Sistemas Inteligentes – Examen Bloque 1, 5 noviembre 2021
Test A (1,75 puntos) puntuación: max (0, (aciertos – errores/3)*1,75/9)

Apellidos:	**Nombre**:
Grupo: A B C D E F G 4IA	

1) Dada la base de hechos inicial: BH={(lista 6 3 5 1 4 7 2 6 3) (pares 0)} y la siguiente regla para calcular el número de pares en una lista de números naturales

```
(defrule contar-pares
  ?f1 <- (lista $?a  ?b  $?c)
  ?f2 <- (pares ?p)
  (test (= 0 (mod ?b 2)))
=>
  (assert (lista $?a  $?c))
  (assert (pares (+ 1 ?p))))
```

Si nuestro objetivo es obtener una base de hechos final (tras la ejecución sucesiva de la regla) en la cual el hecho (pares …) solo puede aparecer una vez (conteniendo el número de pares en la lista). ¿Cuál de las siguientes afirmaciones es CIERTA para obtener nuestro objetivo?

A. La regla es correcta.
B. Sería necesario añadir (retract ?f1)
C. Sería necesario añadir (retract ?f1) y (retract ?f2)
D. Sería necesario añadir (retract ?f2)

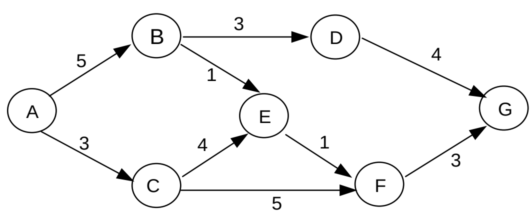
---

2) Se quiere una regla en CLIPS que haga matching con el siguiente hecho: (lista a b a a b b c a b c). ¿Cuál de los siguientes patrones habría que incluir en la parte izquierda de dicha regla?

A. (lista $? $?x $? $?x ?)
B. (lista $?y ?x ?x $?y $?)
C. (?a $?c)
D. (lista $? $?y ?x ?x ?x $? $?y $? ?x)

---

3) Dado el grafo de la figura, donde el nodo G es el nodo Meta, indica la respuesta **CORRECTA** (ante dos nodos con el mismo valor de f(n) se expande antes el nodo alfabéticamente anterior):



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) <= 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) <= 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) <= 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),…,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

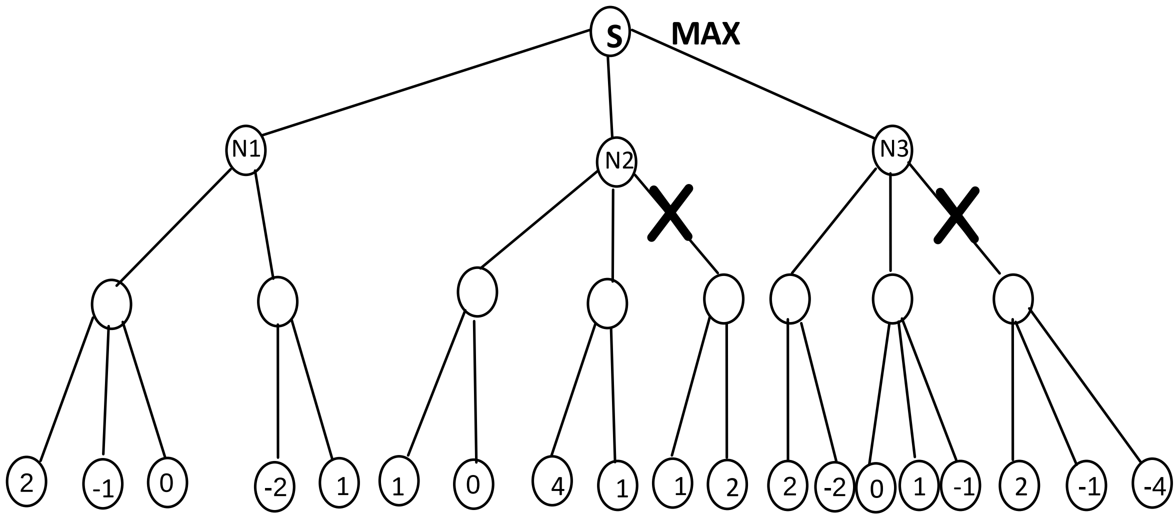
---

7) Sea un grid de 4x4 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

---

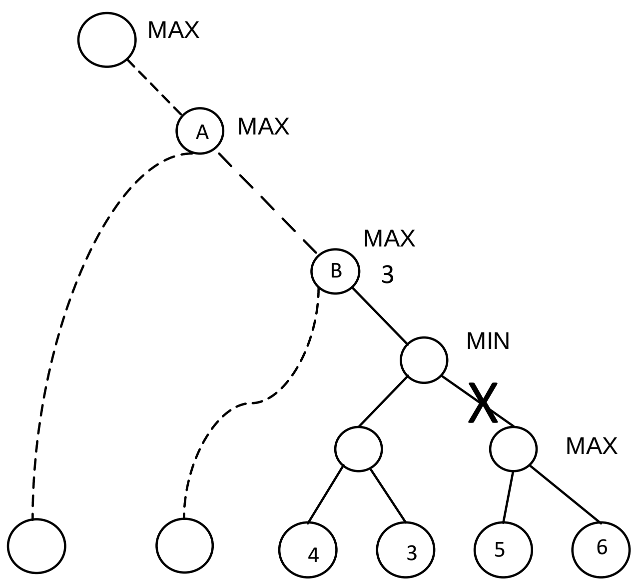
8) Dado el árbol de juego de la figura donde se ha aplicado un procedimiento alfa-beta, indica la respuesta correcta:



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 [-α, 3] provocarían el corte
B. Todos los valores comprendidos en el intervalo [3, +∞] provocarían el corte
C. Todos los valores comprendidos en el intervalo [4, +∞] 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 <= 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 ct^m]^m suelo cs^m)`

donde `ct` ∈ INTEGER y `cs` ∈ 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, y que todas ellas contienen al menos una caja, escribe una única regla que permita coger una caja cualquiera del suelo y colocarla en alguna de las torres ya existentes, respetando la restricción de que caja y torre sean par o impar.

3) Asumiendo que ya hay varias torres creadas en el problema, y que todas ellas contienen al menos una caja, 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, y que todas ellas contienen al menos una caja, 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”.