

# Lógica

## Práctica 3: Sustitución

### Enunciado

En esta práctica implementaremos en prolog la operación de **sustitución de una variable por un término**. Para ello, utilizaremos los siguientes operadores:

```
:- op(1060, yfx, <=>). % doble implicación
:- op(1050, yfx, <->). % implicación hacia la izquierda
:- op(600, yfx, v). % disyunción
:- op(400, yfx, &). % conjunción
:- op(200, fy, ~). % negación
:- op(200, fx, forall). % universal
:- op(200, fx, exists). % existencial
:- op(300, xfy, ::). % separador para cuantificadores
```

de modo que para escribir  $\forall x (p(x) \wedge \exists y q(x,y))$  usaremos la notación `forall x:: ( p(x) & exists y:: q(x,y) )`.

Para simplificar, supondremos que el término que introducimos **no contiene variables ya cuantificadas en F**.

### Apartado 1. Predicado `subs`

El predicado a implementar se denominará `subs(X/T,F,G)` donde **x** es la variable a reemplazar, **t** es el término por el que cambiaremos **las apariciones libres de X**, **F** es la expresión original sobre la que queremos realizar el reemplazo y, por último, **G** es el resultado del reemplazo.

El predicado `subs(X/T,F,G)` deberá trabajar del mismo modo tanto si **F** es una fórmula como si es un término.

Algunos ejemplos de ejecución serían:

```
?- subs(x/f(3), g(x), G).
G = g(f(3)).

?- subs(x/f(4), (forall x:: p(x,y) & p(x)), G).
G = forall x::p(x, y)&p(f(4)).

?- subs(x/3, forall y:: ( p(x,y) => exists x:: ~ q(y,x) ), G).
G = forall y:: (p(3, y)->exists x:: ~q(y, x)).
```

### Apartado 2. Predicado `subs_list`

En un segundo apartado, implementaremos el predicado `subs_list(Rs,F,G)` donde **Rs** es una lista de sustituciones, como por ejemplo `{x/3, y/f(z), z/8}`. Este predicado se apoyará en el anterior de `subs` modo que deberá ejecutar todas las sustituciones de la lista por orden de izquierda a derecha: primero cambiar `x/3` luego, sobre el resultado, cambiar `y/f(z)` y, por último, en lo que nos dé, cambiar `z/8`.

Algunos ejemplos de ejecución serían:

```
?- subs_list([x/3,y/f(z),z/8], p(x,y), G).
G = p(3, f(8)).

?- subs_list([x/f(x),z/f(x)], q(x) v forall x:: p(x,y), G).
G = q(f(f(x)))v forall x::p(x, y).
```

### Evaluación y Entrega

La nota máxima para este ejercicio es de **1.5 puntos = 15% del curso**. Entrega: usar tarea MOODLE correspondiente. Fecha limite: **Lunes 15 de mayo, 23:59**. Las prácticas se pueden realizar por parejas o de forma individual. Si se realiza por parejas, es suficiente con que uno de los dos componentes realice la entrega en moodle, pero el fichero fuente debe contener en comentarios en las primeras líneas los nombres de los dos alumnos que formen la pareja. Se debe entregar **un único fichero**, "**practica2.pl**" con todos los predicados que se requieren en el enunciado.