

Sistemas Inteligentes – Examen Bloque 1, 31 octubre 2019
Test A (1,75 puntos) puntuación: max (0, (aciertos – errores/3)*1,75/9)

Apellidos:								Nombre:
Grupo:	A	B	C	D	E	F	G	INF+ADE

1) Sea el siguiente patrón para el problema de las torres de Hanoi:

(hanoi [tower tw^s d1^s d2^s d3^s d4^s]^m) ;; tw^s ∈ {T1,T2,T3} d_i^s ∈ [0-4]

donde dos ejemplos de hecho serían (hanoi tower T1 2 4 0 0 tower T2 0 0 0 0 tower T3 1 3 0 0)
(hanoi tower T1 2 4 0 0 tower T2 1 3 0 0 tower T3 0 0 0 0).

¿Movería la siguiente regla un disco de la torre T3 a cualquiera de las otras dos torres T1, T2 (a su izquierda)?

```
(defrule move-disk-from-T3-to-Tx
  (hanoi $?rest1 tower ?Tx ?d2 $?rest2 0 $?rest3 tower T3 ?d1 ?a ?b ?c)
  (test (or (= ?d2 0) (< ?d1 ?d2))))
=>
(assert (hanoi $?rest1 tower ?Tx ?d1 ?d2 $?rest2 $?rest3 tower T3 ?a ?b ?c 0)))
```

- A. Sí, la regla es correcta
- B. Sería correcta si se añade la comprobación (not (member tower \$?rest2)) en la LHS de la regla.
- C. Además de la comprobación del apartado (B) debería también comprobarse (<> ?d1 0) en la LHS de la regla.**
- D. Además de la comprobación de los apartados (A) y (B), debería comprobarse (> (length \$?rest2) 0) en la LHS de la regla.

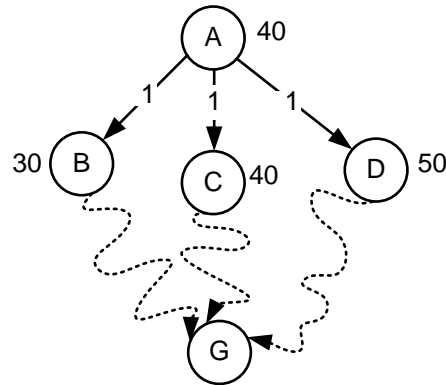
2) Sea un SBR formado por BHinicial={{(lista 3 4 5 6 6 6 8 9)}}, y las siguientes reglas:

<pre>(defrule R1 ?f <- (lista \$?x ?z ?y \$?w) (test (= ?z ?y))) => (assert (lista \$?x ?z ?y \$?w)))</pre>	<pre>(defrule R2 ?f <- (lista \$?x ?z ?y \$?w) (test (> ?z ?y))) => (assert (lista \$?x ?z ?y \$?w)))</pre>
---	--

¿Cuál sería el contenido del Conjunto Conflicto (Agenda) tras el primer pattern-matching?

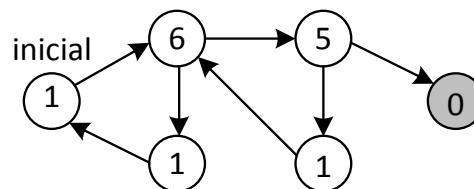
- A. Dos instancias de la regla R1 y dos de la R2
- B. Dos instancias de la regla R1**
- C. Dos instancias de la regla R1 y una instancia de la regla R2
- D. Ninguna instancia

- 3) Supongamos el espacio de búsqueda de la figura en el que se quiere encontrar una senda solución del estado inicial A al único estado meta G. Los arcos tienen un coste unitario y en cada nodo se indica el valor de $h(n)$, que sabemos que es admisible. Aplicando un algoritmo de tipo A ($f(n)=g(n)+h(n)$), con re-expansión, se cumple que:



