Modelos de Computación y de Complejidad

Grado en Ingeniería Informática. Tecnologías Informáticas (Curso 2021–2022)

ETS Ingeniería Informática - Universidad de Sevilla

CODIFICACIONES - PROGRAMAS UNIVERSALES

Ejercicio 1.– Sea $f : \mathbb{N} \longrightarrow \mathbb{N}$ una función total. Se define la *función historia* de f, así:

$$\widehat{f}(n) = [f(0), \ldots, f(n)]$$

Probar que $\widehat{f}$ es GOTO-computable sí y sólo si f es GOTO–computable.

Indicación: Pruébese que $\widehat{f}$ está definida por recursión primitiva, a partir de funciones GOTO-computables.

Ejercicio 2.– Probar que las siguientes funciones f y g son GOTO–computables:

$$\begin{cases} f(0) & = & 0 \\ f(n+1) & = & 1 + g(n) \end{cases} \qquad \begin{cases} g(0) & = & 0 \\ g(n+1) & = & 2 + f(n) \end{cases}$$

Indicación: Considérese la función auxiliar $\psi : \mathbb{N} \to \mathbb{N}$ definida por $\psi(n) = [f(n), g(n)]$, para cada $n \in \mathbb{N}$, y pruébese que ψ está definida por recursión primitiva, a partir de funciones GOTO-computables. Finalmente, teniendo presente que $f(n) = (\psi(n))_1$ y que $g(n) = (\psi(n))_2$, concluir que las funciones f y g son GOTO-computables.

Ejercicio 3.– La sucesión de **Fibonacci** está definida por la siguiente función total $F : \mathbb{N} \to \mathbb{N}$:

$$F(n) = \begin{cases} n & \text{si } n \leq 1 \\ F(n-1) + F(n-2) & \text{si } n \geq 2 \end{cases}$$

Demostrar que la sucesión de Fibonacci es una función GOTO-computable:

Indicación: Considérese la función auxiliar $\psi : \mathbb{N} \to \mathbb{N}$ definida por $\psi(n) = [F(n), F(n+1)]$, para cada $n \in \mathbb{N}$, y pruébese que ψ está definida por recursión primitiva, a partir de funciones GOTO-computables. Finalmente, teniendo presente que $F(n) = (\psi(n))_1$, concluir que la función F es GOTO-computable.

Ejercicio 4.– Sea p_k el k-ésimo número primo ($k \geq 1$). Determinar para qué valores de k el número natural $4p_k + 1$ es el código de una configuración inicial de un programa GOTO.

Indicación: Téngase presente que en una configuración inicial los valores de cualquier variable que **no** sea de entrada, ha de valer 0.

Ejercicio 5.– Sea $n \geq 1$. Probar que para cada función GOTO–computable, f, de aridad n, existen infinitos índices $e \in \mathbb{N}$ tales que $\varphi_e^{(n)} = f$.

Indicación: Téngase presente que para toda función GOTO-computable, existen infinitos programas GOTO que la calcula.

Ejercicio 6.– Sea P el programa GOTO de código 9215. Se pide:

1. Describir explícitamente el programa P.
2. Determinar si la configuración σ de código 109 es una configuración inicial de P. ¿Es una configuración de parada del programa P?
3. Calcular $\varphi_{9215}(a)$ siendo a el valor de la variable X_1 en la configuración σ.

Ejercicio 7.– Sea P el programa GOTO de código 3.981.311. Se pide:

(a) Describir explícitamente el programa P.

(b) Determinar si la configuración σ de código 757 es una configuración inicial de P. En caso afirmativo, hallar la ejecución de P con dato de entrada dicha configuración.

(c) Hallar la configuración siguiente de σ en dicha ejecución de P.

(d) Hallar el código de una configuración de parada del programa P.

Ejercicio 8.– Sea P el programa GOTO de código $900 \cdot 7^{38} - 1$. Se pide:

(a) Describir explícitamente el programa P.

(b) Determinar si la configuración σ de código 217 es una configuración inicial de P. ¿Es una configuración de parada de P?

Ejercicio 9.– Sea P el programa GOTO de código 153055007. Se pide:

(a) Describir explícitamente el programa P.

(b) Determinar si la configuración σ de código 153055007 es una configuración de parada de P.

(c) Hallar la ejecución del programa P con dato de entrada $(0, 0, 0)$.

