



- A. Una búsqueda por profundización iterativa encuentra la solución en 5 iteraciones
  - B. Una búsqueda en anchura encuentra la solución de coste óptimo
  - C. Una búsqueda de coste uniforme encuentra la solución más corta
  - D. Una búsqueda en profundidad con máximo nivel de profundidad  $m=4$  encuentra la solución más corta
- 

4) Sea una búsqueda de tipo A ( $f(n)=g(n)+h(n)$ ) donde la función  $h(n)$  es admisible y consistente. El algoritmo devuelve una solución desde el nodo inicial **A** al nodo objetivo **G** que atraviesa un nodo **n1**. Indica cuál de las siguientes afirmaciones es **CORRECTA**:

- A.  $f(n1) \leq f(G)$
  - B.  $f(G) < h^*(A)$
  - C.  $h^*(A) < h(n1)$
  - D. Ninguna de las opciones anteriores es correcta
- 

5) Sean tres funciones  $f1(n)=g(n)+h1(n)$ ,  $f2(n)=g(n)+h2(n)$  y  $f3(n)=g(n)+h3(n)$  tal que se sabe que  $h2(n)$  es admisible,  $h3(n)$  no lo es y  $\forall n \ h1(n) \leq h2(n)$ . Asumiendo que  $G1$  es el nodo solución que devuelve la búsqueda con  $f1$ ,  $G2$  el que devuelve la búsqueda con  $f2$  y  $G3$  el que devuelve la búsqueda con  $f3$ , indica la respuesta **CORRECTA**:

- A. Se cumple  $f1(G1) < f2(G2)$
  - B. Se cumple  $f2(G2) \leq f3(G3)$
  - C. Se cumple  $g(G1) < f2(G2)$
  - D. Ninguna de las respuestas anteriores es correcta
- 

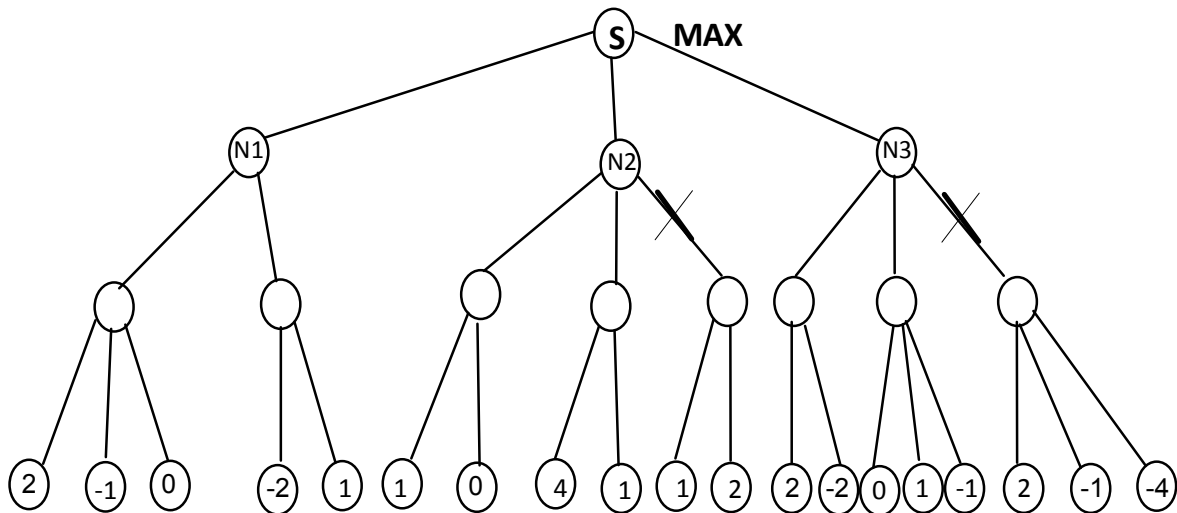
6) Supongamos la aplicación de una función heurística  $h(n)$  al grafo de la pregunta 3. Se sabe que  $h(A)=8$  y que para todo nodo sucesor  $n'$  de un nodo  $n$ ,  $h(n')=h(n)-2$ . En el caso de que  $n'$  tenga  $i$  nodos padre entonces  $h(n')=\min(h(n_1), \dots, h(n_i))-2$ . Indica la respuesta **CORRECTA**:

