

Evaluación (1ª instancia)**Duración:** 3 horas**Pregunta sobre el laboratorio**

- Describa brevemente cómo modeló el tablero de owarelog en su implementación y cuáles son los predicados principales que utiliza para manipularlo.

Ejercicio 1 [25 puntos]

Suponga que tiene un conjunto de fichas de dominó representado mediante una lista de funtores de dos argumentos. Por ejemplo, puede ser el siguiente conjunto de fichas:



Este conjunto se representaría de la siguiente manera en Prolog:

```
Fichas=[f(1,2), f(4,5), f(2,3), f(1,6), f(4,6), f(2,5)]
```

Se denomina una *colocación* de fichas de dominó de largo N a una lista que cumple con las siguientes condiciones:

- N es menor o igual a la cantidad de fichas en el conjunto.
- Cada elemento es una ficha del conjunto. Notar que las fichas pueden rotarse, por lo que si f(2,5) está en el conjunto de fichas, es válido incluir f(2,5) o f(5,2) en la colocación (pero no ambas a la vez).
- No hay fichas repetidas.
- Las fichas respetan la regla de colocación del dominó: el lado derecho de una ficha debe coincidir con el lado izquierdo de la siguiente.

Por ejemplo, para el conjunto de fichas anterior, una colocación válida de largo 4 sería la siguiente: [f(2,1), f(1,6), f(6,4), f(4,5)]

Implemente en Prolog los siguientes predicados:

a) colocacion(+Fichas,?Col) ← Col es una colocación válida de las fichas Fichas, según las reglas definidas anteriormente. Por ejemplo:

```
colocacion([f(1,2), f(4,5), f(2,3), f(1,6), f(4,6), f(2,5)], [f(2,1), f(1,6), f(6,4), f(4,5)]).
```

b) valor(+Col, ?Val) ← Val es el valor de la colocación Col, definiéndose valor como la suma de la cantidad de puntos de todas las fichas de Col. Por ejemplo:

```
valor([f(2,1), f(1,6), f(6,4), f(4,5)], 29).
```

c) mayor_valor(+Fichas, ?Col) ← Col es la colocación de mayor valor para el conjunto de fichas Fichas. Por ejemplo

```
mayor_valor([f(1,2), f(4,5), f(2,3), f(1,6), f(4,6), f(2,5)], [f(2,1), f(1,6), f(6,4), f(4,5), f(5,2), f(2,3)]).
```

Solución:**a)**

```
% posicion_valida/2: una ficha se puede colocar en una orientacion o su opuesta
orientacion_valida(f(Lado1,Lado2),f(Lado1,Lado2)).
orientacion_valida(f(Lado1,Lado2),f(Lado2,Lado1)).
```

```
% colocacion/2: coloca la primera ficha para inicializar la variable Previa
```

```
colocacion(Fichas,[f(Lado1,Lado2)|Resto]):-
  select(F,Fichas,RestoFichas),
  orientacion_valida(F,f(Lado1,Lado2)),
  colocacion(RestoFichas,Lado2,Resto).
```

```
% colocacion/3: coloca fichas comenzando con una que corresponda con el número de la
ficha Previa
```

```
colocacion(_,_,[]).
colocacion(Fichas,Previa,[f(Previa,Lado2)|Resto]):-
  select(F,Fichas,RestoFichas),
  orientacion_valida(F,f(Previa,Lado2)),
  colocacion(RestoFichas,Lado2,Resto).
```

b)

```
valor(Col,Val):-
  valor(Col,0,Val).
valor([],Acum,Acum).
valor([f(Lado1,Lado2)|Resto],Acum,Val):-
  Acum2 is Acum + Lado1 + Lado2,
  valor(Resto,Acum2,Val).
```

c)

```
mayor_valor(Fichas,Col):-
  findall((Col,Val),(colocacion(Fichas,Col),valor(Col,Val)),[(Col1,Val1)|Colocaciones]),
  mayor_valor(Colocaciones,Col1,Val1,Col).
```

```
mayor_valor([],AcumCol,_,AcumCol).
mayor_valor([(Col1,Val1)|Resto],_,AcumVal,Col):-
  Val1 > AcumVal,!,
  mayor_valor(Resto,Col1,Val1,Col).
mayor_valor([(_,Val1)|Resto],AcumCol,AcumVal,Col):-
  % Val1 =< AcumVal,
  mayor_valor(Resto,AcumCol,AcumVal,Col).
```

Ejercicio 2 [20 puntos]

a) Indique si son verdaderas o falsas las siguientes afirmaciones. Fundamente.

