

Willy en el espacio

Jorge Zafra Palma
Diego Rodríguez Riera

4 de mayo de 2018

Índice

1. Recopilación de información del entorno	3
2. Determinación del próximo movimiento	4
2.1. Movimiento willy	4
2.2. Percepción en la casilla actual en la que está willy	5
3. Eliminación del alienígena	6
3.1. Percepción	6
3.1.1. Reglas involucradas	7
3.2. Respuesta	7
3.2.1. Reglas involucradas	7
4. Detalles de suplementarios	8

1. Recopilación de información del entorno

En nuestro programa almacenamos información de las casillas que se visitan, además de las percepciones que se reciben cuando Willy está situado sobre esas casillas. También almacenamos, en caso de completa seguridad, la posición en la que se encuentra el alien.

Para almacenar la información de las casillas visitadas utilizamos la plantilla **field** en la que tenemos slot que almacenan la coordenada x y coordenada y de la casilla y dos slot para ver si se recibe la percepción noise y la percepción gravity, utilizando para esto los valores true y false.

```
(deftemplate field
  (slot x)
  (slot y)
  (slot noise (allowed-symbols true false) (default false))
  (slot gravity (allowed-symbols true false) (default false))
)
```

Para almacenar la posición tanto de Willy como del alien utilizamos la plantilla **position** donde tenemos un slot name para indicar el elemento del que estamos almacenando la información así como la coordenada x y la coordenada y .

```
(deftemplate position
  (slot name (default ?NONE))
  (slot x (type INTEGER) (default 0))
  (slot y (type INTEGER) (default 0))
)
```

- Los hechos de las casillas son afirmados en el modulo **Hazards**. Las reglas de este modulo analizan la casilla en la que esta ahora mismo *Willy* y según las percepciones que se reciben se afirma el hecho **field** con unos valores u otros.
- El poseer la información de las casillas es utilizado principalmente para dirigir el movimiento, siempre intentando visitar antes casillas que no hayan sido visitadas anteriormente. También esta información es utilizada para la localización del alien.

2. Determinación del próximo movimiento

En nuestro programa almacenamos información de las casillas que se visitan, además de las percepciones que se reciben cuando willy está situado sobre esas casillas. También almacenamos, en caso de completa seguridad, la posición en la que se encuentra el alien.

Para almacenar la información de las casillas visitadas utilizamos la plantilla `field` en la que tenemos slot que almacenan la **coordenada x** y **coordenada y** de la casilla y dos slot para ver si se recibe la percepción `noise` y la percepción **gravity**, utilizando para esto los valores `true` y `false`.

Los hechos de las casillas son afirmados en el modulo **Hazards**. Las reglas de este modulo analizan la casilla en la que esta ahora mismo willy y según las percepciones que se reciben se afirma el hecho **field** con unos valores u otros.

El poseer la información de las casillas es utilizado principalmente para dirigir el movimiento, siempre intentando visitar antes casillas que no hayan sido visitadas anteriormente. También esta información es utilizada para la localización del alien

2.1. Movimiento willy

En nuestro programa tenemos un modulo exclusivamente para el movimiento de willy.

Ninguna percepción en la casilla actual en la que está willy

En este caso la prioridad primero estará en desplazarse hacia alguna casilla circunstante no visitada. El conjunto de reglas que se encargan de esto son `moveWilly` "Direccion" tomando dirección los valores North, South, East y West.

Estas reglas tendrán una prioridad superior a las del movimiento hacia casillas visitadas.

Por ejemplo, la regla de **moveWillyNorth** se activará si la casilla que está justo encima de la posición de willy, aún no está visitada. En este caso se realizará el movimiento hacia esa dirección actualizando la posición en la que se encuentra willy con la coordenada y incrementada en una unidad.

En caso de que haya más de una casilla circunstante sin visitar, se activaran más de una regla de `moveWilly`. La elección de cual ejecutar la decide el programa. Una vez ejecutada cualquiera de estas reglas se pasará a sacar de la pila este modulo, ya que una vez realizado un movimiento hay que analizar la casilla a la que se ha desplazado y de esto se encarga otro modulo.

En el programa cada vez que se ejecuta cualquiera de estas 4 reglas, se almacena la dirección contraria a la que se desplazo en el hecho `movimientos-contrarios`,

es decir, si se ejecuto la regla **moveWillyNorth** en movimientos-contrarios se almacenara south. Esto es utilizado para el caso en el que willy no tenga ninguna casilla circunstante no visitada a la que ir.