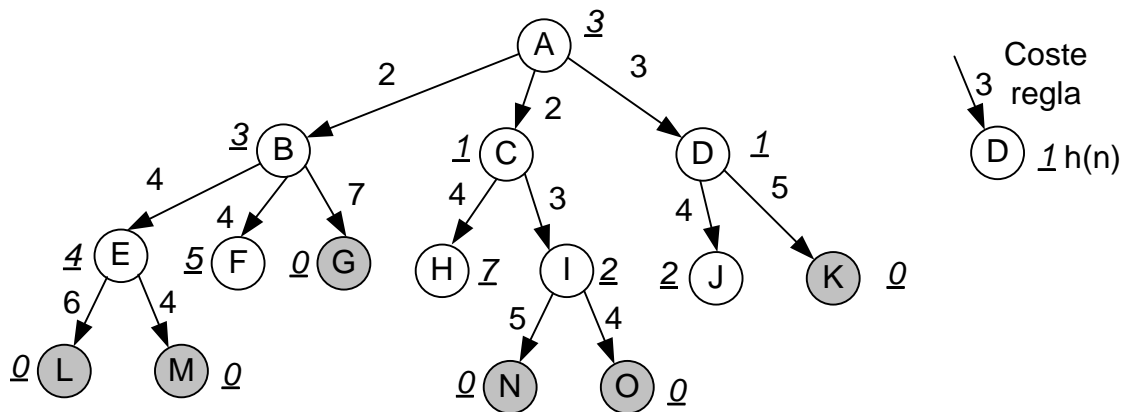
- A. Si obtenemos como solución un camino de A a G de coste 40, será una senda solución óptima.
- B. Si obtenemos como como solución un camino de A a G de coste 51, no será una senda solución óptima.
- C. La aplicación del algoritmo A no obtendrá la solución óptima.
- D. Con los datos de la figura, $h(n)$ es consistente.

- 4) Supongamos que se aplica un algoritmo de tipo A ($f(n)=g(n)+h(n)$) en el siguiente espacio de estados, donde el nodo sombreado es el nodo meta, todos los arcos tienen coste 1, y en los nodos se indica el valor de la función $h(n)$. Indica la respuesta **CORRECTA**:



- A. No se puede aplicar un algoritmo A es el espacio de estados anterior, ya que $h(n)$ no es consistente.
- B. El algoritmo no terminará, pues entrará en un ciclo infinito.
- C. El algoritmo encontrará la solución, pero no tiene que ser necesariamente la óptima.
- D. El algoritmo encontrará la solución óptima.

- 5) Para el espacio de estados de la figura, donde los nodos sombreados son nodos meta, y dada una búsqueda de tipo A ($f(n)=g(n)+h(n)$), ¿cuántos nodos es necesario generar, incluyendo el nodo A, para encontrar la solución?



- A. 10.
- B. 11.
- C. 13.
- D. 15.

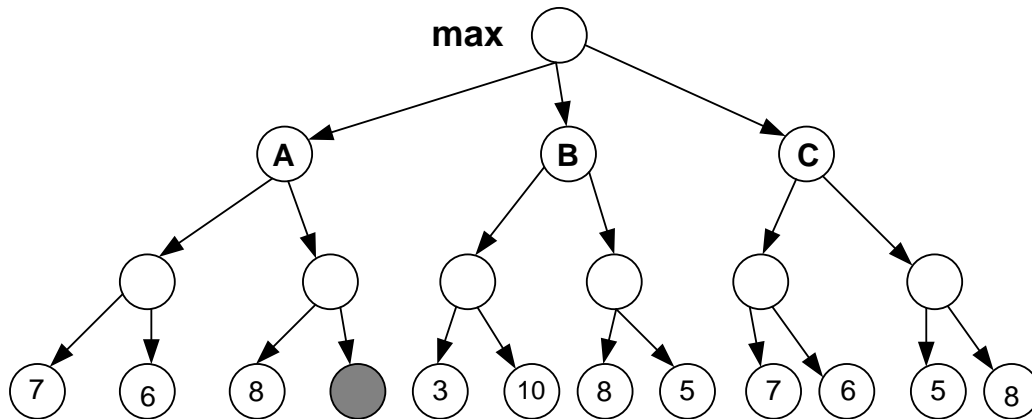
- 6) Sea el espacio de estados de la figura anterior. Asumiendo que dados dos o más nodos con el mismo valor $f(n)$ se escoge un nodo aleatoriamente, indica la respuesta **CORRECTA**:

