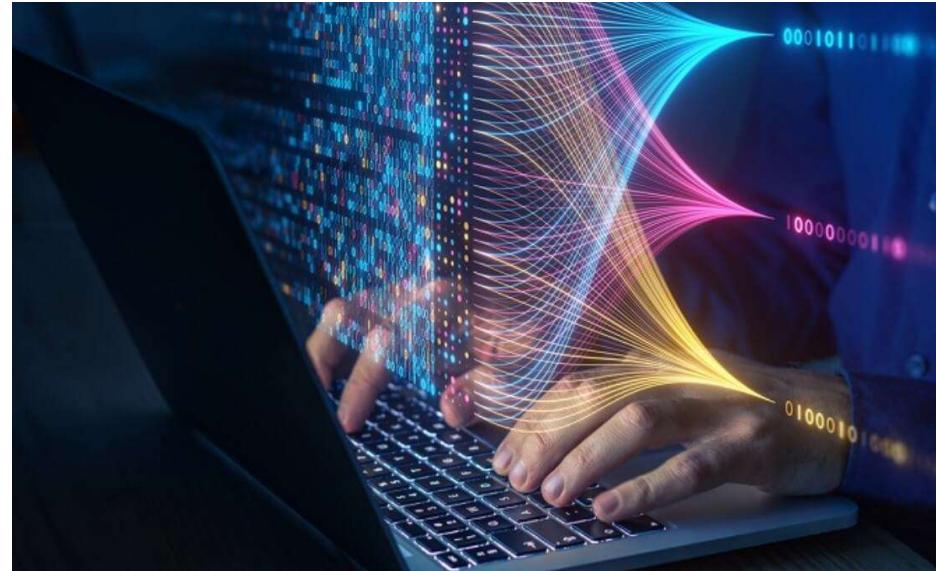


Técnicas de inteligencia artificial

Adriana Cervantes Castillo



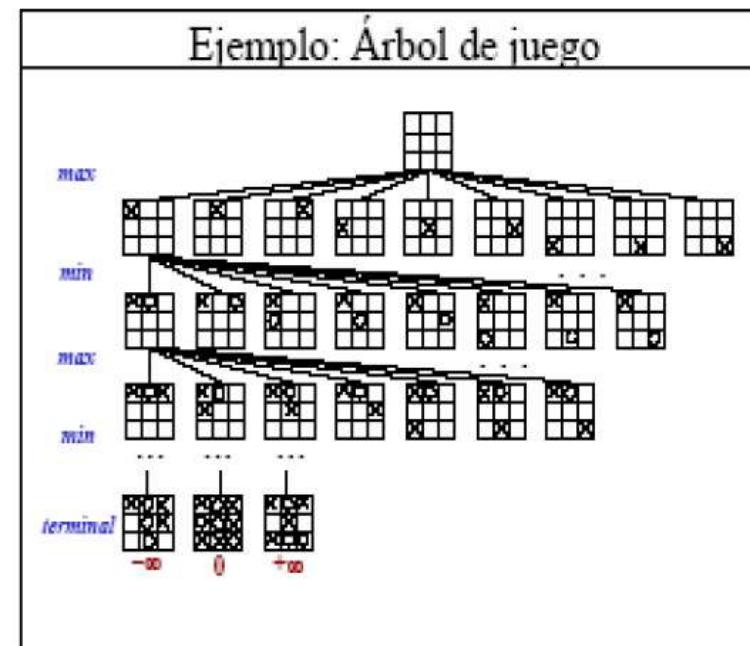
SEMANA 13: RESOLUCIÓN DE PROBLEMAS MEDIANTE BÚSQUEDA

Contenido

- Problemas de búsquedas
- El mundo de los bloques
- Búsqueda exhaustiva o a ciegas (No informadas)
- Búsquedas heurística o informadas

Problemas de búsqueda

Existen problemas de IA que pueden ser resueltos mediante búsqueda. En ellos se define un espacio de estados y unos operadores que permiten la transición entre estados, partiendo de un estado inicial y dirigiéndose a un estado objetivo. Entonces, el problema a resolver es encontrar el camino que lleve del estado inicial al estado objetivo a través del espacio de estados disponible. En cada estado se pueden aplicar una serie de operaciones para llegar a otro estado distinto, que puede ser el final o no.



Donde utilizar un algoritmo de búsqueda

- Juegos.
- Planeación de rutas.
- Problemas de Optimización matemática.
- Motores de búsqueda.
- Planificación automática.
- etc.

