



Programación Funcional y Lógica
Lic. en Sistemas de Información
Primera Evaluación Parcial

Alumno: **Apellido y Nombre**

Leg: **Número**

04/06/2021

Condición de Aprobación:

Menos de 60 => Desaprobado

60 pts => 4 (cuatro)

80 pts => 7 (siete)

LÓGICA PROPOSICIONAL / DE PREDICADOS / CLAUSAL

1. Indicar si los siguientes conjuntos de premisas son consistentes (C) o inconsistentes (I):

8 pts.

- a. $b \Rightarrow \neg e$
 $p \Rightarrow b$
 $a \Rightarrow b$
- b. $p \vee d \Rightarrow b$
 $e \wedge a$
 $b \Rightarrow \neg a$
 $e \Rightarrow p$

2. Indicar si los siguientes razonamientos son válidos (V) o inválidos (I). De los que son válidos, ¿Cuáles son las reglas de inferencia, y en qué orden, se deben aplicar para demostrar que el siguiente razonamiento es válido?

10 pts.

- a. $[(p \Rightarrow q) \wedge (r \Rightarrow q) \wedge p] \Rightarrow (r \Rightarrow p)$
- b. $[(p \Rightarrow q) \wedge \neg(t \vee r) \wedge (t \vee q)] \Rightarrow p$
- c. $[(\neg r \vee \neg b) \wedge (t \vee s \Rightarrow r) \wedge (b \vee \neg s) \wedge \neg t] \Rightarrow \neg(t \vee s)$

3. Dado el siguiente programa:

12 pts.

ganado :- vaca.
vaca :- lechera, ganado.
:- cabra.
lechera.

Indicar cuales de las siguientes cláusulas (son/no son) consecuencia lógica del programa:

- a. vaca; ganado :- lechera.
- b. lechera; cabra :- vaca, ganado.
- c. lechera; ganado :- vaca.

10 pts.

4. Aplicando unificación y resolución proposicional ¿Cuáles son las reglas de inferencia, y en qué orden, se deben aplicar para demostrar que el siguiente razonamiento es válido?

1. $\forall (x): [P(x) \Rightarrow (T(x) \vee G(x))]$

2. $\forall (x): (T(x) \Rightarrow H(x))$

3. $\forall (x) (N(x) \wedge \neg H(x))$

$\therefore N(b) \wedge G(d)$

5. Considérese el siguiente programa:

10 pts.

tieneLetra(opera).

cancion(X) :- tieneRitmo(X), tieneLetra(x).

:- cancion(ruido).

Describir el Universo de Herbrand, la base de Herbrand, las cláusulas fijas y las interpretaciones de Herbrand que son modelos del programa.

PARADIGMA LÓGICO

6. Escribir una predicado que relacione una lista de caracteres y una longitud de palabra, con la cantidad de palabras de esa longitud que hay en la cadena.

20 pts.

Definir un predicado que relacione una lista que representa nuestro String con la cadena limpia.

% cantPalabras(listaDeCaracteres, n, CantidadDeaPalabras).

?- cantPalabras([v, i, v, a, esp, l, a, esp, v, i, d, a, esp, e, n, esp, c, a, s, a], 4, Cantidad).

> Cantidad = 3

?- cantPalabras([v, i, v, a, esp, l, a, esp, v, i, d, a, esp, e, n, esp, c, a, s, a], 3, Cantidad).

> Cantidad = 0

30 pts.

- 7.** Supongamos ahora que contamos con una base de hechos que representan puertas de una casa que unen 2 habitaciones, y deseamos hacer una lista de todos los caminos existentes de una habitación a otra. Por ejemplo :

puerta(cocina, living).

puerta(cocina, patio).

puerta(patio, dormitorio1).

puerta(living, dormitorio1).

puerta(living, dormitorio2).

Escribir un predicado que relacione dos habitaciones de la casa con un camino que una ambas habitaciones. El camino se representará como una lista de habitaciones por donde se debe pasar para salir desde la habitación origen y llegar a la habitación destino, sin pasar 2 veces por la misma habitación.

%recorrido(Origen, Destino, Camino).

?- recorrido(cocina, dormitorio2, Camino).

Camino = [cocina, living, dormitorio2];

Camino = [cocina, patio, dormitorio1, living, dormitorio2]

Nota1: La puerta es bidireccional. Tener cuidado de no generar un bucle.

Nota2: Para evitar bucles infinitos es necesario controlar que no se repitan habitaciones en el camino.