



UNIVERSITAT DE  
BARCELONA

**Departament de Matemàtiques i Informàtica**

Gran Via de les Corts Catalanes, 585

08007 Barcelona

Tel . 93 402 16 50, Fax . 93 403 16 01

*Demanda solucions*

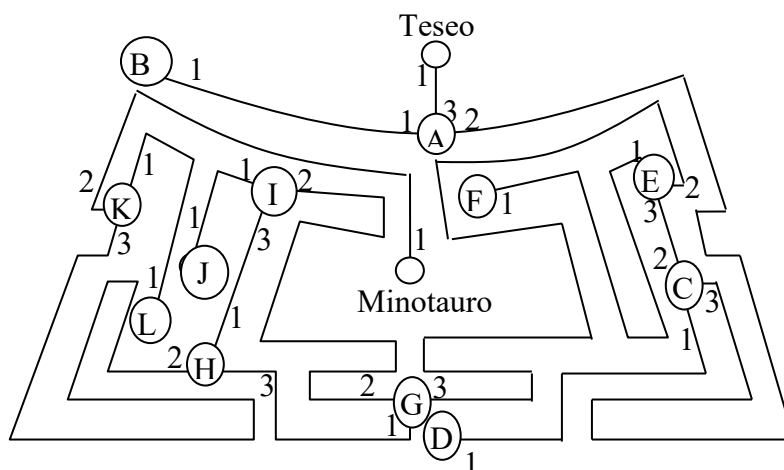
**Problemas de Inteligencia Artificial:  
Búsqueda no informada y búsqueda informada**

## Problema 1: El laberinto del Minotauro



Asterio, el llamado Minotauro, tenía cara de toro y el resto de hombre. El rey Minos lo encerró en el laberinto y le puso vigilancia. El laberinto, construido por Dédalo, era una prisión que a base de intrincados corredores burlaba la salida. (...) Teseo, hijo de Egeo rey de Atenas, fue escogido como tributo para el Minotauro. Cuando llegó a Creta, Ariadna, la hija de Minos, se enamoró de él y se ofreció a ayudarlo si prometía llevarla a Atenas y hacerla su mujer. Y habiéndolo prometido Teseo bajo juramento, rogó ella a Dédalo que le revelara la salida del laberinto.

Entonces por consejo de aquél, le dio a Teseo, cuando entraba en el laberinto un hilo: Teseo lo ató a la puerta y arrastrándolo tras de sí iba entrando. Cuando encontró al Minotauro en la parte extrema del laberinto, lo mató golpeándolo con sus puños; y recogiendo el hilo, salió. (Apolodoro, Biblioteca, III, 11 y Epítome, I, 8-9).



Actualizando este fragmento de mitología griega es necesario programar un Teseo que sea capaz de calcular el camino más directo desde su posición actual hasta la del Minotauro considerando el laberinto anterior. Interpreta las líneas como pasillos y los círculos (etiquetados con letras) como lugares singulares en el trayecto (intersecciones o finales de trayecto). En estos lugares singulares, Teseo decidirá el pasillo a seguir, tomando los pasillos en orden creciente de la numeración del pasillo que aparece en cada entrada de pasillo que da al lugar singular correspondiente. Antes de hacer la programación se solicita:

1. La formalización del problema de búsqueda, definiendo con claridad (de manera formal) el conjunto de estados.
2. Dibujad el árbol de búsqueda que se crea al aplicar el algoritmo de búsqueda en grafos primero en anchura (BFS o amplitud prioritaria). Anotad cada nodo con el estado que representa y cada rama con la acción que lleva de un estado a otro. Diferenciad con un contorno o color diferente los nodos que representan tanto los estados ya alcanzados como los estados objetivo. Indicad la secuencia de estados recorridos ¿Qué solución retornará?
3. Repetid el apartado anterior, pero aplicando esta vez el algoritmo de búsqueda en grafos primero en profundidad (DFS o profundidad prioritaria).