Mundo de los bloques

SOBRE(C, A)

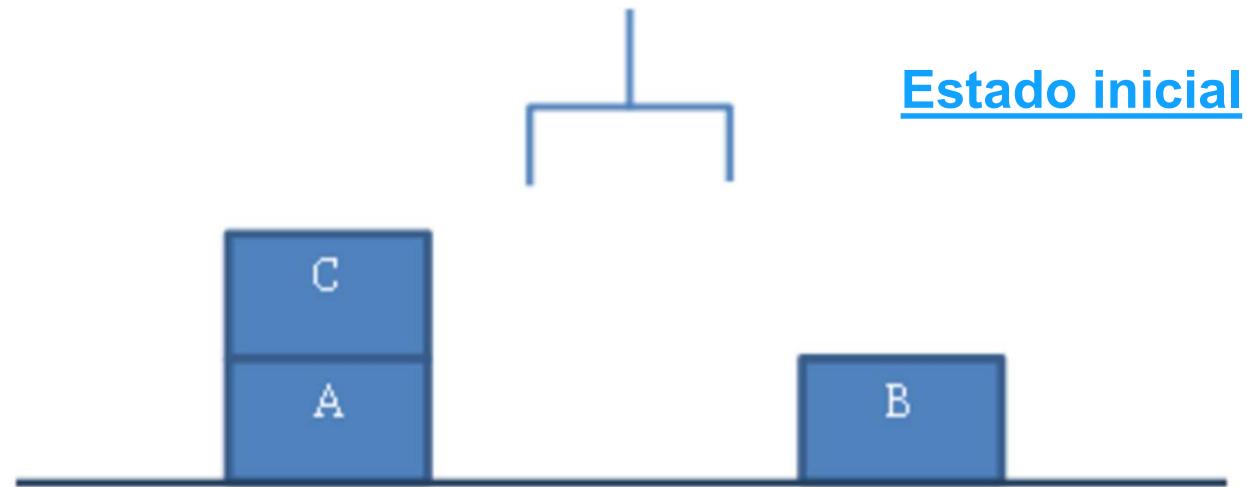
SOBRE_LA_MESA(A)

LIBRE(C)

MANO_LIBRE

SOBRE_LA_MESA(B)

LIBRE(B)

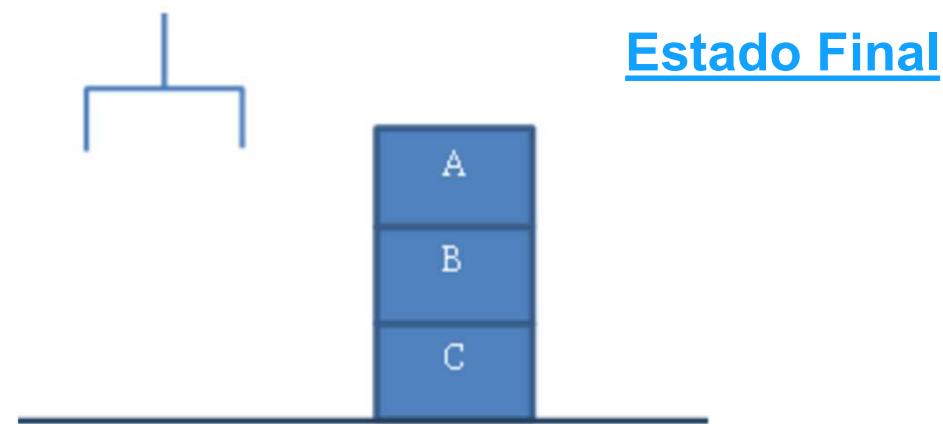


MANO_LIBRE

SOBRE(B,C)

SOBRE(A,B)

SOBRE_LA_MESA(C)



Operadores STRIPS (Estados, acciones)

- Precondiciones (P): prerequisitos que han de ser ciertos en la descripción del entorno para la realización de la acción.
- Borrar (B): fórmulas que se han de borrar de la descripción del entorno una vez aplicada la acción ya que dejan de ser ciertas.
- Añadir (A): fórmulas que se han de añadir a la descripción del entorno una vez aplicada la acción.

- Tomar(X) → Tomar el bloque X de la mesa con el brazo
- Poner(X) → Colocar el bloque X sobre la mesa
- Apilar(X,Y) → colocar el bloque X sobre el bloque Y
- Quitar(X,Y) → Quitar el bloque X que esta sobre el bloque Y

- Tomar (X)

P: LIBRE (X) , MANO_LIBRE

B: LIBRE (X) , MANO_LIBRE

A: tomando (X)

APILAR (X, Y)

P: Tomando (X) , LIBRE (Y)

B: Tomando (X) , LIBRE (Y)

A: MANO_LIBRE, SOBRE (X, Y) , LIBRE (X)

Poner (X)

P: Tomando (X)

B: Tomando (X)

A: SOBRE_LA_MESA (X) , LIBRE (X) , MANO_LIBRE

Quitar (X, Y)

P: MANO_LIBRE, LIBRE (X) , SOBRE (X, Y)

B: MANO_LIBRE, LIBRE (X) , SOBRE (X, Y)

A: Tomando (X) , LIBRE (Y)

- La solución es el siguiente plan de acciones:

Quitar(C,A)

poner(C)

tomar(B)

APILAR(B,C)

tomar(A)

APILAR(A,B)

La búsqueda consiste en encontrar un camino a través del conjunto de estados entre el estado o estados iniciales y el estado o los estados finales u objetivos. Esta búsqueda puede hacerse en dos direcciones:

- Hacia adelante (del inicio al fin).
- Hacia atrás (del fin al inicio).

