se conozcan por cada agente todas las posibles nuevas posiciones, se procede a concatenar todas las acciones posibles de todos los agentes. Las nuevas posiciones de cada agente se obtiene mediante un algoritmo de ordenación tal que:

1. El agente i solo puede realizar una acción,
2. Se va tomando siempre la nueva posición que mayor prioridad tenga tal que no se repita esa nueva posición con las ya seleccionadas como óptima.

De esta manera se garantiza que no se repitan posiciones y que cada agente vaya a realizar una sola tarea. Una vez analizado el ambiente y el desplazamiento de los robots hacia las nuevas posiciones se actualiza la información del ambiente y se da paso a la próxima ronda.

0.3. Inteligencias Artificiales implementadas

0.3.1. Comportamiento deductivo:

Se esclarece en el tópico anterior con la explicación de las reglas y los predicados, siempre tomando la de mayor prioridad posible.

Dicho comportamiento se encuentra basado en lógica, el proceso de toma de decisión de un agente es modelado por un conjunto de predicados (reglas de deducción).

0.3.2. Arquitectura de capas verticales de dos pases (InterRap)

Cada capa tiene asociada una base de conocimiento (los agentes y su ambiente a diferentes niveles de abstracción). La base de conocimiento de mayor nivel representa los planes y acciones de otros agentes en el ambiente. La base de conocimiento intermedia representa los planes y acciones del agente mismo. La base mas baja representa la información *real* sobre el ambiente.

La capa de conducta

Aborda el comportamiento reactivo del agente (se encuentra definida en las reglas, cuando un agente *i* no tiene un propósito definido sino que busca qué acción cercana es la más óptima, no obstante, todo este comportamiento se visualiza en la concatenación de todas las posibles acciones de todos los agentes).

La capa cooperativa

Cuando se realiza el análisis de las nuevas posiciones dándole importancia a la prioridad de todos los agentes y escogiendo a través de una negociación, el conjunto de acciones óptimas.

La capa de planeamiento

Como se ven en las reglas, los agentes si no tienen una acción inmediata, analiza futuras acciones, por lo que se acerca a su *nuevo objetivo*. Se puede observar en las reglas *goCorral(x, y)*, *goRadar(x, y)*.

0.4. Detalles para ejecutar el programa

Se considera que las entradas siempre serán números enteros. Si desea aumentar la cantidad de rondas debe cambiar el valor del penúltimo parámetro de `init_` en `Main.hs`.

0.5. Ejecución del programa

En el directorio `code/src`:

```
>> stack ghci Main.hs
```