## Problema 2: Muñecas rusas



Tal y como muestra la figura de la izquierda, tenemos tres muñecas rusas geeks, donde la más pequeña es de un Megabyte que no se puede abrir, la mediana es un Gigabyte y la mayor representa un Terabyte. Asumimos que tenemos un robot capaz de meter una muñeca X dentro de otra muñeca Y siempre que:  $X < Y$ ; X está cerrada y sobre la mesa, e Y está abierta, vacía y sobre la mesa. Se trata pues de que este robot determine la manera de guardar todas las muñecas dentro de la mayor, tal y como se muestra en la parte derecha de la figura. Este robot siempre explora las posibles acciones a hacer con el mismo orden: primero intenta las acciones de abrir muñeca, luego la de meter una muñeca dentro de otra y finalmente la de cerrar muñeca.

Estado inicial



Objetivo

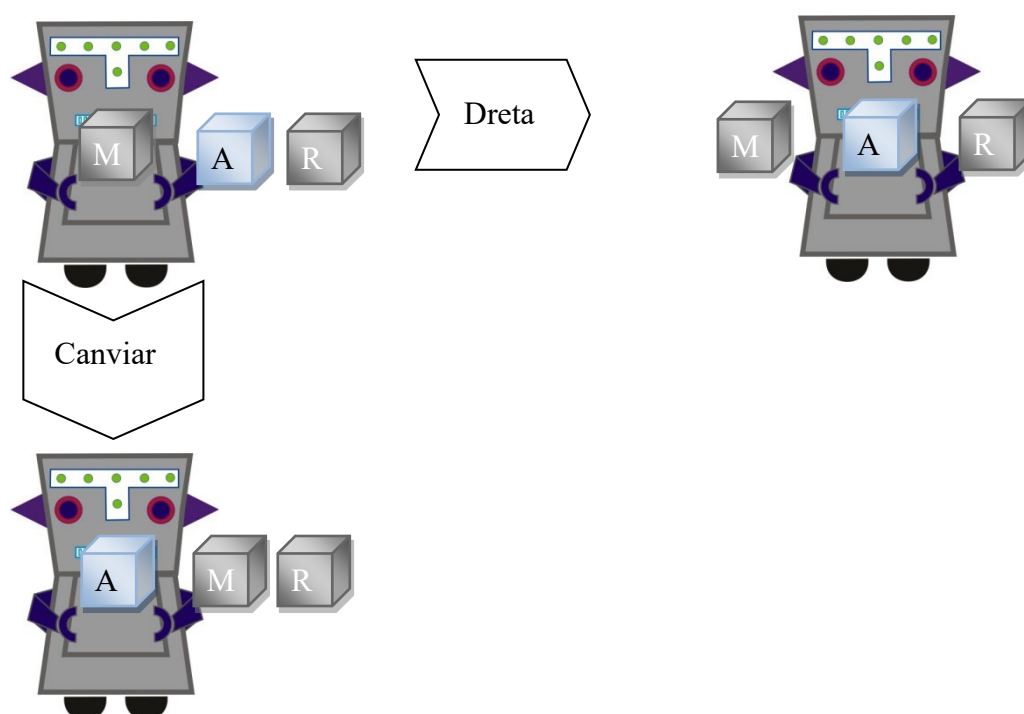


Preguntas:

1. La formalización del problema de búsqueda, definiendo con claridad (de manera formal) el conjunto de estados.
2. Dibujad el árbol que se va abriendo durante la búsqueda en grafos primero en amplitud (o amplitud prioritaria). Representad cada nodo dentro de un cuadrado. Etiquetad las aristas del árbol con la acción que lleva de un estado a otro. Mantened el orden de aristas indicado en el enunciado. Indicad las repeticiones de nodo mediante un cuadrado con contorno de doble línea. Utilizad doble subrayado para resaltar el nombre del estado final.
3. Indicad la solución que retornará.