- A. La aplicación de una estrategia en Profundidad con $m=3$ (máximo nivel de profundidad) encontrará como solución cualquier nodo meta.
- B. La aplicación de una estrategia por Coste Uniforme encontrará como solución los nodos G o N.
- C. La aplicación de una estrategia por Profundización Iterativa encontrará como solución los nodos L, M, N o O.
- D. Ninguna de las repuestas anteriores es correcta.

- 7) Supongamos un tablero de 4x4 donde un robot está situado en la casilla inferior izquierda $(x,y)=(1,1)$ y desea llegar a la casilla superior derecha $(x,y)=(4,4)$. El robot puede realizar movimientos horizontal, vertical o diagonalmente, y todos ellos tienen un coste unitario. Indica la respuesta **INCORRECTA**:

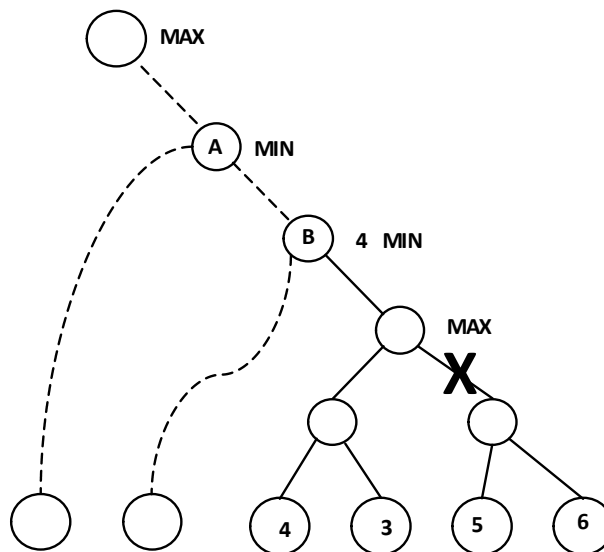
- A. El coste de la solución óptima a este problema es 3.
- B. La distancia de Manhattan del robot a la casilla meta es una heurística admisible.
- C. La aplicación de un algoritmo en anchura devolverá la solución óptima.
- D. Al expandir el nodo $(x,y)=(2,3)$ se generarán 8 nodos hijo.

- 8) Dado el árbol de juego de la figura, donde aplicamos un procedimiento alfa-beta, indica la respuesta CORRECTA:



- A. Si el nodo sombreado toma un valor ≤ 8 , se producirá un corte alfa en el nodo B y C.
- B. Si el nodo sombreado toma un valor ≥ 10 , se producirá un corte alfa en el nodo B y C
- C. Para cualquier valor del nodo sombreado, se producirá siempre un corte alfa en B.
- D. Para cualquier valor del nodo sombreado, se producirá siempre un corte alfa en C.**

- 9) Dado el desarrollo parcial de una búsqueda alfa-beta indicado en la figura, donde el nodo B tiene un valor volcado provisional de 4 ¿Qué valor volcado provisional debe tener el nodo A para que se produzca el corte efectivo indicado en la figura?