- i) El mgu de un conjunto de expresiones (cuando existe) siempre es único.
- ii) Las cláusulas de Horn contienen al menos un literal positivo.
- iii) Un objetivo definido no puede contener ningún literal positivo.
- iv) Dado un programa lógico y un objetivo, si no existen respuestas computadas, entonces no existen respuestas correctas.

b) Aplique el algoritmo para hallar el m.g.u. de las siguientes dos expresiones, mostrando su aplicación paso a paso:

$$q(g(Y,b),f(X,X),Y) \quad q(W,f(Z,h(W)),a)$$

Solución:

a)

- i) Falso: Por ejemplo, si $\{X|Y\}$ fuera el mgu de un conjunto de expresiones, $\{Y|X\}$ también lo sería.
- ii) Falso: Por definición las cláusulas de Horn son cláusulas definidas (con un literal positivo) u objetivos definidos (que no tienen ningún literal positivo).
- iii) Verdadero: Por definición de objetivo definido.
- iv) Verdadero: como las respuestas correctas siempre pueden expresarse como la composición de una respuesta computada con otra sustitución, si no hay respuestas computadas no hay respuestas correctas.

b)

$$\{ q(g(Y,b),f(X,X),Y) = q(W,f(Z,h(W)),a) \} \quad \mu = \varepsilon$$

$$\{ g(Y,b)=W, f(X,X)=f(Z,h(W)), Y=a \} \quad \mu = \varepsilon \circ \{ W/g(Y,b) \}$$

$$\{ f(X,X)=f(Z,h(g(Y,b))), Y=a \} \quad \mu = \{ W/g(Y,b) \}$$

$$\{ X=Z, X=h(g(Y,b)), Y=a \} \quad \mu = \{ W/g(Y,b) \} \circ \{ X/Z \}$$

$$\{ Z=h(g(Y,b)), Y=a \} \quad \mu = \{ W/g(Y,b), X/Z \} \circ \{ Z/h(g(Y,b)) \}$$

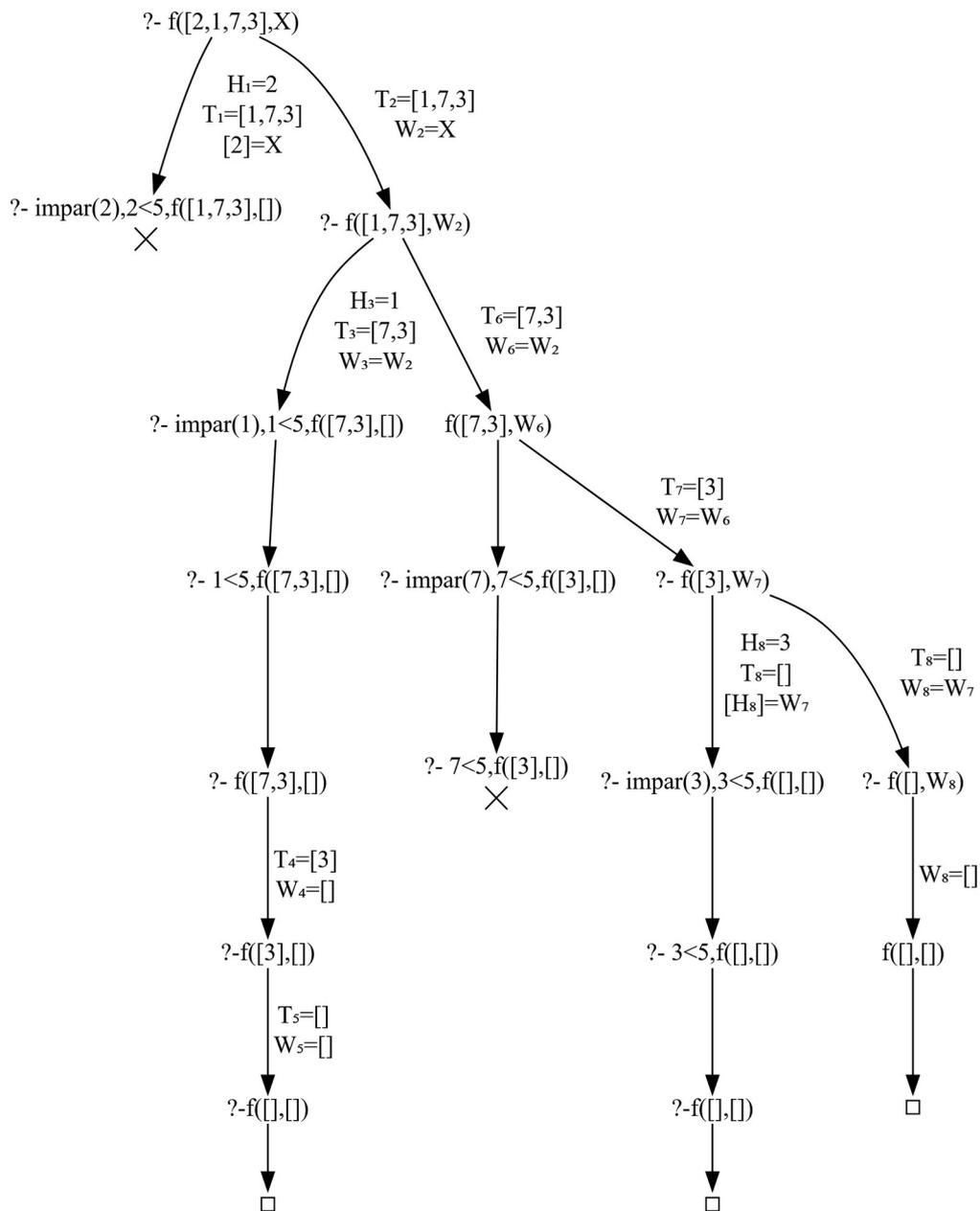
$$\{ Y=a \} \quad \mu = \{ W/g(Y,b), X/h(g(Y,b)), Z/h(g(Y,b)) \} \circ \{ Y/a \}$$