### Problema 3: Robot ordenant paraules

Ens caldria programar un robot per a que resolgui el següent problema de cerca. Tenim tres cubs, un amb una M, un altre amb una A i un altre amb una R. Inicialment els cubs es troben alineats, de manera que componen la paraula MAR i el robot es situa davant del cub amb la lletra M. El robot es pot moure cap a la dreta i l'esquerra (sempre que se situï davant d'algun cub, no pot sortir). A més, el robot pot (sempre que no es trobi a la posició de més a la dreta) canviar el cub que té davant pel que té a la dreta, quedant-se el robot en la mateixa posició que estava. Per exemple:



El robot haurà acabat el seu treball quan es trobi davant del cub de més a l'esquerra i en els cubs es pugui llegir la paraula MRA. Es considera que l'ordre d'exploració d'accions segueix la descripció de l'enunciat.

1. Formalitzeu el problema com un problema de cerca: Definiu amb claretat (formalment, matemàticament) el conjunt d'estats i les accions.
2. Què retornarà la funció successor pel primer estat? I la funció successor de cadascun dels estats successors del primer estat?
3. Especifica com ha quedat la frontera quan l'algorisme DFS retorna la solució
4. En quina iteració l'algorisme DFS retorna la solució?
5. Especifica la solució que retorna l'algorisme BFS. En quina iteració l'algorisme BFS retorna la solució?
6. Especifica com ha quedat assolits ("alcanzados"/ "reached") quan l'algorisme BFS retorna la solució.

### Problema 4: Mini escacs

Imagina un mini tauler d'escacs com el de la imatge de la Figura 2 al que tenim un alfil blanc a la posició (b,1) i que es pot moure seguint les diagonals que es creuen a la seva posició (és a dir, en direcció NE, NO, SO i SE) tal i com es mostra a la Figura 1. Imagina ara que tenim la possibilitat de fer alguns moviments sense que el cavall situat a la posició (b,3) es mogui i ens proposem trobar la millor manera de matar a aquest cavall (tal i com mostra la Figura 3). A l'hora de pensar possibles moviments de l'alfil, primer pensarem en els que son a posicions més properes i comencen pel Nord-Est (NE) i seguirem en l'ordre donat (NO, SO i SE).

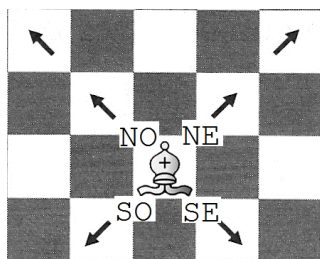


Figura 1: Possibles moviments de l'alfil en les direccions NE (Nord-Est), NO (Nord-Oest), SO (Sud-Oest), SE (Sud-Est).

Figura 2: Inicialment:

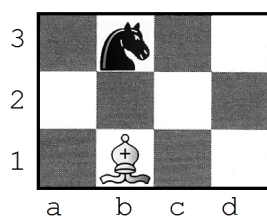
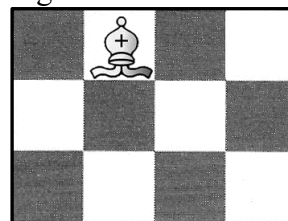