Un factor que permite clasificar los algoritmos de búsqueda es la aplicación de la información disponible sobre el problema para su resolución.

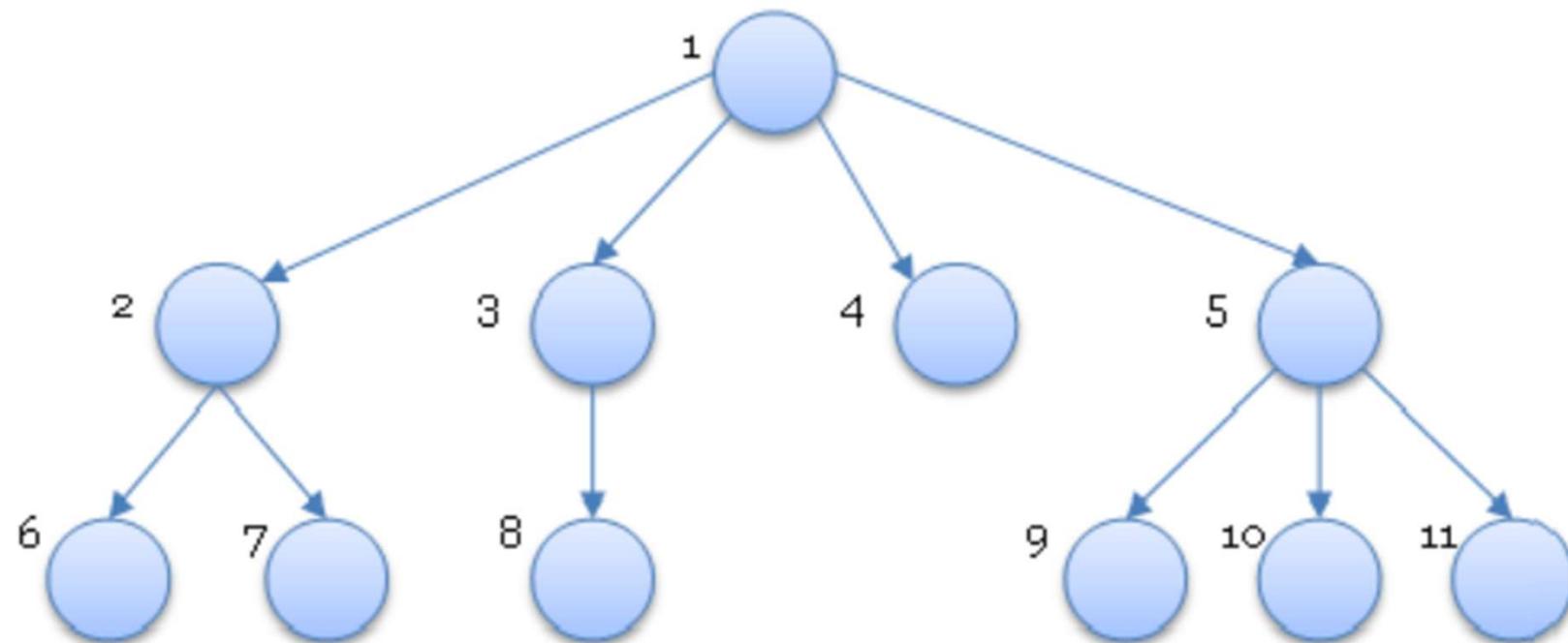
- **Búsqueda exhaustiva o a ciegas**: no se tiene en cuenta la posible localización del objetivo. Estos algoritmos no utilizan ninguna información del problema e ignoran hacia dónde se dirigen hasta que encuentran el objetivo.
- **Búsqueda heurística**: se tiene una estimación de lo que falta para llegar al estado objetivo. Los algoritmos utilizan información del problema para localizar el objetivo.