- A. Nunca se producirá el corte
- B. Menor o igual que 3**
- C. Mayor o igual que 3
- D. Menor que 3

Sistemas Inteligentes – Examen Bloque 1, 31 octubre 2019

Problema: 2 puntos

Una empresa fabrica tres gamas distintas de un determinado producto: gama media, gama alta y gama suprema. Cada gama del producto se fabrica con un número de piezas de dos tipos distintos:

- Un producto de gama media se fabrica con 3 piezas de clase A y 1 de clase B
- Un producto de gama alta se fabrica con 2 piezas de clase A y 2 de clase B
- Un producto de gama suprema se fabrica con 3 piezas de clase B

La empresa dispone en stock de un número de piezas de clase A y B, y dispone además de un número de pedidos para cada una de las tres gamas de productos. El patrón que representa la información dinámica del problema es:

$(\text{empresa } [\text{pieza tipo}^s \text{ cant}^s]^m [\text{pedido-producto } \text{gama}^s \text{ num}^s]^m)$

$\text{tipo} \in \{A, B\}$

$\text{cant} \in \text{INTEGER}$;; cantidad de piezas del tipo correspondiente

$\text{gama} \in \{\text{media, alta, suprema}\}$

$\text{num} \in \text{INTEGER}$;; número de productos pedidos de la gama correspondiente

Un ejemplo de situación inicial del problema es:

- La empresa dispone de 10 piezas de tipo A y 15 piezas de tipo B
 - Existen pedidos de 2 productos de gama media, 1 producto de gama alta y 1 producto de gama suprema
- 1) (0,4 puntos) Escribe la Base de Hechos correspondiente a la situación inicial que se muestra arriba. Incluye los patrones adicionales que necesites para representar la información estática del problema, así como los hechos asociados a dichos patrones. NOTA: se recomienda mantener el mismo orden de los dos tipos de piezas (A y B) en el patrón dinámico y en los patrones estáticos necesarios.
 - 2) (0,5 puntos) Escribe una regla para servir un producto de cualquier gama. Como consecuencia de servir un producto se deberá decrementar las piezas disponibles en la fábrica y decrementar el número de productos pedidos de la gama servida.
 - 3) (0,5 puntos) Escribe una regla para servir todos los productos solicitados de una misma gama.
 - 4) (0,6 puntos) Escribe una regla para determinar si, para dos productos pedidos de distinta gama, solo existen piezas para servir uno de los dos. La regla debe mostrar un mensaje del siguiente tipo: "Solo hay piezas para servir un producto de la gama XXX o de la gama YYY".

```
(def facts datos
  (empresa pieza A 10 pieza B 15 pedido-producto media 2 pedido-producto
alta 1 pedido-producto suprema 1)
  (gama media A 3 B 1)
  (gama alta A 2 B 2)
  (gama suprema A 0 B 3)
)
```

```

(defrule producir_uno
  (gama ?gama A ?pa B ?pb)
  (empresa pieza A ?na pieza B ?nb $?x pedido-producto ?gama ?numped $?y)
  (test (>= ?numped 1))
  (test (and (>= ?na ?pa)(>= ?nb ?pb)))
  =>
  ;;(printout t "gama " ?gama "piezas A " ?pa " " ?na crlf)
  (assert (empresa pieza A (- ?na ?pa) pieza B (- ?nb ?pb) $?x pedido-
producto ?gama (- ?numped 1) $?y)))

(defrule producir_todos
  (gama ?gama A ?pa B ?pb)
  (empresa pieza A ?na pieza B ?nb $?x pedido-producto ?gama ?numped $?y)
  (test (>= ?numped 1))
  (test (and (>= ?na (* ?pa ?numped))(>= ?nb (* ?pb ?numped))))
  =>
  (assert (empresa pieza A (- ?na (* ?pa ?numped)) pieza B (- ?nb (* ?pb
?numped)) $?x pedido-producto ?gama 0 $?y)))

(defrule servir_solo_uno
  (empresa pieza A ?na pieza B ?nb $?x pedido-producto ?gama1 ?numped1 $?y
pedido-producto ?gama2 ?numped2 $?z)
  (test (and (>= ?numped1 1)(>= ?numped2 1)))
  (gama ?gama1 A ?pa1 B ?pb1)
  (gama ?gama2 A ?pa2 B ?pb2)
  (test (or (< ?na (+ ?pa1 ?pa2)) (< ?nb (+ ?pb1 ?pb2))))
  =>
  (printout t "Solo hay piezas para servir un producto de la gama " ?gama1
" o de la gama " ?gama2 crlf))

```