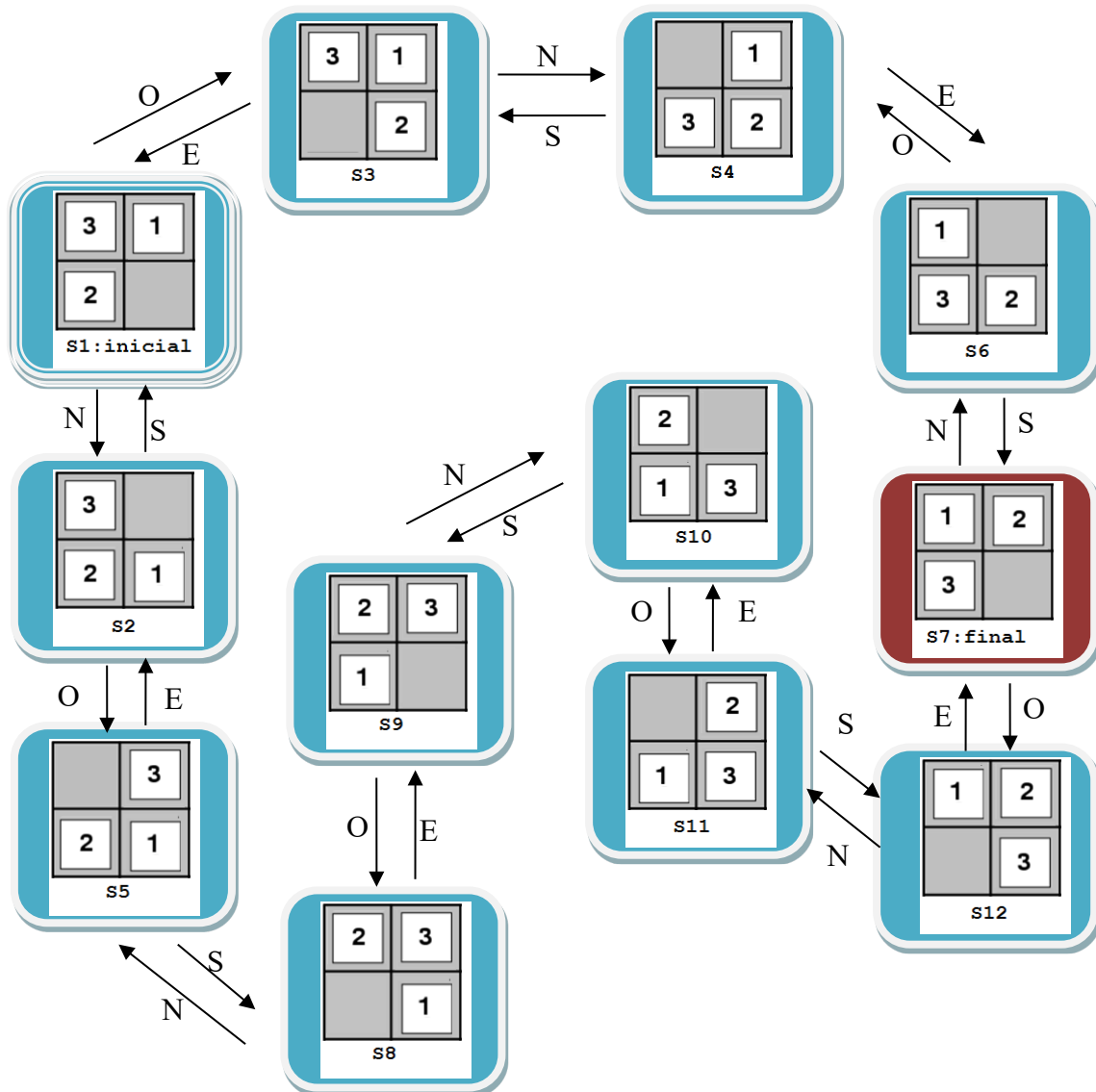
Figura 3: Mata el cavall:



1. Què retornarà la funció successor pel primer estat?
2. Especifica com ha quedat la frontera quan l'algorisme DFS retorna la solució
3. En quina iteració l'algorisme DFS retorna la solució?

## Problema 5: Espai de cerca donat

Considera l'espai de cerca que es mostra a continuació a on l'estat inicial és S1 i l'estat objectiu és el S7.



1. Què retornarà la funció successor pel primer estat? I la funció successor de cadascun dels estats successors del primer estat?
2. Especifica com ha quedat la frontera quan l'algorisme DFS retorna la solució:
3. En quina iteració l'algorisme DFS retorna la solució?
4. Especifica la solució que retorna l'algorisme BFS. En quina iteració l'algorisme BFS retorna la solució?
5. Especifica com ha quedat la frontera quan l'algorisme BFS retorna la solució.

