

Evaluación

Duración: 3 horas

Ejercicio 1 [evaluación individual del laboratorio]

- i. Describa la función objetivo usada en el laboratorio y consideraciones de eficiencia, si se tuvieron, para la implementación de la misma.
- ii. Describa la estrategia minimax.

Ejercicio 2 [30 puntos]

Se quiere resolver el problema de acomodar un conjunto de muebles sobre el piso de una habitación rectangular. Cada mueble se representa mediante un término `mueble(Id,Largo,Ancho)`. La habitación se representa mediante una matriz de M filas y N columnas que puede contener los valores `vacio` o un `Id` de un mueble.

Implemente en Prolog los siguientes predicados:

a)

`ubicar_mueble(+Mueble,+X,+Y,+H,?H2)` \leftarrow `Mueble` es un mueble en el formato anteriormente descrito, x e y son las coordenadas donde se quiere ubicar el mueble. H es la matriz que representa la habitación antes de colocar el nuevo mueble y $H2$ es una matriz igual a H pero con el `Id` de `Mueble` en las celdas que este ocupa.

b)

`ubicar_muebles(+M,+N,+Muebles,?Habitacion)` \leftarrow M y N contienen la cantidad de filas y columnas de la habitación. `Muebles` es una lista con los términos que representan los muebles. `Habitacion` es una solución del problema, que consiste en una matriz que representa la ubicación de todos los muebles en la habitación. Se considera que una solución es correcta si se ubicaron todos los muebles y ningún par de ellos se superpone. Como simplificación, consideramos que los muebles pueden ubicarse en cualquier posición de la habitación, pero no rotarse.

Asuma que cuenta con los siguientes predicados ya implementados para el manejo de matrices. La implementación interna de la matriz no es relevante, debido a que solamente utilizará los predicados para manipularla.

`matrix(+M,+N,+Cell,?Matrix)` \leftarrow `Matrix` es una matriz de M filas y N columnas donde cada celda tiene el valor `Cell`.

`get_cell(+Matrix,+X,+Y,?Cell)` \leftarrow `Cell` es el valor de la celda en la fila x , columna y de la matriz M .

`set_cell(+Matrix,+X,+Y,+Cell,?Matrix2)` \leftarrow `Matrix2` es una matriz igual a `Matrix`, sustituyendo la celda (X,Y) por el valor `Cell`.

Ejercicio 3 [20 puntos]

- a) Indique si las siguientes afirmaciones son verdaderas o falsas. Justifique.
- i. Si $\{F_1, \dots, F_n\}$ es insatisfactible entonces existe i tal que $\{F_i\}$ es insatisfactible.
 - ii. Dado un programa lógico y un objetivo, el conjunto de respuestas correctas depende de la regla de computación.
 - iii. Dado un programa lógico y un objetivo, el conjunto de respuestas correctas incluye al conjunto de respuestas computadas.
- b) Sea el programa P:
- ```

prog(X,Y) <- pred1(X), pred2(Y).
prog(X,X) <- pred2(X).
pred1(X).
pred1(a).
pred1(c).
pred2(a).

```
- i. Dé una respuesta correcta que no sea una respuesta computada, si existe alguna, para  $P \mid - \{ \text{<- prog}(X,Y) \}$ .
  - ii. Construya un árbol SLD para  $P \mid - \{ \text{<- prog}(W,b) \}$ , con regla de computación primer átomo del objetivo.
  - iii. Construya un árbol SLD para  $P \mid - \{ \text{<- prog}(W,b) \}$ , con regla de computación último átomo del objetivo.

**Ejercicio 4** [20 puntos]

Dado el siguiente programa:

```

p(X,X,[X]).
p(X,Y,[X|R]):-
 Y > X,
 X1 is X+1,
 p(X1,Y,R).
p(X,Y,[X|R]):-
 X1 is X-1,
 p(X1,Y,R).

```

- a) Diga qué resultado se obtiene para la consulta  $?- p(3,5,L)$ .
- b) Diga qué resultado se obtiene para la consulta anterior, si se agrega al programa de a):
- i. un cut luego de cada llamada recursiva.
  - ii. un cut en el cuerpo del paso base.
- c) Diga en dónde es necesario poner cuts (la mínima cantidad posible) para obtener una nueva versión del predicado  $p$  que, para las consultas  $?-p(3,5,L)$ ,  $?-p(5,3,L)$  y  $?-p(5,5,L)$ , dé los resultados que se muestran a continuación:
- ```

?- p(3,5,L). --> L=[3,4,5]
?- p(5,3,L). --> L=[5,4,3]
?- p(5,5,L). --> L=[5]
    
```

Ejercicio 5 [15 puntos]

Escriba el siguiente predicado en prolog. La solución no podrá invocarse recursivamente más de n veces, siendo n el largo de la lista original. Se sugiere utilizar listas de diferencias.

$\text{listaCapicua}(+L, ?LCap) \leftarrow$ **LCap** es una lista capicúa con el doble de elementos de **L**, y se cumple que **L** es prefijo de **LCap**.

Ejemplos:

```

listaCapicua([a,b,c], [a,b,c,c,b,a]).
listaCapicua([a,b], [a,b,b,a]).
listaCapicua([a], [a,a]).
listaCapicua([], []).
    
```