

Apellido y Nombres: L.U.:
 DNI: Cantidad de hojas entregadas (sin enunciado):

Nota: Resolver los ejercicios en HOJAS SEPARADAS. Poner nombre y número a todas las hojas.

Ejercicios

Lógica Proposicional y Cálculo de Predicados

- (i) Defina cuándo, en Lógica Proposicional, dos fórmulas bien formadas (fbfs) son *equivalentes*.
 (ii) Para cada uno de los siguientes incisos, indique si F_1 y F_2 son equivalentes, justificando adecuadamente su respuesta.

(i) $F_1 = \neg(a \rightarrow (b \rightarrow a))$ $F_2 = \neg(((a \vee b) \wedge \neg b) \rightarrow a)$

(ii) $F_1 = \neg b \vee \neg c \vee a$ $F_2 = (b \wedge c) \rightarrow a$

Defina cuándo un conjunto de fbfs $L = \{F_1, F_2, \dots, F_n\}$ es unificado por una sustitución σ .

Para cada uno de los siguientes incisos halle, de ser posible, una sustitución unificadora σ para las fbfs F_1 y F_2 , o indique por qué no es posible hallarla. OBSERVACIÓN: Los símbolos x , y y z denotan variables.

(i) $F_1 = p(f(a), f(x), c)$ $F_2 = p(y, f(b), z)$

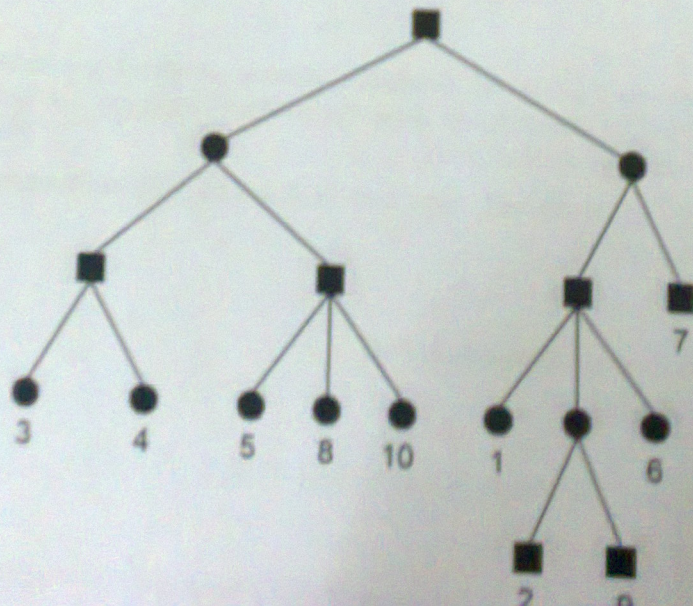
(ii) $F_1 = p(a, b, c)$ $F_2 = p(b, x, y)$

(iii) $F_1 = p(x, f(b), y)$ $F_2 = p(a, x, x)$

Búsqueda Adversaria

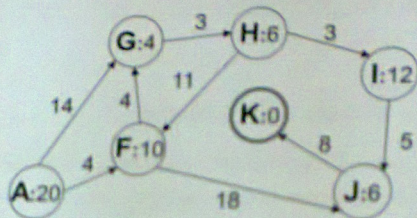
Dado un espacio de búsqueda adversaria, explique en qué situaciones ocurren las podas α y β .
 Aplique el método minimax con poda $\alpha - \beta$ al árbol de la figura visitando los nodos de izquierda a derecha. Indique claramente cuál es el valor final que se asigna a la raíz y las podas que se producen.

- maximiza
 ● minimiza



3. Búsqueda Ciega y Heurística

- Explique qué significa que una estrategia de búsqueda sea ciega.
- Describa la estrategia de búsqueda del método *Lowest-Cost-First Search* (LCFS).
- Explique qué es una heurística.
- Indique bajo qué condiciones el método de búsqueda A^* es optimal.
- Considere el siguiente grafo representando el espacio de búsqueda para un cierto problema, donde A es el estado inicial y K es el único estado meta. Considere como heurística para un nodo el valor que se halla dentro de dicho nodo, junto a su nombre. El costo de los operadores está determinado por los valores que etiquetan los arcos del grafo.



Trazar la ejecución de los métodos *Breadth-First Search* y A^* al ser aplicados sobre el grafo. Indique claramente el orden en que se visitan los nodos y la solución encontrada por cada método.

Consideraciones de importancia relativas a las trazas:

- Deberá efectuarse control de ciclos.
 - Se asume que los nodos vecinos de un nodo N en el espacio de búsqueda se generan en orden alfabético (es decir, el nodo con *menor* etiqueta queda como hijo izquierdo de N en el árbol de búsqueda).
- Para cada uno de los siguientes incisos indique si es posible brindar un espacio de búsqueda tal que se cumpla lo establecido por el inciso. Justifique adecuadamente en cada caso, ya sea efectuando trazas sobre el espacio de búsqueda brindado, o explicando por qué no es posible brindar tal espacio de búsqueda.
 - El método *Depth-First Search* no encuentra una solución y el método *Breadth-First Search* sí.
 - El método *Best-First Search* encuentra una solución y el método *Hill Climbing* no.

4. Prolog

Implemente en PROLOG un predicado `eliminar_vocales/2` que reciba como argumento una lista de letras L (posiblemente conteniendo otras listas anidadas) y retorne la imagen aplanada L_A de L , eliminando además todas las vocales de L .

OBSERVACIÓN: Dada una lista $L = [a, [b, c, e], f, [i, [h]]]$, su imagen aplanada (sin eliminar las vocales) es $L_A = [a, b, c, e, f, i, h]$.

Por ejemplo:

?- eliminar_vocales([a, b, c], Rta).

Rta = [b, c]

?- eliminar_vocales([a, [b, c, e], f, [i, [h]]], Rta).

Rta = [b, c, f, h]

Importante: Deberá implementar TODO predicado que utilice para la resolución del ejercicio.