$$\{ \} \quad \mu = \{ W/g(a,b), X/h(g(a,b)), Z/h(g(a,b)), Y/a \}$$

Salida: $\mu = \{ W/g(a,b), X/h(g(a,b)), Z/h(g(a,b)), Y/a \}$

Ejercicio 3 [25 puntos]

a) Dibuje el árbol SLD correspondiente a la consulta $?- f([2,1,7,3],X)$ suponiendo que la regla de computación toma el átomo de más a la izquierda para la resolución.



b) ¿Qué respuestas dará el intérprete Prolog para el objetivo anterior?

$?- f([2,1,7,3],X).$
 $X = [1] ;$
 $X = [3] ;$
 $X = [] .$

c) ¿Cambian las respuestas si la primera cláusula se sustituye por la regla siguiente? Justifique.

$f([], []) :- !.$

No, no cambia. La cláusula es un hecho (por lo que no hay variables a satisfacer), y ninguna de las otras cláusulas del predicado matchean cuando los dos argumentos son listas vacías.

d) ¿Qué respuestas dará el intérprete Prolog si se agrega un cut luego del objetivo $H < 5$ en la segunda cláusula? ¿Cambia su respuesta si el cut se pone en otra posición del lado derecho? ¿Es un cut rojo o verde? Justifique.

```
?- f([2,1,7,3],X).
X = [1].
```

El cut está podando la rama derecha de $f([1,7,3],W)$, y por lo tanto no hay otras asignaciones posibles para X luego de satisfacer el primer objetivo (y llegar al cut). Como hay soluciones que no se están dando, el cut es rojo.

Si el cut se pone al comienzo del lado derecho de la regla, entonces se poda la rama derecha de la raíz, y no hay soluciones.

Si el cut se cambia de lugar el resultado es el mismo que en el primer caso, porque los dos primeros objetivos se satisfacen solamente una vez (y por lo tanto no hay ramas alternativas a explorar, y por lo tanto podar, en esa rama del árbol).

e) ¿Qué respuesta dará el intérprete Prolog a la siguiente consulta, a partir del programa original? `?- findall(X, f([2,1,7,3,11,4,9,1,3,2],X),L).`

```
L = [[1], [3], [1], [3], []].
```

Ejercicio 4 [15 puntos]

a) Implemente el siguiente predicado en Prolog de manera eficiente utilizando listas de diferencias:

`solo_pares(+MultiLista,?Pares)` ← MultiLista es una estructura compuesta por listas de números enteros con cualquier nivel de anidamiento, y Pares es una lista plana que solamente contiene los números pares de MultiLista.

Ejemplo: `solo_pares([14,[5,4],[2,[8,3,[],10]],6],[14,4,2,8,10,6]).`

b) Utilizando DCG, implemente un programa en Prolog para reconocer el siguiente lenguaje:

$$L = \{ a^n b^{2n} c^{3n} / n \geq 0 \}$$

Solución:

a)

```
solo_pares(MultiLista,Pares):-
    solo_pares_ld(MultiLista,Pares-[]).

solo_pares_ld([],Pares-Pares).
solo_pares_ld([Num|RestoListas],[Num|Pares]-Resto):-
    number(Num), 0 is Num mod 2, !,
    solo_pares_ld(RestoListas,Pares-Resto).
solo_pares_ld([Num|RestoListas],Pares-Resto):-
    number(Num),!, % 1 is Num mod 2,
    solo_pares_ld(RestoListas,Pares-Resto).
solo_pares_ld([MultiLista|RestoListas],ListaPares-Resto):-
    solo_pares_ld(MultiLista,ListaPares-RestoPares),
    solo_pares_ld(RestoListas,RestoPares-Resto).
```

b)

```
s --> a(N), {N2 is 2*N}, b(N2), {N3 is 3*N}, c(N3).
a(0) --> [].
a(N) --> [a], a(N1), {N is N1 + 1}.
b(0) --> [].
b(N) --> [b], b(N1), {N is N1 + 1}.
c(0) --> [].
c(N) --> [c], c(N1), {N is N1 + 1}.
```