Ejercicio 10.– Se considera la computación del siguiente programa con dato de entrada $(2, 1)$:

$$P \equiv \begin{cases} I_1 \equiv & [A] & Z_2 \longleftarrow Z_2 - 1 \\ \ldots & \ldots & \ldots \\ I_6 \equiv & & X \longleftarrow X + 1 \\ I_7 \equiv & & IF\ X_2 \neq 0\ GOTO\ A \\ \ldots & \ldots & \ldots \end{cases}$$

Al cabo de $k-1$ pasos se sabe que la configuración σ_k, es tal que $\#(\sigma_k) = 6623295$. Se pide:
a) Obtener $\#(\sigma_{k+1}), \#(\sigma_{k+2})$ y $\#(\sigma_{k+3})$. b) Obtener el valor almacenado en la variable Y de la configuración σ_{k+3}.

Indicación: Teniendo presente que $6623296 = 64 \cdot 103489$, determinar explícitamente la configuración σ_k, a fin de hallar las configuraciones $\sigma_{k+1}, \sigma_{k+2}$ y σ_{k+3}

Ejercicio 11.– Sea $\mathbf{U_1}$ un programa universal asociado al número natural 1. Hallar la ejecución de dicho programa con dato de entrada $(2, 575)$, y comprobar que el resultado de la computación $\mathbf{U_1}(2, 575)$ coincide con el obtenido al ejecutar el programa GOTO cuyo código es 575, con dato de entrada $X_1 = 2$.

Ejercicio 12.– Sean $\mathbf{U_1}$ y $\mathbf{U_2}$ programas universales asociados a los números naturales 1 y 2, respectivamente. Si e es el código de $\mathbf{U_1}$, hallar la computación $\mathbf{U_2}(1, 168, e)$.

Indicación: Téngase presente que si $e = \#(\mathbf{U_1})$ entonces $\mathbf{U_2}(1, 168, e) = \mathbf{U_1}(1, 168)$.

Ejercicio 13.– Sea $\mathbf{U_2}$ un programa universal asociado al número natural 2. Hallar las computaciones $\mathbf{U_2}(4, 5, e)$ y $\mathbf{U_2}(7, 0, e)$, siendo $e = 2^{105} \cdot 3^2 \cdot 5^{110} - 1$.

Ejercicio 14.– Probar que los siguientes conjuntos son GOTO–computables:

- $A = \{x \in \mathbb{N}:$ el programa GOTO de código x carece de instrucciones del tipo CONDICIONAL$\}$.
- $B = \{x \in \mathbb{N}:$ el programa GOTO de código $x + 8$ posee, a lo sumo, 8 instrucciones$\}$.
- $C = \{x \in \mathbb{N}:$ el programa GOTO de código x posee 10 instrucciones y ninguna de ellas es SKIP$\}$.
- $D = \{x \in \mathbb{N}:$ el programa GOTO de código $2x + 3$ posee, a lo sumo, 7 instrucciones y alguna de ellas es un DECREMENTO$\}$.
- $E = \{x \in \mathbb{N}:$ el programa GOTO de código $7x+7$ posee un número par de instrucciones y todas ellas son de tipo CONDICIONAL $\}$
- $F = \{x \in \mathbb{N}:$ x es el código de un programa GOTO tal que todas sus variables son de entrada$\}$
- $G = \{x \in \mathbb{N}:$ el programa GOTO de código x posee alguna instrucción etiquetada a la izquierda$\}$.
- $H = \{x \in \mathbb{N}:$ el programa GOTO de código x^2 **para** sobre $x + 4$ en, **exactamente**, x pasos $\}$.
- $I = \{x \in \mathbb{N}:$ el programa GOTO de código $3x$ para sobre $4x$ en, **exactamente**, $5x$ pasos$\}$.

Ejercicio 15.– Justificar la veracidad o falsedad de las siguientes afirmaciones.