## Problema 6: El 3-puzzle

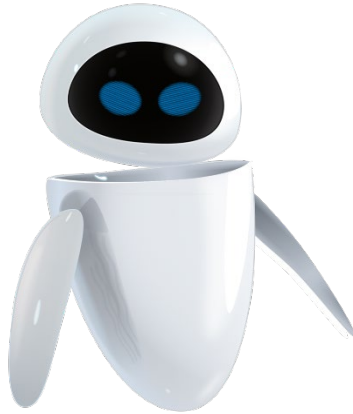


Mientras empaquetaba basura, nuestro amigo Wall-e ha encontrado este puzle y quiere parar un momento a resolverlo pero no tiene cargado el algoritmo de búsqueda informada  $A^*$  y va a necesitar vuestra ayuda. En concreto se solicita:

1. La formalización del problema de búsqueda, definiendo con claridad (de manera formal) el conjunto de estados (incluyendo el inicial y el final), las posibles acciones y la heurística a utilizar.
2. Indicad los estados que explorará el algoritmo  $A^*$ , representando en forma de árbol los estados explorados e indicando el orden y el criterio de recorrido. ¿Qué solución retornará?

Nota: Dibujad cada nodo como un cuadrado que contenga la información relevante. Etiquetad las aristas del árbol con la acción que lleva de un estado a otro. Aplicad las acciones a cada estado siguiendo siempre el mismo orden. Denotad que una acción no aplica en un estado tachando el nombre de la acción. Indicad las repeticiones de nodo mediante un cuadrado con contorno de doble línea. Rodear con un círculo el/los estado/s final/es.

## Problema 7: Un problema teòric

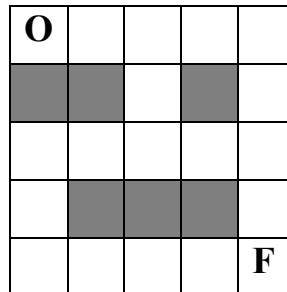


En clase de teoría hemos visto que el algoritmo  $A^*$  es óptimo con heurísticas admisibles (y en grafos con heurísticas consistentes). En este ejercicio exploraremos un poco más a fondo la relación entre estos conceptos. En concreto, debéis hacer lo siguiente:

1. Demostrad que si una heurística es consistente, entonces debe ser también admisible.
2. Dad un ejemplo de una heurística admisible pero no consistente.



## Problema 8: Encontrando el camino en el laberinto



En el laberinto representado por la cuadrícula 2D que muestra la figura, un agente debe encontrar el camino más corto desde su posición de origen (marcada con una O en la figura) hasta la posición final marcada como F. El laberinto está compuesto por celdas que pueden contener obstáculos o paredes. Dichas celdas están representadas sombreadas en gris y el agente no las puede ocupar ni atravesar. El resto de las celdas (es decir, las blancas) representan espacios vacíos por el que el agente puede moverse libremente. El agente sólo puede moverse en las cuatro direcciones principales: arriba, abajo, izquierda y derecha.

Ayuda al agente a aplicar el algoritmo de búsqueda informada A\* para llegar a su destino. En concreto se solicita:

1. La formalización del problema de búsqueda, definiendo con claridad (de manera formal) el conjunto de estados (incluyendo el inicial y el final), las posibles acciones y la heurística a utilizar.
2. Indicad los estados que explorará el algoritmo A\*, representando en forma de árbol los estados explorados e indicando el orden y el criterio de recorrido. ¿Qué solución retornará?

Nota: Dibujad cada nodo como un cuadrado que contenga la información relevante. Etiquetad las aristas del árbol con la acción que lleva de un estado a otro. Aplicad las acciones a cada estado siguiendo siempre el mismo orden. Denotad que una acción no aplica en un estado tachando el nombre de la acción. Indicad las repeticiones de nodo mediante un cuadrado con contorno de doble línea. Rodear con un círculo el/los estado/s final/es.