Búsqueda Exhaustiva o a ciegas

Búsqueda en amplitud

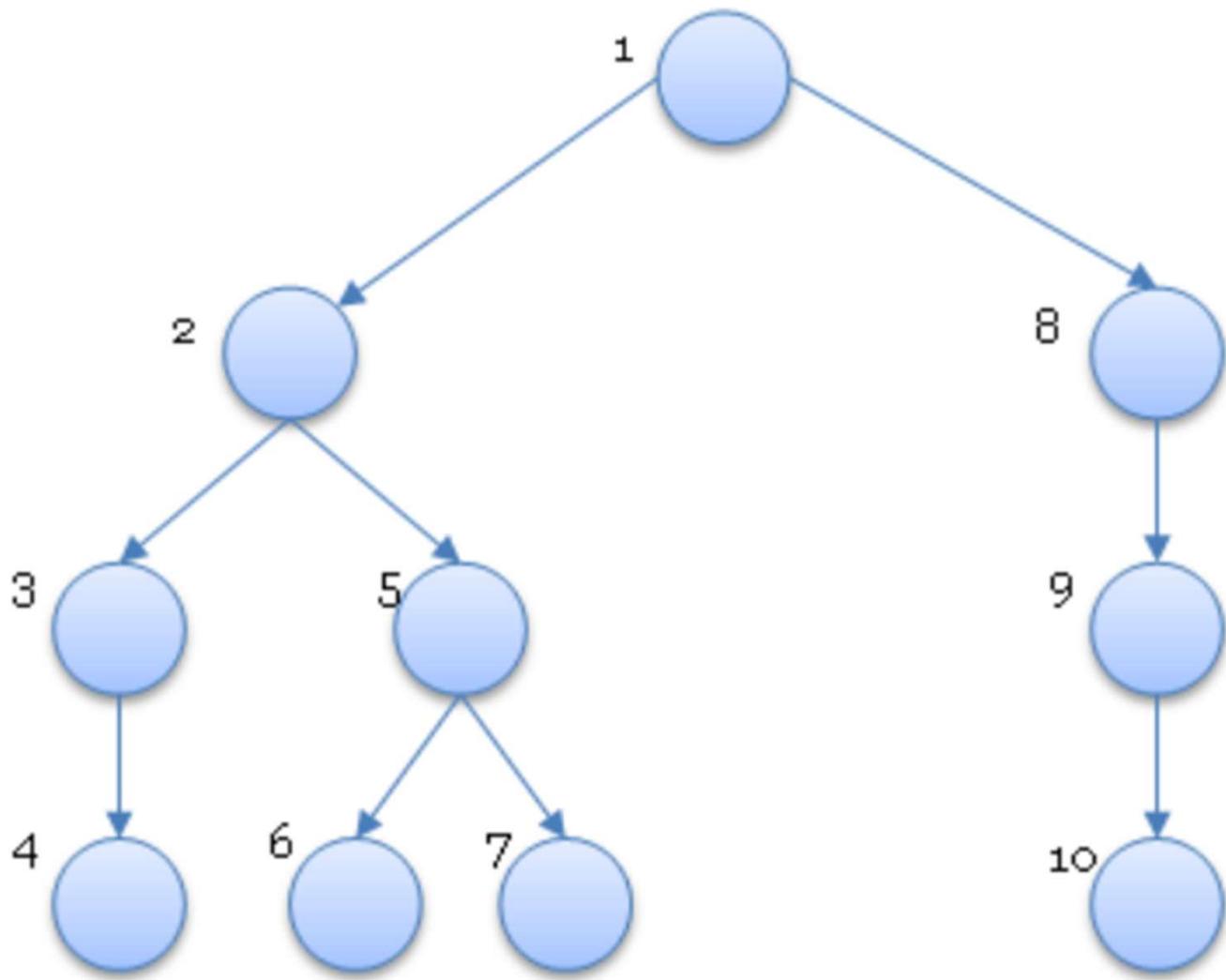
- A partir de un nodo se generan todos los hijos y, para cada hijo, todos los nietos, y así sucesivamente.
- Por tanto, cada nivel del árbol de estados se desarrolla completamente antes de desarrollar el siguiente nivel. Según los nodos de un nivel se generan, si no se ha llegado a la solución, se va al siguiente nivel para generar los nodos siguientes. Así, hasta encontrar el objetivo.
- El algoritmo sigue una estrategia FIFO (*First-In, First-Out*)

Búsqueda en amplitud



Búsqueda en profundidad

- Da prioridad a los nodos más profundos en el grafo de búsqueda
- Tras cada expansión o desarrollo de un nodo, **de los hijos que acaban de ser generados se selecciona uno para ser de nuevo desarrollado**. Esta exploración hacia abajo se desarrolla hasta que, por alguna razón, el proceso queda bloqueado. En tal caso, el proceso continúa desde el nodo más profundo que haya sido dejado atrás sin explorar.



- Expande el nodo más recientemente creado, siguiendo la estrategia LIFO (*Last-In, First-Out*).

Búsqueda heurística

Métodos en los que se aplica **información del problema** para encontrar la solución. A esta información disponible se le denomina información heurística.

Emplean una función denominada heurístico, que mide la proximidad de los nodos al objetivo.

El objetivo de la búsqueda heurística es, por tanto, reducir el número de nodos examinados y la efectividad depende del heurístico seleccionado.

El problema es seleccionar la heurística.

- Lo que se puede decir de un heurístico útil es que ofrece soluciones que son suficientemente buenas la mayoría de las veces.