- A. La heurística es admisible
  - B. La heurística es consistente
  - C. La aplicación de una estrategia de tipo A ( $f(n)=g(n)+h(n)$ ) no devuelve la solución óptima porque la heurística no es admisible.
  - D. Ninguna de las respuestas anteriores es correcta
- 

7) Sea un grid de  $4 \times 4$  donde la casilla inferior izquierda es  $(x,y) = (1,1)$  y la casilla superior derecha es la  $(x,y) = (4,4)$ . Hay un robot en la casilla  $(1,3)$ , una lata en la casilla  $(1,2)$  y un contenedor triturador en la casilla  $(4,1)$ . El robot puede moverse a una casilla adyacente en cuatro direcciones (arriba, abajo, derecha e izquierda) y puede empujar la lata en cualquiera de las cuatro direcciones, respetando los límites del grid. Para empujar una lata, el robot debe situarse en una casilla adyacente a la lata. Como resultado de una acción de empujar, la posición del robot y la posición de la lata se desplazan una casilla en la dirección del empuje, respetando los límites del grid. El objetivo del problema es empujar la lata al contenedor triturador. Indica la respuesta **INCORRECTA** aplicada a este estado del problema:

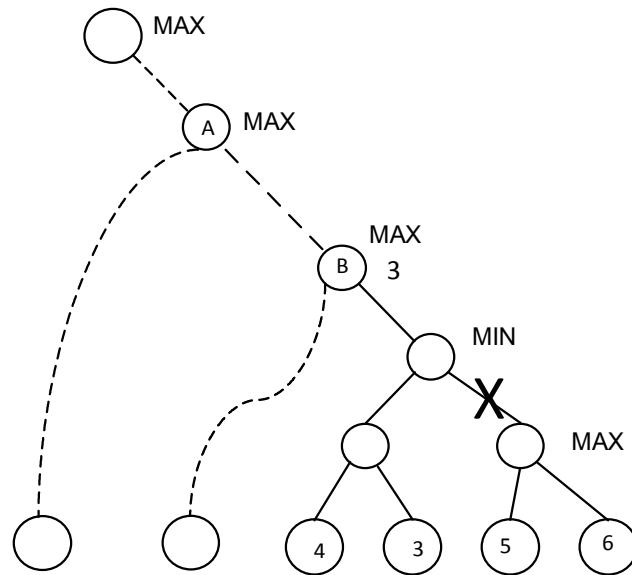
- A. Hay dos acciones de movimiento del robot aplicables en el estado
- B. Hay una acción de empujar lata aplicable en el estado
- C. La solución óptima se encuentra en el nivel 5
- D. Una búsqueda en profundidad (limitada a máximo nivel  $m=5$ ) expandiría el mismo número de nodos que los nodos expandidos en los 5 primeros niveles de una búsqueda en anchura

8) Dado el árbol de juego de la figura donde se ha aplicado un procedimiento alfa-beta, indica la respuesta correcta:



- A. Se produce un corte en el nodo N2 que podaría también la rama intermedia de N2
- B. No se produce corte en N2 y por tanto no se poda la rama de la derecha de N2
- C. Se produce un corte en el nodo N3 que podaría también la rama intermedia de N3
- D. No se produce corte en N3 y por tanto no se poda la rama de la derecha de N3

9) Dado el desarrollo parcial de una búsqueda alfa-beta indicado en la figura, ¿qué valor provisional debería tener el nodo A para que se produzca el corte de la figura?



