

Computabilidad

Apellidos, Nombre: DNI:

1. (2 puntos) Sea un programa P1, que utiliza exactamente k variables, y cuya función unaria semántica es $\varphi_{P1}^{(1)}(x) = g(x)$. Completa las TRES instrucciones que faltan en el siguiente programa while P, de tal forma que su función unaria semántica sea:

$$\varphi_P^{(1)}(x) = \sum_{i=1}^x g(i)$$

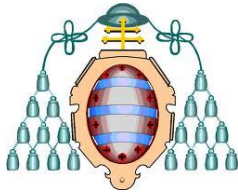
Nota: se permite utilizar la macro de la suma y de la asignación.

```
begin
  Xk+1 := X1;
  Xk+2 := 0;
  Xk+3 := 0;
  while Xk+2 != Xk+1 do
    begin
      
      
      X2 := 0; X3 := 0; ... ; Xk := 0;
      P1;
      
    end
  end
  X1 := Xk+3;
end
```

2. (1,25 puntos) Construye una expresión para el test $\neg((X \leq Y) \wedge (Y < Z))$, de tal forma que la expresión devuelva un número mayor que 0 si el test es verdadero y 0 si es falso.

3. (1,75 puntos) Indica la función binaria semántica de la siguiente máquina de Turing, siendo q0 su estado inicial y f su único estado final.

q0 0 0 D q1	q2 1 0 I q3	q4 0 0 D q5
q0 1 1 D q0	q3 0 0 I q4	q4 1 1 I q4
q1 0 1 I q2	q3 1 1 I q3	q5 0 0 H f
q1 1 1 D q1		q5 1 0 D q0



Computabilidad

Apellidos, Nombre: DNI:

4. (1,75 punto) Rellénense los CUATRO huecos de modo que la MT compute la siguiente función:

$$f(x) = 3x$$

(q0, 1, 0, D, q1)		
(q1, 1, 0, D, q2)	(q3, 1, 1, D, q3)	(q5, 0, 0, I, q6)
	(q3, 0, 1, D, q4)	(q6, 1, 1, I, q6)
(q2, 1, 1, D, q2)	(q4, 0, 1, I, q5)	

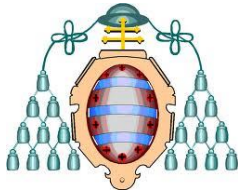
5. Queremos determinar la irresolubilidad del siguiente problema **C**: “Dado un programa while P, determinar si la función unaria asociada a P está definida sí y solo sí su entrada es mayor o igual que 10”. Responde a los siguientes ejercicios:

- a) (0,5 puntos) Construye un posible programa que más adelante nos permita reducir el problema de la parada al problema **C**.

- b) (0,5 puntos) Indica la función unaria semántica del programa construido en el apartado a).

- c) (0,5 puntos, pero una respuesta incorrecta resta 0,2 puntos) ¿Cómo utilizaría el método de reducción el programa del apartado a) para demostrar la irresolubilidad del problema **C**? Marca la única respuesta correcta:

- ☐ Si tuviésemos un algoritmo (macro) **A** que resuelve el problema **C**, aplicándola al número que codifica el programa, dicho algoritmo podría resolver también el problema de la parada, que sabemos que es irresoluble, lo cual nos lleva a una contradicción, y por lo tanto **A** no puede existir.
- ☐ Ejecutando el programa se resolvería directamente el problema de la parada. Como sabemos que el problema de la parada es irresoluble, esto nos lleva a una contradicción y por tanto **C** es irresoluble.



Computabilidad

Apellidos, Nombre: DNI:

- ☐ Si tuviésemos un algoritmo (macro) **A** que resuelve el problema de la parada, aplicándola al número que codifica el programa, dicho algoritmo podría resolver también el problema **C**, lo cual nos lleva a una contradicción.

d) **(0,75 puntos, pero una respuesta incorrecta resta 0,25 puntos)** Queremos demostrar, utilizando el Teorema de Rice, que el problema **C** es irresoluble. Para aplicarlo, necesitamos construir dos programas while **Q1** y **Q2**. Marca la única respuesta que nos permitiría aplicar el Teorema de Rice:

<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Q1: begin while ($X1 \geq 10$) do $X1 := \text{succ}(X1)$; end Q2: begin while ($X1 < 10$) do $X1 := \text{succ}(X1)$; end	Q1: begin while ($X1 < 10$) do $X1 := \text{pred}(X1)$; end Q2: begin $X1 := 20$; end	Q1: begin while ($X1 < 10$) do $X1 := 0$; end Q2: begin $X2 := X1$; while ($X2 \geq 10$) do $X2 := 0$; while ($X1 < 10$) do $X2 := \text{succ}(X2)$; end

6. **(1 punto)** Sea un programa while **P** cuyo código es 'e' y su función semántica ternaria es $\varphi_e(x, y, z) = \varphi_x(z) + 2z^y$. Queremos construir un programa while **P2** cuya función unaria semántica sea igual a $\varphi_a(z) = \varphi_{e1}(z) + 2z^3$, siendo $e1$ una constante. ¿cómo se podría calcular el código 'a' de este programa? ¿se podría calcular para cualquier valor de $e1$?