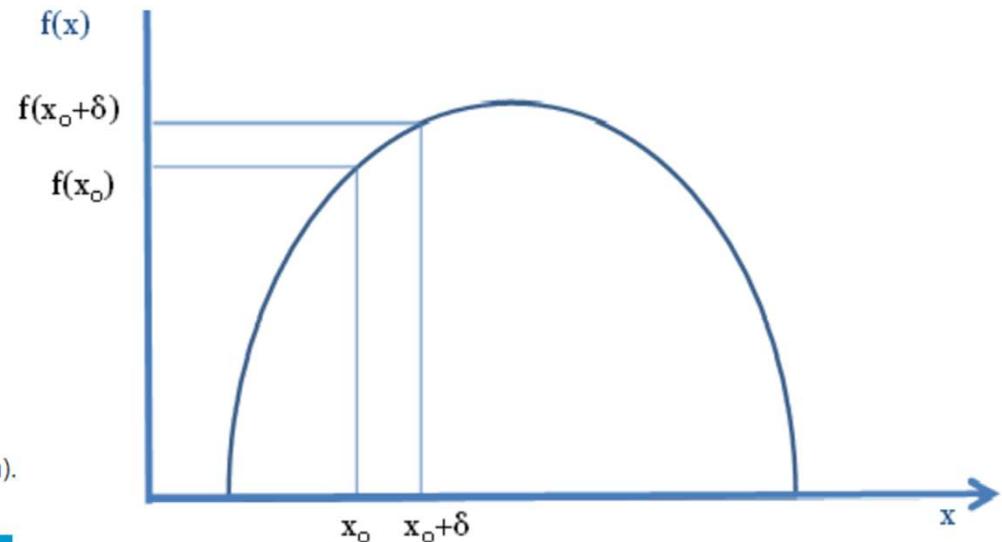
Escalada simple o *hill climbing*

- El proceso, igual como la escalada a una montaña: el objetivo es alcanzar la cima y el estado de búsqueda es el punto donde se encuentra el escalador. La regla de la que se dispone es seguir escalando hasta que se llegue a un punto a partir del cual no se pueda ascender más.
- En el momento que se encuentra un estado más favorable que el actual, se avanza al nuevo estado encontrado

- En una función real de variable real: f . El escalador hipotético, si se encuentra en un punto x_0 , si los saltos que puede dar son de amplitud δ , podría dirigirse a $x_0 + \delta$ o quedarse donde está.
- Dependiendo de la altura $f(x_0 + \delta)$ que se obtenga al avanzar y la altura actual del escalador $f(x_0)$ se decidirá avanzar a la nueva posición si la situación tras el avance es mejor que la situación sin avanzar, porque la altura se incrementa. Si la altura no se incrementa al avanzar, se da como solución la posición x_0



Figura 13. Máximo local (función de la izquierda) y meseta (función de la derecha).



Algoritmo A*

- Intenta combinar las ventajas de la búsqueda en amplitud y en profundidad.
- Usa una función de evaluación $f(n)$ se define como sigue:

Para cada nodo n ,

- $f(n) = h(n) + g(n)$

siendo:

- $h(n)$: heurístico, coste estimado del camino más corto del nodo actual al nodo objetivo ($h(n)=0$ si n es un nodo objetivo).
- $g(n)$: coste real del camino conocido más corto desde el nodo origen al nodo n a evaluar.

- Por lo tanto, la función de evaluación $f(n)$ es el coste del camino más corto desde el nodo inicial al nodo objetivo condicionado a pasar por el nodo n .
- Seleccionará como nodo a expandir el que tenga un menor valor de $f(n)$.

Ejemplo A*

- El problema consiste en que el Robot R debe alcanzar el objetivo M1
- El estado inicial/final del problema lo vamos a representar en una matriz 4x4 de la siguiente manera:

Estado inicial

	0	1	2	3
0	M1	#		
1		#		
2			R	
3				

Estado final

	0	1	2	3
0	R-M1	#		
1	(1,0)	#		
2	(2,0)	(2,1)	(2,2)	
3				

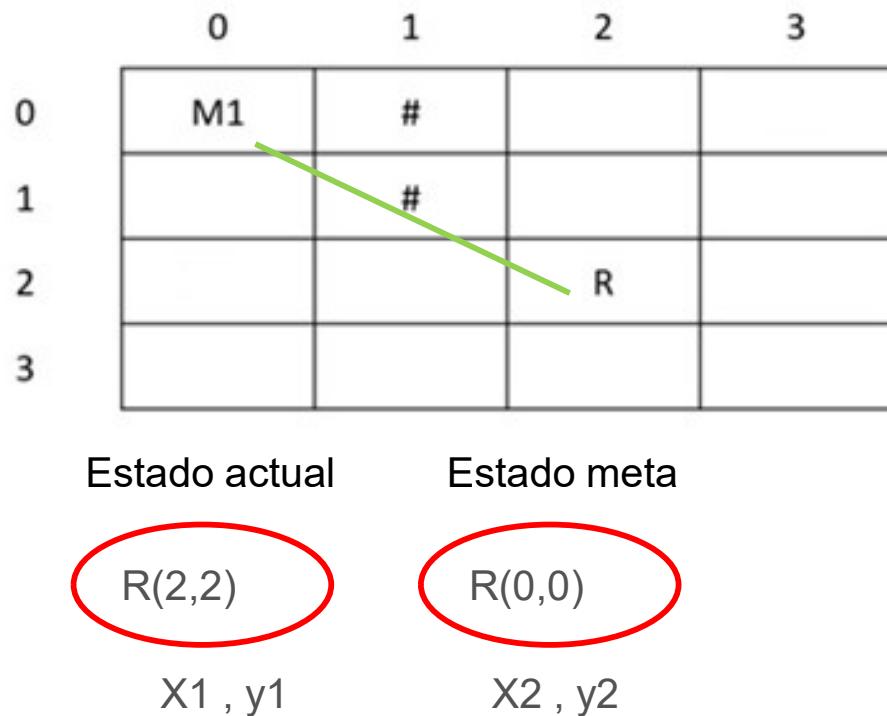
- R: representa el robot. Inicialmente está ubicado en la posición [2,2]
- #: representa una pared.