- A. Todos los valores comprendidos en el intervalo  $[-\alpha, 3]$  provocarían el corte
- B. Todos los valores comprendidos en el intervalo  $[3, +\infty]$  provocarían el corte
- C. Todos los valores comprendidos en el intervalo  $[4, +\infty]$  provocarían el corte
- D. El corte de la figura no se puede producir nunca

## Sistemas Inteligentes – Examen Bloque 1, 5 noviembre 2021

### Problema: 2 puntos

Se dispone de un conjunto de cajas esparcidas por el suelo cada una de las cuales lleva asociada una etiqueta que es un número entero. Puede haber más de una caja con el mismo número. Por ejemplo: 17, 5, 6, 22, 5, 4, 7, 12, 6, 1, 21, ....

Se desea apilar las cajas en torres en orden decreciente del número de etiqueta de modo que una caja con número M solo se puede apilar encima de una caja con número N tal que  $M \leq N$ . Las condiciones sobre las torres son:

- 1) Puede haber un número indeterminado de torres
- 2) Hay dos tipos de torres: las que apilan cajas de número par y las que apilan cajas de número impar
- 3) Puede haber un número indeterminado de cajas en cada torre

El patrón para representar un estado de este problema es:

`(colocar [torre ctm]m suelo csm)`

donde  $ct \in \text{INTEGER}$  y  $cs \in \text{INTEGER}$ .

**NOTA:** el predicado `(evenp <arg>)` devuelve TRUE si `<arg>` es un número par.

Usando CLIPS y búsqueda en grafos (GRAPH-SEARCH), se pide:

- 1) Escribir la **base de hechos inicial** sabiendo que inicialmente hay dos torres, una contiene las cajas 13 y 17, y otra torre contiene la caja 22. Además, en el suelo tenemos las siguientes cajas: 17, 5, 6, 22, 5, 4, 7, 12, 6, 1, 21.
- 2) Asumiendo que ya existen varias torres creadas en el problema, escribe una única regla que permita coger una caja cualquiera del suelo y colocarla en alguna de las torres existentes.
- 3) Asumiendo que ya hay varias torres creadas en el problema, queremos diseñar una operación de desapilar una caja con etiqueta X de una torre T (desapilar=quitar la caja que está en la cima de la torre) con la condición de que haya una o más torres distintas de T donde la caja que está en la cima también tenga la misma etiqueta X. Escribe una única regla que permita realizar esta operación.
- 4) Asumiendo que ya hay varias torres creadas en el problema, escribe una única regla que devuelva un mensaje SI HAY AL MENOS DOS cajas en el suelo con el mismo número de etiqueta X tal que ninguna de las torres contiene cajas con la misma etiqueta X. La regla debe mostrar el siguiente mensaje: "Hay al menos dos cajas con número X en el suelo y ninguna torre contiene cajas con número X".

```
(deffacts datos
  (colocar torre 13 17 torre 22 suelo 17 5 6 22 5 4 7 12 6 1 21)
)
```

```
(defrule colocar-caja
  (colocar $?x torre ?top $?resto suelo $?y ?caja $?z)
  (test (or (and (evenp ?top)(evenp ?caja))
             (and (not (evenp ?top))(not (evenp ?caja)))))
  (test (<= ?caja ?top))
=>
  (assert (colocar $?x torre ?caja ?top $?resto suelo $?y $?z)))
```

```
(defrule desapilar-mismo-tope
  (colocar $?x1 torre ?top $?x2 suelo $?s)
  (test (or (member (create$ torre ?top) $?x1)
            (member (create$ torre ?top) $?x2)))
=>
  (assert (colocar $?x1 torre $?x2 suelo $?s)))
```

```
(defrule comprobacion
  (colocar $?torres suelo $?x1 ?cj1 $?x2 ?cj1 $?x3)
  (test (not (member ?cj1 $?torres)))
=>
  (printout t "Hay al menos dos cajas con número " ?cj1 " en el suelo y ninguna torre contiene cajas con
dicho número " crlf))
```