

PLP - Recuperatorio del Segundo Parcial - 2^{do} cuatrimestre de 2022

Este examen se aprueba obteniendo al menos dos ejercicios bien menos (B-) y uno regular (R). Las notas para cada ejercicio son: -, I, R, B-, B. Entregar cada ejercicio en hojas separadas. Poner nombre, apellido y número de orden en todas las hojas, y numerarlas. Se puede utilizar todo lo definido en las prácticas y todo lo que se dio en clase, colocando referencias claras. El orden de los ejercicios es arbitrario. Recomendamos leer el parcial completo antes de empezar a resolverlo.

Ejercicio 1 - Programación Lógica

- a) Definir el predicado `sublistaMasLargaDePrimos(+L,?P)` que es verdadero cuando `P` es una sublista de `L` que contiene la mayor cantidad de números primos contiguos. Puede haber más de una solución. Por ejemplo:

```
?- sublistaMasLargaDePrimos([2,3,4,5,6,13,7],S).
S = [2,3];
S = [13,7];
false.
```

- b) Dado el predicado `simbolo(?S)` que instancia en `S` ciertos elementos de un alfabeto, se pide definir el predicado `clausura(-L)` que es verdadero cuando `L` es la lista que representa la clausura de Kleene¹ del lenguaje conformado por los símbolos `S`. Por ejemplo, si `a` y `b` son los únicos elementos que hacen verdadero a `simbolo`, entonces:

```
?- clausura(L).
L = [];           L = [b,a];
L = [a];          L = [b,b];
L = [b];          L = [a,a,a];
L = [a,a];        L = [a,a,b];
L = [a,b];        ...
```

- c) El predicado del inciso anterior, ¿es reversible en `L`? Justificar.

Ejercicio 2 - Resolución

Considerar las siguientes definiciones en prolog:

```
insertar(X,L,LX) :- append(L1,L2,L), append(L1,[X|L2],LX).

append([],L,L).
append([X|XS],YS,[X|L]) :- append(XS,YS,L).
```

y la consulta:

```
?- insertar(3,[1,2],[1,3,R]).
```

- a) Convertir la base de conocimientos y la consulta a forma clausal.
Sugerencia: pueden pensar la lista `[1,2]` como `[1|[2|[]]]`
- b) Utilizar el método de resolución para obtener el resultado de la consulta.
- c) La resolución utilizada en el inciso anterior, ¿fue SLD? Justificar.

¹La Clausura de Kleene es una operación de los lenguajes que representa el conjunto de cadenas que se pueden formar con los símbolos del lenguaje. Las cadenas pueden ser vacías e incluir repeticiones.

Ejercicio 3 - Objetos

a) Sea el siguientes código escrito en JavaScript:

```
let a = { v: 1, f: function(x){ return this.v + x; } };  
let b = Object.create(a);  
let c = { v: 2 };  
Object.assign(b, c);  
console.log(b.f(3));
```

Indicar qué mensajes se envían durante la ejecución de la última línea, cuál es el objeto receptor de cada mensaje, y qué valor se imprime en la consola. (La suma no se considera un mensaje, ya que en JavaScript los números no son objetos).

b) Sean a, b y c los siguientes objetos del cálculo Sigma:

$$a \stackrel{\text{def}}{=} [v = \text{uno}, f = \zeta(z) \lambda(x) \text{ suma } (z.v) (x)]$$
$$b \stackrel{\text{def}}{=} [v = \text{dos}, f = a.f]$$
$$c \stackrel{\text{def}}{=} [v = \text{dos}]$$

Decidir si estos objetos se comportan de la misma manera que los correspondientes a, b y c del punto anterior, luego de haber ejecutado la línea del `Object.assign()`. Justificar.

Mostrar cómo reduce la expresión `b.f(cero)`. Se puede usar la regla APP vista en clase. Asumir que `suma (uno) (cero) → uno` y `suma (dos) (cero) → dos`. Consideramos que `cero`, `uno` y `dos` son valores.

$$\frac{f \rightarrow g \quad g \equiv \lambda(x)t \quad t\{x \leftarrow a\} \rightarrow v}{f(a) \approx v} \text{ [APP]}$$