- Cómo función heurística, la Distancia en Manhattan.
- El coste real (g) de cada acción del robot es 1.
- El robot solo se puede mover horizontal y vertical

Inicial → R(2,2)
Final → R(0,0)

	0	1	2	3
0	M1	#		
1		#		
2			R	
3				

moverR(x,y)



$h = \text{distancia de Manhattan}$

$$= |x_2 - x_1| + |y_2 - y_1|$$

$g = 1 = \text{costo de moverme una casilla a la vez}$

Iteración	Lista cerrada	Lista abierta
1	R(2,2)	

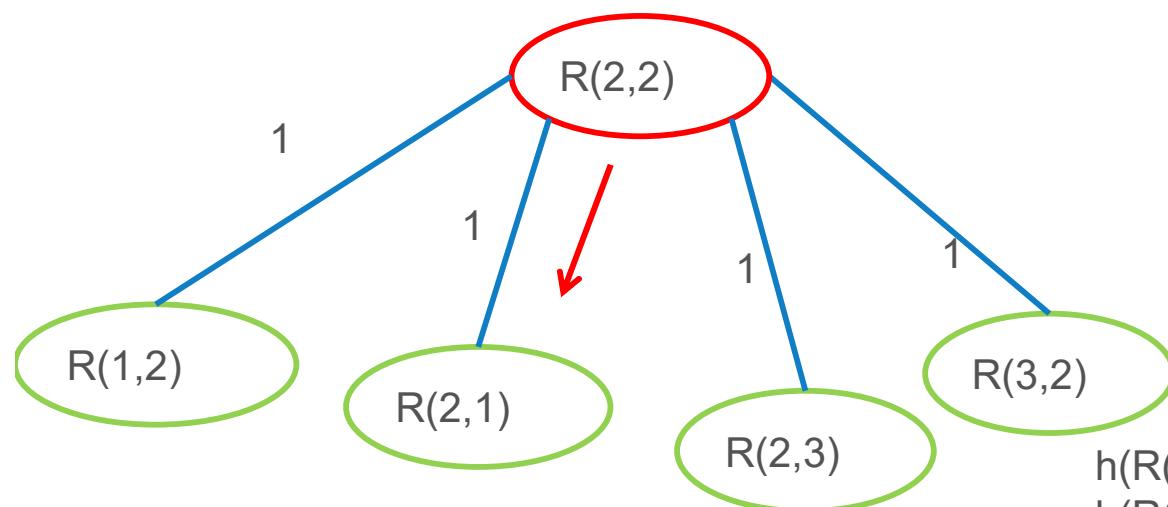
$$f = g + h$$

$$f(R(2,2)) = g(R(2,2)) + H(R(2,2)) = 0 + 4 = 4$$

$$h(R(2,2) \rightarrow (0,0)) = |0-2| + |0-2| = 4$$

moverR(x,y)

	0	1	2	3
0	M1	#		
1		#		
2			R	
3				



$$h = \text{distancia de Manhattan} = |x_2 - x_1| + |y_2 - y_1|$$

$$g = 1$$

Iteración	Lista cerrada	Lista abierta
1	R(2,2)	R(1,2), R(2,1), R(2,3), R(3,2)

