Computabilidad

Apellidos, Nombre:……………………………………………….DNI:…………

1. (1,75 puntos) Completa las instrucciones que faltan en el siguiente programa while P para que su función binaria semántica sea:

$$\varphi_P^{(2)}(x,y) = x^{x \dot{-} y}$$

Se permite utilizar la macro de la diferencia acotada y de la multiplicación.

Nota: Puede haber más de una instrucción por hueco

```
begin
 X3 := X1 ∸ X2;
 X4 := 0;
 X5 := X1;
 X1 := 1;
 while X3 ≠ X4 do
  begin
   X1 := X1 * X5;
   X3 := pred(X3);
  end
end
```

2. (1,5 puntos) Dado el siguiente programa while P, identifica todas las macros utilizadas en él y construye un programa while equivalente sin macros.

begin
X3 := 1
while X2 ≠ X1 do
begin
X2 := succ(X2)
X3 := X3 + X2
end
X1 := X3
end
	begin
	X3 := 0;
	X3 := succ(X3);
	while X2 ≠ X1 do
	begin
	X2 := succ(X2)
	X4 := 0;
	while X4 ≠ X2 do
	begin
	X3 := succ(X3);
	X4 := succ(X4);
	end;
	end
	X1 := succ(X3);
	X1 := pred(X1);
	end

3. **(1,75 puntos)** Indica las funciones semánticas **unaria** y **binaria** de la siguiente máquina de Turing, siendo q0 su estado inicial y f su único estado final.

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

$$\varphi_M^{(1)}(x) = \perp$$

$$\varphi_M^{(2)}(x,y) = \begin{cases} x/_2 + y & si\ x\ es\ par \\ \perp & si\ x\ es\ impar \end{cases}$$

4. **(1,75 puntos)** Disponemos de una Máquina de Turing M_G con estados {q0, q1, q2, q3}, dónde q0 es su estado inicial, q3 su estado final y cuya función semántica es $\varphi_G(x,y) = g(x,y)$. A partir de esta, crea una Máquina de Turing que calcule la siguiente función unaria especificando quiénes son sus estados inicial y final:

$$f(x) = g(x,0)$$

M=({0,1}, {p_0,p_1, p_2, p_3,q_0,q_1,q_2,q_3}, T∪$\mathbf{T_g}$, p_0, {q_3}), donde T es:

p_0 1 1 D p_0	p_1 0 1 I p_2	p_3 1 1 I p_3
p_0 0 0 D p_1	p_2 0 0 I p_3	p_3 0 0 D q_0

5. Queremos determinar la irresolubilidad del siguiente problema **C**: "Dado un programa while P, determinar si P devuelve un valor impar sí y solo sí su entrada es distinta de 1". Responde a los siguientes apartados:

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

```
begin
  X2 := U(e,k);
  if X1 = 1 then X1:=2
       else X1:=3
end
```

b) **(0,25 puntos)** Indica la función unaria semántica del programa construido en el apartado a).

$$\varphi_P(x) = \begin{cases} 2 & x = 1\ \wedge\ el\ programa\ P_e\ para\ con\ entrada\ k \\ 3 & x \neq 1\ \wedge el\ programa\ P_e\ para\ con\ entrada\ k \\ \perp & \varphi_e(k) = \perp \end{cases}$$

c) **(0,5 puntos)** Indica cómo podríamos utilizar el programa desarrollado en el apartado a) para demostrar que el problema **C** es irresoluble

Solución resumida:

Dado que el programa anterior *P* usa las constantes *e* y *k*, podemos, utilizando el teorema de parametrización, encontrar una función *f* total y computable que nos proporciona un código para esta función semántica, es decir: $\varphi_{f(e,k)}(x) = \varphi_P(x)$.

Si asumimos que el problema C es resoluble, entonces podemos crear una macro C(X) para él, y conjuntamente con la macro F que computa $f(e,k)$, podemos crear el siguiente programa:

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

cuya función binaria semántica es:

$$\varphi(e,k) = C\big(f(e,k)\big) = \begin{cases} 1 & si\ \varphi_{f(e,k)}(x)\ es\ impar\ si\ y\ solo\ si\ x \neq 1 \\ 0 & en\ otro\ caso \end{cases}$$

Pero, recuérdese que el $\varphi_{f(e,k)}(x) = \varphi_P(x)$ es un valor impar (3) si, y sólo si, $x \neq 1$ exactamente cuando el programa P_e para con entrada k , por lo que esta función se puede reescribir como

$$\varphi(e,k) = \begin{cases} 1 & Pe \text{ para con entrada } k \\ 0 & en\ otro\ caso \end{cases}$$

Esta es la función característica del problema de la parada. Es decir, hemos reducido al problema de la parada a nuestro problema, lo que indica que, si pudiésemos resolver nuestro problema (si C fuera computable), podríamos resolver también el problema de la parada. Sin embargo, el problema de la parada es irresoluble, por lo que esto es una contradicción, lo que quiere decir que nuestro problema no puede ser resoluble.

d) **(0,75 puntos)** Aplica el Teorema de Rice para demostrar que el problema **C** es irresoluble.

C se puede reescribir como dado P decidir si o no $\varphi_P(x)$ es un número impar si, y sólo si, $x \neq 1$. Visto así, obviamente es una propiedad semántica.

Veamos que es no trivial:

P_1: begin if X1 = 1 then X1:=2 else X1:=3 end	P_2: begin X1:=0 end
cumple C	**no cumple C**

Como es una propiedad semántica no trivial, por el teorema de Rice, el problema de decidir si un programa cumple esa propiedad es irresoluble.

6. (1,25 punto) Dado el siguiente programa P con función semántica $\varphi(x) = x + \varphi_e(k)$:

```
begin
  X2 := U(e,k)
  X1 := X1 + X2
end
```

Construir un programa a cuya función semántica se le pueda aplicar el teorema de parametrización para demostrar que para cualquier '*e*' y cualquier '*k*', se puede obtener el código '*c*' de un programa cuya función semántica es equivalente a la del programa anterior. Dar la función que nos permite obtener '*c*'.

P: begin

```
  X2 := U(X1,X2)
  X1 := X3 + X2
end
```

Aplicando el teorema de parametrización en P, obtenemos la función $s_1^2(d,e,k)$ que nos permite obtener c, siendo $d = cod(P)$. Como esta función es total, se demuestra que c se puede obtener para cualquier 'e' y cualquier 'k'.