RICOCHET ROBOTS

Diseño de Sistemas Inteligentes

Universidad Carlos III de Madrid

Adrián Borja Pimentel | Francisco Rodríguez Melgar

miércoles, 25 de abril de 2018

Contenido

Índice de tablas

Índice de ilustraciones

1. Introducción

Aquí habrá que poner algo

2. Dominio

La representación del juego en lenguaje pddl se encuentra en el fichero domain.pddl. Es una ontología sencilla donde se definen todos los componentes que intervienen en el juego y las reglas necesarias para que los robots se muevan siguiendo las reglas y el conteo de movimientos sea también coherente.

El aspecto más importante que se debe tener en cuenta en lo relativo al dominio, es la decisión de diseño de no representar las paredes, sino la adyacencia entre casillas. La existencia de una pared que separa las casillas C03 y C04 no se representa con un predicado de pared C03 C04, en vez de eso, para todas las casillas existen 4 predicados que indican que casilla esta adyacente en cada una de las cuatro direcciones, y en caso de que haya una pared, la casilla adyacente en esa dirección es ella misma. En el ejemplo anterior, C03 sería "adyacente" a C03 por la derecha, y C04 a C04 por la izquierda.

La razón de hacerlo así era la facilidad de escalabilidad, ya que de esta manera el dominio no dependia de las dimensiones del tablero del problema, y las reglas eran atómicas. Además, desde el planteamiento inicial se contaba con el desarrollo de un script que generara automáticamente el tablero de un problema desde un fichero que tuviera una representación simplificada de la información importante, por lo que el objetivo del dominio era ser universal a cualquier problema que siguiera las reglas originales del juego Ricochet Robots. Esto provocó que los archivos de problemas fueran realmente densos, pero gracias al script eso no ha sido un inconveniente.

2.1 Objetos y constantes

Para representar las entidades presentes en el juego se han utilizado los siguientes objetos:

- **Robot**. Los robots son las entidades más importantes del juego, ya que son los únicos elementos interactivos y variables.
- **Casilla**. Las casillas representan un espacio en el tablero, se distinguen por dos coordenadas formadas por una letra y un número.
- **Color**. En el juego, ciertos elementos poseen asociado un color que condiciona el resultado de ciertas acciones o estados. En el juego original eran rojo, amarillo, verde, azul y gris.

Además, están presentes 2 tipos de constantes que permiten gestionar como se producen los movimientos de los robots:

- **Orientación**. Cuyos valores posibles son slash y backslash. Permiten determinar la orientación en la que se coloca una barrera mágica.
- **Dirección**. Cuyos valores posibles son left, up, right, down y stop. Permiten codificar la adyacencia entre casillas y gestionar la dirección de los movimientos de los robots.

2.2 Predicados

Los predicados se utilizan para representar distintos estados o situaciones referentes a las entidades. En el dominio implementado se han utilizado los siguientes predicados:

Predicado	Parámetros	Descripción
At	Robot, casilla	Indica que un robot concreto está en una casilla determinada
colorRobot	Robot, color	Indica el color que posee un determinado robot
Moviendo	Robot, dirección	Indica el movimiento que está realizando un robot, si está parado será stop, y si no, será otra de las posibles direcciones.
movimientoEnCurso	Dirección	Dado que las acciones no permiten que haya más de un robot con un movimiento distintos de stop, este predicado se utiliza para indicar en que dirección es el movimiento que se está realizando.
Free	Casilla	Indica que la casilla puede ser ocupa por un robot
adyacente	Casilla1, casilla2, dirección	Indica para una casilla concreta, cual es la casilla adyacente en una determinada dirección.
hasBarrier	Casilla	Indica la presencia de una barrera mágica en la una casilla concreta
barraMagica	Casilla, color, orientacion	Indica el color y la orientación que tiene la barrera mágica de una determinada casilla
casillaObjetivo	Casilla, color	Indica que una casilla concreta es una casilla objetivo y su color, aunque en esta versión del planificador es una información irrelevante

2.3 Acciones

Para modelar el total de reglas del juego se han necesitado 7 acciones:

- **MoverACasillaLibre**, utilizada para que un robot comience a moverse en una dirección, entre sus restricciones más importantes, se encuentra que el movimiento en curso sea stop y que exista una casilla adyacente distinta de sí misma que esté libre. Los efectos más relevantes son los cambios en los predicados para que el robot siga moviéndose en la misma dirección hasta detenerse, y el incremento de movimientos realizados.

- **LetItGo**, dado el modelado del dominio en el que cada casilla tiene cuatro predicados de adyacencia, esta acción es la que se ejecuta para que el robot en movimiento continúe desplazándose. Su restricción más trascendental es que la siguiente casilla donde el robot va a estar, esté libre, y que la casilla origen no tenga una barrera de color que afecte al robot.
- **StopMiHada**, cuando el robot ha llegado a una casilla donde, según las reglas del juego, se detendría, se ejecuta esta acción que cambia los hechos de forma que se pueda iniciar otro movimiento o terminar la ejecución si se ha alcanzado la casilla objetivo. La restricción más importante es que la casilla a la que movería no esté libre, lo que significa que hay un robot obstaculizando el movimiento, o que la adyacencia en el sentido del movimiento sea la misma casilla de origen, lo que representa la existencia de una pared. Adicionalmente, no puede detenerse en una casilla donde haya una barra mágica.
- **ToLeft**, **ToHeaven**, **ToRight** y **ToHell**, son las acciones que se pueden ejecutar en caso de que en el desplazamiento del robot se encuentre una barra mágica que le afecte. Estas 4 acciones contemplan los 8 casos posibles que se pueden dar en función de la dirección que llevaba el robot y la orientación de la barra, siendo el efecto de cualquiera de ellas modificar los hechos en los que se basa el desplazamiento del robot. Adicionalmente, el cambio de dirección debe ser posible, es decir, no puede haber un obstáculo que obligara al robot a detenerse en la casilla de la barra mágica.

3. Experimentación

El objetivo de los experimentos estaba enfocado a estudiar empíricamente la complejidad temporal asociada al problema. Para ello se ha utilizado el planificador FastDownward y 9 experimentos donde se variaba la cantidad de robots y el tamaño del tablero.

Los resultados se encuentran en el anexo pruebas, y en la siguiente tabla se reflejan los tiempos obtenidos para cada experimento.

Nº Robot\Tamaño	8x8	8x16	16x16
1	0.00147033	0.00235309	0.00518803
3	3.10068	5.58564	12.4929
5	2.01516	5.99734	269.473