$$f = g + h$$

$$f(R(2,2)) = g(R(2,2)) + H(R(2,2)) = 0 + 4 = 4$$

$$f(R(1,2)) = g(R(1,2)) + H(R(1,2)) = 1 + 3 = 4$$

$$f(R(2,1)) = g(R(2,1)) + H(R(2,1)) = 1 + 3 = 4$$

$$f(R(2,3)) = g(R(2,3)) + H(R(2,3)) = 1 + 5 = 6$$

$$f(R(3,2)) = g(R(3,2)) + H(R(3,2)) = 1 + 5 = 6$$

$$h(R(2,2) \rightarrow (0,0)) = |0-2| + |0-2| = 4$$

$$h(R(1,2) \rightarrow (0,0)) = |0-1| + |0-2| = 3$$

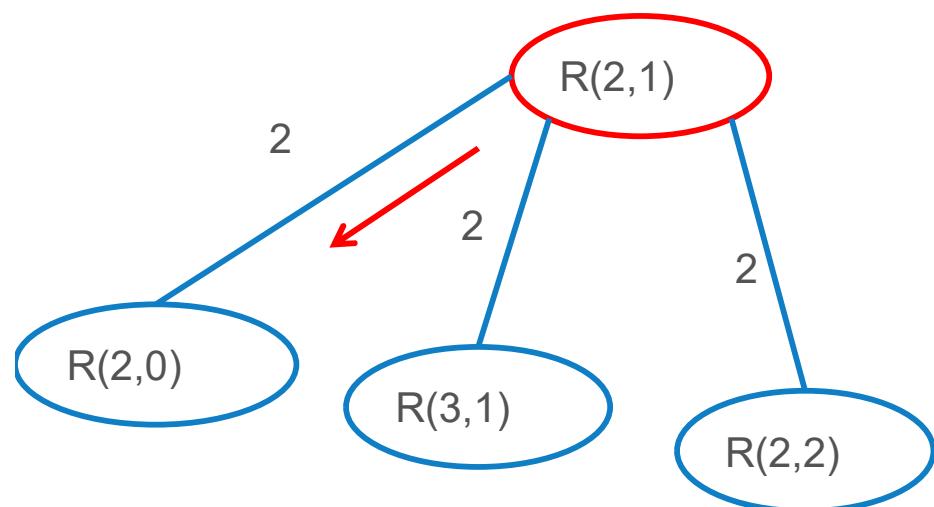
$$h(R(2,1) \rightarrow (0,0)) = |0-2| + |0-1| = 3$$

$$h(R(2,3) \rightarrow (0,0)) = |0-2| + |0-3| = 5$$

$$h(R(3,2) \rightarrow (0,0)) = |0-3| + |0-2| = 5$$

moverR(x,y)

	0	1	2	3
0	M1	#		
1		#		
2		R		
3				



$$h = \text{distancia de Manhattan} \\ = |x_2 - x_1| + |y_2 - y_1|$$

$$g = 1$$

Iteración	Lista cerrada	Lista abierta
1	R(2,2)	R(1,2), R(2,1),(R2,3), R(3,2)
2	R(2,1)	R(2,0),R(3,1), R(2,2)

$$f = g + h$$

$$f(R(2,1)) = g(R(2,1)) + h(R(2,1)) = 1 + 3 = 4$$

$$f(R(2,0)) = g(R(2,0)) + h(R(2,0)) = 2 + 2 = 4$$

$$f(R(3,1)) = g(R(3,1)) + h(R(3,1)) = 2 + 4 = 6$$

$$f(R(2,2)) = g(R(2,2)) + h(R(2,2)) = 2 + 4 = 6$$

$$h(R(2,1) \rightarrow (0,0)) = |0-2| + |0-1| = 3$$

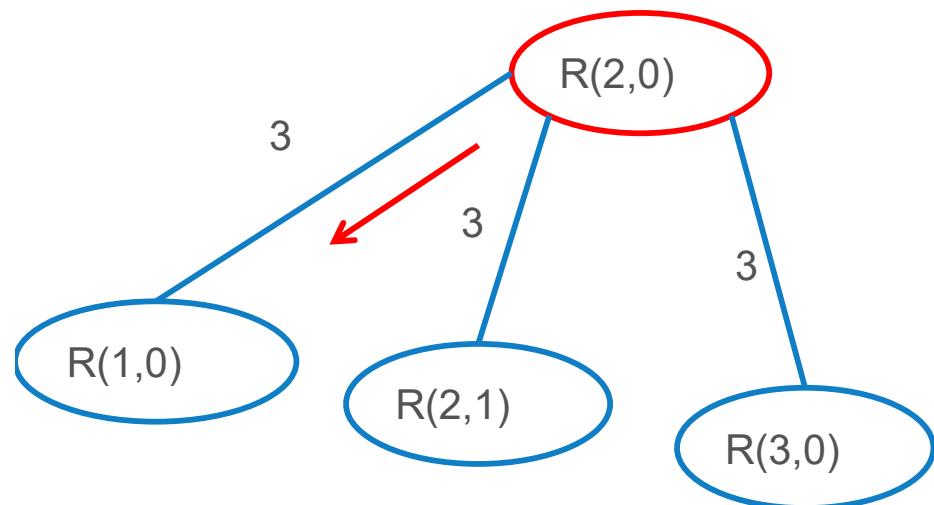
$$h(R(2,0) \rightarrow (0,0)) = |0-2| + |0-0| = 2$$

$$h(R(3,1) \rightarrow (0,0)) = |0-3| + |0-1| = 4$$

$$h(R(2,2) \rightarrow (0,0)) = |0-2| + |0-2| = 4$$

moverR(x,y)

	0	1	2	3
0	M1	#		
1		#		
2	R			
3				



$$h = \text{distancia de Manhattan} = |x_2 - x_1| + |y_2 - y_1|$$

$$g = 1$$

Iteración	Lista cerrada	Lista abierta
1	R(2,2)	R(1,2), R(2,1),(R2,3), R(3,2)
2	R(2,1)	R(2,0),R(3,1), R(2,2)
3	R(2,0)	R(1,0), R(2,1), R(3,0)