En el caso de que no haya ninguna percepción en la casilla y tampoco se tenga ninguna casilla circunstante no visitada, lo que se activaran son las reglas **retroceder'dirección'** tomando dirección los valores Norte, Sur, Este y Oeste. La finalidad de estas reglas es básicamente ver cual es el ultimo valor que se inserto en **movimientos-contrarios** y se mueve hacia ese lugar, es decir, básicamente lo que hace es ir retrocediendo sobre sus pasos hasta que se vuelva a encontrar con casillas circunstante no visitadas en cuyo caso se volverán a activar las reglas **moveWilly**. El movimiento que acaba de ejecutar del hecho **movimientos-contrarios** es eliminado del hecho. Cuando se ejecute alguna de estas reglas no se saldrá de la pila ya que lo que se busca es la activación de algunas de las reglas **moveWilly** como se ha dicho antes. Se tiene una regla de retroceso por cada uno de los posibles últimos valores que pueda haber en el hecho **movimientos-contrarios**.

2.2. Percepción en la casilla actual en la que está willy

Cuando willy se mueve hacia una casilla en la que se detecta alguna percepción, la estrategia que se sigue es la de retroceder. De esto se encargan las reglas **backtrack**. El procedimiento sería el siguiente:

Previamente se habría ejecutado una regla **moveWilly** que habría desplazado a willy a una casilla no visitada. Ahora se analizaría esta casilla en el modulo **Hazards**. Si esa casilla presenta alguna percepción se afirma un hecho **warning** el cual permitiría la activación de las reglas de **backtrack**.

Por tanto cuando se entre al modulo de movimientos, las reglas **backtrack** tienen mas prioridad que el resto, por lo que serán las primeras en activarse para evitar que willy haga un posible movimiento hacia el peligro.

Para el retroceso se utiliza el mismo concepto que se indico antes, con el hecho **movimientos-contrarios**.

3. Eliminación del alienigena

3.1. Percepción

La percepcion del entorno es llevada a cabo por el modulo **HazardsModule**, este tiene tres reglas principales:

- blackHole
- alien
- blackHoleAndAlien
- noHazard

La seguridad se basa en el back-tracking, es decir, vuelve un paso atras si se encuentra con un peligro.

Sin embargo en el siguiente modulo, **AfterHazardsModule**, nos centramos en el posicionamiento del alienigena, para ello el sistema reconoce los siguientes patrones:

- Patrón 1

N A N

- Patrón 2

N A

S N

- Patrón 3

A

S N S

S

Donde:

S = Safe

N = Noise

A = Alien

Dichas reglas se basan en intentar encontrar agrupaciones de casillas con las mismas características que las expuestas, estas reglas estan duplicadas cuatro veces desfasadas 90 grados entre si.

Una vez detectada la formación asertan El alienigena en la posición correspondiente.

3.1.1. Reglas involucradas

```

HazardsModule:: blackHole_and_alien
HazardsModule:: blackHole
HazardsModule:: alien
HazardsModule:: noHazardmax
AfterHazardsModule:: locate_alien_middle_horizontal
AfterHazardsModule:: locate_alien_middle_vertical
AfterHazardsModule:: locate_alien_oblique_negative_alien_up
AfterHazardsModule:: locate_alien_oblique_negative_alien_down
AfterHazardsModule:: locate_alien_oblique_positive_alien_up
AfterHazardsModule:: locate_alien_oblique_positive_alien_down
AfterHazardsModule:: locate_alien_single_point_from_bottom
AfterHazardsModule:: locate_alien_single_point_from_top
AfterHazardsModule:: locate_alien_single_point_from_right
AfterHazardsModule:: locate_alien_single_point_from_left

```

3.2. Respuesta

Una vez localizado el alienigena, se pasa su eliminación.

Cuando Willy se encuentra en una coordenada cartesiana x o y iguales a el alienigena, este evalua en cual de las cuatro posiciones se encuentra (arriba, abajo, a la derecha o la izquierda del alienigena) y dispara en la dirección correspondiente

Finalmente, se comprueba (si se ha matado al alien) si hay mas peligros cercanos (atracción gravitatoria) si no lo hay, se eliminan todos los peligros, ya que solo hay un alien por mapa.

3.2.1. Reglas involucradas

```

AfterHazardsModule:: shoot_alien_right
AfterHazardsModule:: shoot_alien_left
AfterHazardsModule:: shoot_alien_up
AfterHazardsModule:: shoot_alien_down
AfterHazardsModule:: retract_warning_when_kill

```

4. Detalles de suplementarios

Los modulo se gestionan desde el modulo **myMAIN**, este va enfocando modulo a modulo en serie y en un orden preestablecido:

HazardsModule → **AfterHazardsModule** → **MovementModule**

Este se repite siempre que exista **repeat**, estos se concatenan asertando y eliminando instancias como **passToAfterHazardsModule** o **passToMovementModule** para asegurar el correcto orden de entrada de los distintos módulos.