

Evaluación

Duración: 3 horas

Ejercicio 1 [evaluación individual del laboratorio]

Responda en pocas líneas las siguientes preguntas en base a la implementación del laboratorio realizada por su grupo:

- (i) ¿Qué predicados utilizaron para incorporar nueva información a la base de conocimiento? ¿Estos predicados tienen algún inconveniente?
- (ii) ¿Tuvieron en cuenta el criterio de *last-call optimization*? ¿En qué momento? ¿Esto mejoró el tiempo de ejecución?
- (iii) ¿Cómo identificaron los nombres propios en el reconocimiento de texto libre?

Ejercicio 2 [20 puntos]

- a) Indique si las siguientes afirmaciones son verdaderas o falsas. Justifique brevemente.
- i. Si un conjunto de expresiones tiene un mgu, este es único.
 - ii. Dados un programa lógico y un objetivo, podemos construir un único árbol SLD.
 - iii. Dado un programa lógico y un objetivo, se cumple que toda respuesta correcta es una respuesta computada.
- b) Para los siguientes pares de términos dar un mgu, si existe.
- i. $\{ p(f(a, x), g(y), z), p(z, w, f(v, y)), p(u, g(f(a, b)), t) \}$
 - ii. $\{ a(f(y), w, g(z)), a(x, y, x) \}$
 - iii. $\{ q([X|Xs]), q([a, b, c|Ys]) \}$
- c) Sea el programa P:
- ```

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

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

### Ejercicio 3 [20 puntos]

Considere el siguiente programa:

```

p(X) :- X > 2, !, a(X, X1), p(X1).
p(X) :- X > 0, b(X).
p(0).
a(X, Y) :- X > 2, Y is X - 1.
a(X, Y) :- Y is X - 4.
b(3).
b(1).

```

- a) Diga cuáles de las siguientes consultas dan true: ?- p(0), ?- p(1), ?- p(2), ?- p(3), ?- p(4).

b) Considere las siguientes variantes del programa:

- |                                                                                                                                                                                                                                                                                                 |                                                                                                                                                                                                                                                                                               |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <p>i. <math>p(X) :- X &gt; 2, !, a(X, X1), p(X1).</math><br/> <math>p(X) :- X &gt; 0, b(X).</math><br/> <math>p(0).</math><br/> <math>a(X, Y) :- X &gt; 2, !, Y \text{ is } X - 1.</math><br/> <math>a(X, Y) :- Y \text{ is } X - 3.</math><br/> <math>b(3).</math><br/> <math>b(1).</math></p> | <p>ii. <math>p(X) :- X &gt; 2, a(X, X1), p(X1).</math><br/> <math>p(X) :- X &gt; 0, !, b(X).</math><br/> <math>p(0).</math><br/> <math>a(X, Y) :- X &gt; 2, Y \text{ is } X - 1.</math><br/> <math>a(X, Y) :- Y \text{ is } X - 4.</math><br/> <math>b(3).</math><br/> <math>b(1).</math></p> |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

¿Cuáles de las consultas dadas en a) dan true para cada variante?

c) Defina cut rojo y cut verde. Indique, para cada uno de los cuts de las tres variantes del programa dado, si se trata de cuts verdes o rojos. Justifique.

**Ejercicio 4 [20 puntos]**

a) Considere el siguiente programa R para Prolog puro, proposicional:

```
p :- a,b.
a :- b.
a :- r.
b :- p.
b :- s.
r.
s.
```

- i. ¿p es consecuencia lógica de R?
- ii. Explique por qué no se obtiene una respuesta 'true' al plantear la consulta  $\leftarrow p$ .

b) Escriba un meta-intérprete para Prolog proposicional que detecte la repetición de un átomo previamente reducido, evitando así entrar en loop. El meta-intérprete, cada vez que está por entrar en loop, escribe un mensaje con el texto 'cuidado: loop' y el nombre del predicado. El predicado a implementar es loop\_solve(+Obj,+Historia), en la invocación inicial Historia es la lista vacía.

c) Muestre los mensajes escritos ante la consulta loop\_solve(p,[]). ¿Es exitosa la consulta?

d) ¿Cuál es la conducta de su meta-intérprete ante la consulta p(0), si el programa que toma como entrada es:

```
p(X) :- p(s(X)).
```

**Ejercicio 5 [25 puntos]**

Se piden varios predicados que permiten resolver un sudoku simple (que se define más adelante). Para todo el ejercicio, se asume que M es una matriz N X N, representada del siguiente modo: [Fila1, Fila2, ..., FilaN]. Cada fila es una lista de N elementos.

Escriba los predicados siguientes:

- |                                           |                                                                                                                                                                                                                             |
|-------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <p>fila_completa_valida(+N, ?F) &lt;-</p> | <p>F es una lista con valores entre 1 y N sin repetidos. Se asume que al ser invocado este predicado, F es una lista con algunos elementos instanciados y otros sin instanciar.</p>                                         |
| <p>col_i_esima(+N, +M, +I, -C) &lt;-</p>  | <p>C es la columna I-ésima de M.</p>                                                                                                                                                                                        |
| <p>sudoku_simple(+N, ?M) &lt;-</p>        | <p>M es una matriz N X N que contiene números de 1 a N. Los números no se repiten en una misma fila ni en una misma columna. M puede estar completamente instanciada o puede contener valores instanciados y variables.</p> |