

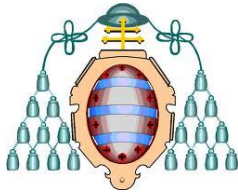
1. (2 puntos) Indica cuáles serían las funciones semántica unaria y binaria del siguiente programa while P, donde Pg representa otro Programa While k-variables cuya función semántica binaria es g(x,y) para todo x, y:

```
begin
  X1 := X1 + X1;
  Xk+1 := X2;
  Pg;
  X2 := 5;
  while X2 ≠ Xk+1 do
  begin
    X1 := succ(X1);
    Xk+1 := pred(Xk+1)
  end
  X1 := X1 + X2
end
```

2. (1.5 puntos) Completa los 5 huecos del siguiente programa Programa While de forma que compute la función dada. Cada hueco puede contener más de una instrucción. Se permiten las macros de la suma, resta, producto y asignación:

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

```
begin
  [ ]
  X3 := 0;
  while [ ] ≠ X3 do
  begin
    X1 := X2;
    [ ]
  end
  while [ ] ≠ X3 do
  begin
    X1 := 0;
    [ ]
  end
end
```



3. (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 binaria semántica  $g(x,y)$ , completa la Máquina de Turing  $M_f$  siguiente para que compute la función  $f(x,y) = g(0, x+y)$ . Se deben completar las 4 transiciones incompletas y la quintupla que define  $M_f$ :

$$M_f = (\{0,1\}, \quad , \quad , q_0, \quad )$$

Tf :

$(q_0, 1, 1, D, q_0)$	$(q_1, 0, 0, I, q_2)$	<b><math>(q_3, \quad )</math></b>
<b><math>(q_0, \quad )</math></b>	<b><math>(q_2, \quad )</math></b>	$(q_4, 1, 0, I, q_5)$
$(q_1, 1, 1, D, q_1)$	$(q_3, 1, 1, I, q_3)$	<b><math>(q_5, \quad )</math></b>

4. (1.5 puntos) Determina la función semántica unaria que computa la siguiente máquina de Turing ( $q_0$  estado inicial y  $q_f$  único estado final).

$(q_0, 1, 1, D, q_1)$	$(q_2, 1, 0, D, q_3)$	$(q_4, 1, 1, I, q_4)$
$(q_1, 1, 1, D, q_0)$	$(q_2, 0, 1, H, q_f)$	$(q_4, 0, 1, I, q_2)$
$(q_0, 0, 0, I, q_5)$	$(q_3, 1, 1, D, q_3)$	$(q_5, 1, 0, I, q_5)$
$(q_1, 0, 0, I, q_2)$	$(q_3, 0, 1, I, q_4)$	$(q_5, 0, 0, N, q_2)$

5. Se pretende demostrar que el problema de decidir si un programa while  $P$  cumple que su función semántica binaria es 10 si, y solo si,  $x > y$  es irresoluble.

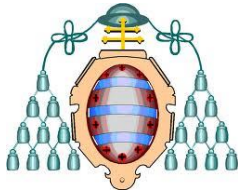
- a) (2 puntos) Rellena los huecos que faltan en el proceso de demostración por reducción:

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, utilizando el Teorema de Rice, que el problema descrito en el apartado anterior es irresoluble (se permite usar todas las macros vistas en clase).

6. **(1 punto, incorrecta resta 0.3)** Consideramos un programa while 'e' tal que  $\varphi_e(x, y, z) = x^z + y$ . Basándonos en el teorema de parametrización, ¿cuál de las siguientes opciones podemos asegurar?

- a) La función  $g(z) = 2^z$  es computable y el código del programa que la computa es  $S_1^2(e, 2, 0)$
- b) La función  $g(z) = 3^z$  es computable y el código del programa que la computa es  $S_1^1(e, 3)$
- c) La función  $g(x, y) = x^3 + 2$  es computable y el código del programa que la computa es  $S_1^2(e, 2, 3)$
- d) La función  $g(x, y) = x^2 + y$  es computable y el código del programa que la computa es  $S_2^1(e, 2)$