

## PLP - Recuperatorio segundo parcial - 1<sup>er</sup> cuatrimestre de 2007

Se aprueba obteniendo al menos 70 puntos. Poner nombre, apellido y número de orden en cada hoja, y numerarlas. Se puede utilizar todo lo definido en las prácticas y todo lo que se dio en clase, colocando referencias claras.

**Ejercicio 1 - (30 puntos)** Dado un grafo  $G$  (no dirigido), representado por una lista de nodos y una lista de ejes (cada eje está dado por un par de nodos), se desea hallar con Prolog un cubrimiento mínimo de ejes por nodos. Vale decir, se busca un conjunto  $S$  de nodos de  $G$  tal que todos los ejes del grafo incidan en  $S$  (tengan algún extremo en  $S$ ) y además  $S$  sea lo más chico posible.

Escribir los siguientes predicados respetando en cada caso la instanciación pedida. Ninguno de los predicados perdidos debe dar soluciones repetidas. No utilizar cut (!) ni predicados de alto orden (como `setof`). La única excepción es el `not`, que está permitido.

- a) **(5 puntos)** Escribir el predicado `incide(+G, +N, +S)` que evalúa si en el grafo  $G$ , si algún eje de  $S$  incide en el nodo  $N$ .

Ejemplos

```
incide([a, b, c], [(a,b), (b,c), (a,c)], a, [(a,b), (a,c), (d,f)]).
```

Yes

```
incide([a, b, c], [(a,b), (b,c), (a,c)], a, [(a,b), (a,c), (b, c), (d,f)]).
```

No

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

No

- b) **(9 puntos)** Escribir el predicado `cubre(+G, +S)` que evalúa si el conjunto de nodos  $S$  cubre a todos los ejes de  $G$ .

Ejemplos

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

No

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

Yes

- c) **(16 puntos)** Escribir el predicado `minCub(+G, -M)` que determina en  $M$  un mínimo cubrimiento de  $G$  (podría haber varias soluciones).

Ejemplo

```
cubrimientoMinimo([a, b, c], [(a,b), (b,c), (a,c)], C).
```

C = [b, c] ;

C = [a, c] ;

C = [a, b] ;

No

**Ejercicio 2 - (20 puntos)** Utilizando resolución *SLD* si es posible, o general si no, determinar si es tautología, contradicción, o nada:

$$[(\forall x \exists y P(x, y)) \wedge (\forall x \forall y P(x, y) \Rightarrow P(y, x))] \Rightarrow (\forall y \exists x P(x, y))$$

### Ejercicio 3 - (30 puntos)

Se cuenta con la clase `Arbol` cuyos objetos representan a los árboles n-arios no vacíos. Estos tienen un nodo y una `OrderedCollection` de ramas. Las variables de instancia son `nodo` y `ramas`. La clase cuenta con los siguientes métodos:

```
nodo
^nodo

nodo:aNodo
nodo := aNodo.

ramas
^ramas

ramas:aRamas
ramas := aRamas.

initialize: aNodo ramas: aRamas
  self nodo: aNodo.
  self ramas: aRamas.
  ^self.

new: aNodo ramas:aRamas
  ^ (super new) initialize: aNodo ramas: aRamas.
```

Implementar el mensaje `arbolConAltura` que dado un árbol devuelva otro árbol que contenga del primero sólo aquellas ramas de longitud máxima.

**Ejercicio 4 - (20 puntos)** Contar cuántos tipos `T` existen tales que:

```
Top -> {x: Bool, y:Nat, z:Int} <: T <: (Nat -> Top) -> {x:Int,y:Int}
```

Nota: Las permutaciones de un registro con los mismos labels se consideran un mismo tipo.