1. Dos funciones GOTO–computables pueden tener asociadas el **mismo** índice.
2. El conjunto $A = \{\#(P) : P$ es un programa GOTO cuya última instrucción es $Y \longleftarrow Y + 1\}$ es GOTO–computable.
3. La configuración cuyo código sea 5293 es una configuración inicial
4. El código de la configuración de parada de un programa GOTO **nunca** puede ser un número **impar**.
5. La configuración de código 31755 es una **configuración inicial** del programa GOTO de código 3968 y el resultado de la ejecución del programa con esa configuración inicial es 1.
6. Una configuración $\sigma = (j, s)$ de un programa $P = (I_1, \ldots, I_r)$ es **inicial** si el valor de j es 1; es decir, si la primera instrucción a ejecutar es I_1. Además, la configuración σ es de **parada** si $j = r + 1$.
7. Los códigos de las configuraciones iniciales y las configuraciones de parada de un programa GOTO siempre son números impares.
8. Sean P y Q programas GOTO cuyos códigos son m y n, respectivamente. Si $m+1$ y $n+1$ poseen el mismo número de primos (distintos) en su descomposición factorial, entonces dichos programas pueden tener **distinta longitud**.
9. Un programa universal es capaz de reproducir el comportamiento de cualquier programa GOTO (es decir, **todas** las computaciones de cualquier programa P).
10. El programa universal $\mathtt{U}_2$ tiene la capacidad de simular **todas** las computaciones de **cualquier** programa GOTO con dato de entrada un par ordenado (a_1, a_2) **arbitrario**.
11. Si e_n es el código de un programa universal $\mathbf{U_n}$ de orden n, con $n \geq 1$, entonces $\mathbf{U_n}(\underbrace{e_n, \ldots, e_n}_{k}) = 0$, para cada número natural $k \geq 1$.
12. Si e_n, e_p son los códigos de los programas universales $\mathbf{U_n}$ y $\mathbf{U_p}$, con $n, p \geq 1$, respectivamente, entonces $\mathbf{U_n}(\underbrace{e_p, \ldots, e_p}_{k}) = 0$, para cada número natural $k \geq 1$.

13. Es GOTO-computable la función f se aridad 3 sobre $\mathbb{N}$ definida como sigue: para cada $a, b, c \in \mathbb{N}$, $f(a, b, c)$ es el resultado de la computación del programa de código a con dato de entrada $X_1 = b$ y $X_2 = c$.

14. Si $\mathbf{U_2}$ es un programa universal (correspondiente al número natural 2), entonces $\mathbf{U_2}(1, 2, 3) = 4$.

15. Sean e, e' números naturales tales que $\varphi_e^{(1)} = \varphi_{e'}^{(1)}$. Entonces se tiene que $\varphi_e^{(2)} = \varphi_{e'}^{(2)}$.

Ejercicio 16.–

1. ¿Qué se puede asegurar de un **estado** cuyo **código** coincide con el código de una **variable de entrada**?

2. ¿Qué se puede asegurar de un **estado** cuyo **código** coincide con el código de una **variable** que **no** es de **entrada**?

3. ¿Qué se puede asegurar de una **configuración** cuyo **código** coincide con el código de una **variable de entrada**?

4. ¿Qué se puede asegurar de una **instrucción** GOTO cuyo **código** coincide con el código de una **variable de entrada**?

5. ¿Qué se puede asegurar de un **programa** GOTO cuyo **código** coincide con el código de una **variable de entrada**?

6. Qué se puede asegurar de un **programa** GOTO cuyo **código** coincide con el código de una **instrucción** GOTO?

7. **¿Qué se puede asegurar de un programa GOTO cuyo código es el anterior del código de un estado inicial?**

8. ¿Qué se puede asegurar de una **instrucción** GOTO cuyo **código** coincide con el código de una **configuración inicial**?

9. ¿Qué se puede asegurar de una **configuración** cuyo **código** es un **número primo**?

10. ¿Qué se puede asegurar de un **programa** GOTO cuyo **código** es un **número primo**?

11. ¿Qué se puede asegurar de un **programa** GOTO cuyo **código** es el anterior a un **número primo**?

12. ¿Qué se puede asegurar de un **programa** GOTO cuyo **código** es el siguiente a un **número primo**?

13. ¿Qué se puede asegurar de un **programa** GOTO cuyo **código** coincide con el **código** de una **configuración** del programa?

14. ¿Qué se puede asegurar de la **función de aridad** n cuyo **código** que calcula el programa universal $\mathbf{U_n}$?

15. ¿Qué se puede asegurar de la computación $\mathbf{U_3}(e_1, e_2, e_1, e_2)$, siendo $e_i = \#(\mathbf{U_i})$, $1 \leq i \leq 2$?

16. ¿Es GOTO–computable la función binaria f sobre $\mathbb{N}$ definida por $f(a, b) = P_a(b)$, siendo $P_a(b)$ la computación del programa de código a con dato de entrada $X_1 = b$?