**COMPUTABILIDAD**

1. **(1,5 puntos)** Determinar la **función semántica binaria** asociada al siguiente programa while P. Este programa utiliza el programa Pf, que calcula la función binaria $f(x,y)$ y tiene k variables, y el programa Pg, que calcula la función unaria $g(x)$ y también tiene k variables.

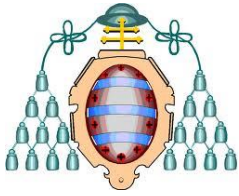
```
begin
  Xk+1 := X1;
  Xk+2 := X2;
  X1 := X1+X1;
  Pf;
  X1 := succ(X1);
  X2 := 0; ...; Xk := 0;
  Pg;
  X1 := X1 * Xk+2;
end
```

2. **(1,2 puntos)** Completa los cuatro huecos del siguiente programa while de modo que su función semántica binaria sea $f(x,y) = \sum_{i=1}^y (x \div i)$. Cada hueco contiene una sola instrucción. Las únicas macros permitidas son las de asignación y suma.

```
begin
  (hueco 1)
  X4 := 0;
  X7 := 0;
  while X3 ≠ X7 do
    begin
      (hueco 2)
      X6 := X1;
      while X5 ≠ X3 do
        begin
          (hueco 3)
          X5 := succ(X5);
        end
        (hueco 4)
        X3 := pred(X3);
      end
      X1 := X4;
    end
  end
```

3. **(0.8 puntos)** Indica, para el test $(X > Y \geq Z)$, una expresión que devuelva un número mayor que 0 cuando el test es verdadero y devuelva 0 cuando es falso.
4. **(1.5 puntos)** Determina la función semántica unaria que computa la siguiente máquina de Turing $M = (\{0,1\}, \{q_0, q_1, q_2, q_3, q_f\}, T, q_0, \{q_f\})$, con T:

$(q_0, 1, 0, D, q_1)$	$(q_2, 1, 0, D, q_3)$	$(q_3, 1, 1, D, q_1)$
$(q_1, 1, 0, D, q_2)$	$(q_2, 0, 0, N, q_3)$	$(q_3, 0, 0, D, q_3)$
$(q_1, 0, 1, H, q_f)$		



5. (1.5 puntos) Dada una Máquina de Turing $M_g = (\{0,1\}, \{p_0, \dots, p_f\}, T_g, p_0, \{p_f\})$ con función semántica unaria $g(x)$, completa los cinco huecos de la siguiente Máquina de Turing M_f para que compute la función:

$$f(x, y) = \begin{cases} g(x) & \text{si } x > 0 \\ g(y) & \text{si } x = 0 \end{cases}$$

$M_f = (\{0,1\}, (\text{hueco1}))$

(q0, 1, 1, D, q1)

(q1, 0, 0, l, q2)

(q2, (*hueco2*))

(q2, (*hueco3*))

(q1, 1, 1, D, q3)

(q3, 1, 1, D, q3)

(q3, 0, 0, D, q4)

(q4, (*hueco4*))

(q4, 0, 0, l, q5)

(q5, 0, 0, l, q5)

(q5, 1, 1, l, q6)

(q6, 1, 1, l, q6)

(q6, (*hueco5*))

6. Se pretende demostrar que el problema de decidir si un programa while P calcula la función binaria $f(x, y) = x * y$.

- a) (2 puntos) Rellena los huecos que faltan en el proceso de demostración por reducción:

Suponemos que existe un algoritmo A cuya función semántica es:

y definimos la macro $A(X)$ a partir de él.

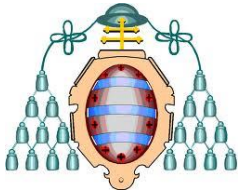
Hacemos ahora este programa P_d , que nos permitirá más adelante reducir el problema de la parada (se permite usar todas las macros vistas en clase):

y cuya función semántica es:

Aplicando el teorema de parametrización, el código d se puede calcular mediante una función $f(c, k)$ total y computable. Dado que es total y computable, se puede definir una macro $F(X, Y)$ que la compute.

A partir de las macros A y F podemos ahora definir el siguiente programa:

```
begin
  X1 := F(X1, X2);
  X1 := A(X1);
end
```



cuya función semántica es:

Dada la definición de P_d , esta función es equivalente a la función característica del problema de la parada. Por tanto, hemos encontrado una contradicción, ya que...

, por lo que nuestro problema es irresoluble.

- b) **(0.5 puntos)** Demuestra la irresolubilidad del problema utilizando el Teorema de Rice (se permite usar todas las macros vistas en clase).

7. (1 punto, pero si la respuesta es incorrecta resta 0,3 puntos) El teorema de **Universalidad**:

- a) Asegura la existencia de una única función computable capaz de simular todas las funciones computables.
- b) Asegura que para cada aridad j , existe una función computable de aridad $j+1$ que simula todas las funciones computables de aridad j .
- c) Asegura la existencia de funciones que no son computables.
- d) Asegura la computabilidad de una función construida cambiando algunas de las variables de otra función computable por constantes.