$$f = g + h$$

$$f(R(2,0)) = g(R(2,0)) + h(R(2,0)) = 2 + 2 = 4$$

$$f(R(1,0)) = g(R(1,0)) + h(R(1,0)) = 3 + 1 = 4$$

$$f(R(2,1)) = g(R(2,1)) + h(R(2,1)) = 3 + 3 = 6$$

$$f(R(3,0)) = g(R(3,0)) + h(R(3,0)) = 3 + 3 = 6$$

$$h(R(2,0) \rightarrow (0,0)) = |0-2| + |0-0| = 2$$

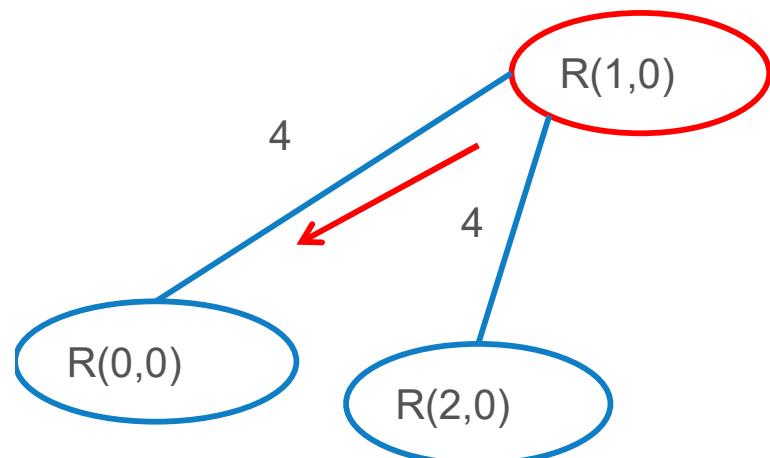
$$h(R(1,0) \rightarrow (0,0)) = |0-1| + |0-0| = 1$$

$$h(R(2,1) \rightarrow (0,0)) = |0-2| + |0-1| = 3$$

$$h(R(3,0) \rightarrow (0,0)) = |0-3| + |0-0| = 3$$

moverR(x,y)

	0	1	2	3
0	M1	#		
1	R	#		
2				
3				



$h = \text{distancia de Manhattan}$

$$= |x_2 - x_1| + |y_2 - y_1|$$

$$g = 1$$

Iteración	Lista cerrada	Lista abierta
1	R(2,2)	R(1,2), R(2,1),(R2,3), R(3,2)
2	R(2,1)	R(2,0),R(3,1), R(2,2)
3	R(2,0)	R(1,0), R(2,1), R(3,0)
4	R(1,0)	R(0,0), R(2,0)

$$f = g + h$$

$$f(R(1,0)) = g(R(1,0)) + h(R(1,0)) = 3 + 1 = 4$$

$$f(R(0,0)) = g(R(0,0)) + h(R(0,0)) = 4 + 0 = 4$$

$$f(R(2,0)) = g(R(2,0)) + h(R(2,0)) = 4 + 2 = 6$$

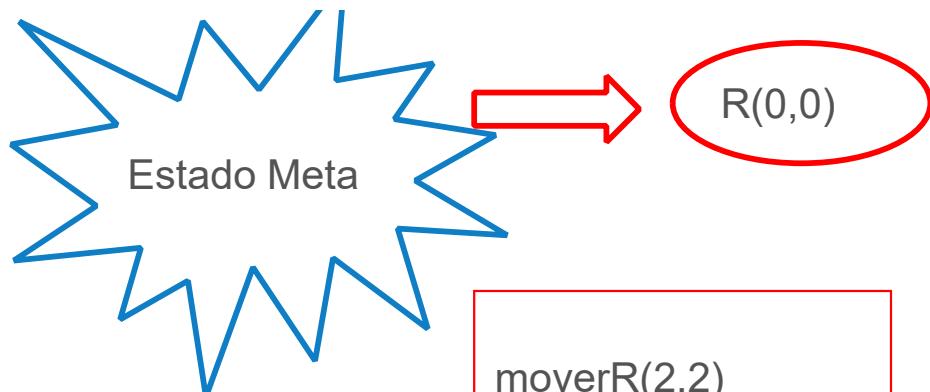
$$h(R(1,0) \rightarrow (0,0)) = |1-0| + |0-0| = 1$$

$$h(R(0,0) \rightarrow (0,0)) = |0-0| + |0-0| = 0$$

$$h(R(2,0) \rightarrow (0,0)) = |2-0| + |0-0| = 2$$

moverR(x,y)

	0	1	2	3
0	RM1	#		
1		#		
2				
3				



moverR(2,2)
moverR(2,1)
moverR(2,0)
moverR(1,0)
moverR(0,0)
cargarR(M,0,0)

Iteración	Lista cerrada	Lista abierta
1	R(2,2)	R(1,2), R(2,1),(R2,3), R(3,2)
2	R(2,1)	R(2,0),R(3,1), R(2,2)
3	R(2,0)	R(1,0), R(2,1), R(3,0)
4	R(1,0)	R(0,0), R(2,0)
5	R(0,0)	----

$$f = g + h$$
$$f(R(0,0)) = g(R(0,0)) + h(R(0,0)) = 4 + 0 = 4$$
$$h(R(0,0) \rightarrow (0,0)) = |0 - 0| + |0 - 0| = 0$$

Gracias por su atención



www.unir.net