

Modelos de Computación y Complejidad

(4º curso del Grado en Ingeniería Informática. Tecnologías Informáticas)

EXAMEN EVALUACIÓN ALTERNATIVA: Junio de 2020

Apellidos..... Nombre.....

Primera parte: Preguntas cortas (2'5 puntos)

1. ¿Existe algún programa GOTO “singular”, relacionado con un conjunto recursivamente enumerable?
2. Dado un programa GOTO, ¿cuántas **funciones distintas** se puede calcular con ese programa?
3. ¿Podría indicar cómo se ejecuta una instrucción condicional IF $V \neq 0$ GOTO L a una configuración $\sigma = (j, s)$ de un programa GOTO.
4. ¿Qué es el **código** de una **instrucción** GOTO?
5. ¿Porqué a un programa GOTO se le exige que la última instrucción no pueda ser del tipo $Y \leftarrow Y$?
¿Podría, en cambio, ser $[A] Y \leftarrow Y$ la última instrucción?
6. ¿Qué podría indicar acerca del comportamiento del **programa universal** U_1 ?
7. ¿Qué significa que cualquier modelo de computación tenga **limitaciones**?
8. ¿Podría citar alguna diferencia importante entre una **MTD** y una **MTND**?

Segunda parte: Justificar la veracidad o falsedad de las siguientes afirmaciones. (3'5 puntos)

(Responder sólo a cuatro asertos)

1. La función vacía y la función idénticamente nula se pueden calcular mediante programas GOTO que contengan únicamente instrucciones del tipo **incremento** y del tipo **condicional**.
2. Si s es el estado de código 100, entonces $\sigma = (6, s)$, es una **configuración de parada** del programa GOTO de código 100.
3. Si U_1 es un programa universal (correspondiente al número natural 1), entonces se verifica que $U_1(1, 2) = 3$.
4. La cuantificación universal de un predicado GOTO-computable **no** tiene por qué ser un predicado GOTO-computable.
5. El teorema del complemento proporciona un método para establecer que un conjunto de números naturales **no es recursivamente enumerable**.
6. Si X_1, X_2 son problemas de decisión tales que X_1 es *reducible en tiempo polinomial* al problema X_2 y, además, X_1 pertenece a la clase **P**, entonces el problema X_2 también pertenece a la clase **P**.

Tercera parte: Ejercicios (4 puntos)

(Resolver sólo dos ejercicios)

1. Diseñar un programa GOTO, sin usar macros (salvo, quizás, GOTO L), que calcule la función parcial $f : \mathbb{N} \rightarrow \mathbb{N}$ definida como sigue:

$$f(x) = \begin{cases} x + 2 & \text{si } x \text{ es un número impar} \\ \uparrow & \text{si } x \text{ es un número par} \end{cases}$$

Comprobar que el programa diseñado funciona correctamente para algunos valores concretos.

2. Dado el programa GOTO, P , de código $153055007 = 2^5 \cdot 3^{14} - 1$, se pide: (a) describir explícitamente P ; y (b) determinar si la configuración de código 1607 es una configuración de parada de P .
3. Consideremos el siguiente problema de decisión:

“Dado un número natural n , determinar si la función $\varphi_n^{(1)}$ **no es inyectiva**”

Utilizando el teorema de Rice, demostrar que dicho problema **es indecidible**. Demostrar que, en cambio, dicho problema **sí es semidecidible**.

Modelos de Computación y Complejidad

(4º curso del Grado en Ingeniería Informática. Tecnologías Informáticas)

EXAMEN EVALUACIÓN ALTERNATIVA: Junio de 2020

Apellidos..... Nombre.....

Primera parte: Preguntas cortas (2'5 puntos)

1. ¿Existe alguna relación entre conjuntos GOTO-computables y conjuntos recursivamente enumerables?
2. El **programa vacío** no contiene ninguna instrucción pero ¿calcula alguna función? ¿Cuántas funciones distintas puede calcular?
3. ¿Qué es un **índice** de una función GOTO-computable?
4. ¿Podría indicar explícitamente cuál sería la configuración inicial de un programa GOTO asociada al dato de entrada (5, 1, 0, 2)?
5. ¿Podría describir cómo se puede utilizar el teorema de definición por casos para funciones totales, a fin de demostrar la GOTO-computabilidad de una función?
6. El programa universal **U₁** ¿puede **simular** el comportamiento del programa universal **U₂**?
7. ¿Qué es la **potencia computacional** de un **modelo de computación**?
8. ¿En qué consiste el *problema P versus NP*? ¿Qué es un problema NP-completo?

Segunda parte: Justificar la veracidad o falsedad de las siguientes afirmaciones. (3'5 puntos)

(Responder sólo a cuatro asertos))

1. Un programa GOTO calcula funciones de distinta aridad, pero **no puede calcular** dos funciones distintas y con la **misma aridad**.
2. La función factorial en \mathbb{N} , definida por $f(0) = 1$ y $f(n + 1) = (n + 1) \cdot f(n)$, para cada $n \in \mathbb{N}$, es GOTO-computable.
3. El código de un **estado** correspondiente a cualquier configuración inicial de un programa GOTO, **siempre** ha de ser un **número impar**.
4. Si **U₂** es un programa universal (correspondiente al número natural 2), entonces el resultado de la computación **U₂**(1, 2, 3, 4, 5, 6) coincide con el de la computación del programa de código 3 con dato de entrada (1, 2).
5. La cuantificación existencial y la cuantificación universal de un predicado parcialmente decidible **no tienen** por qué ser un predicado parcialmente decidible.
6. El teorema de Rice proporciona un método para establecer que un conjunto de números naturales **no es GOTO-computable**.

Tercera parte: Ejercicios (4 puntos)

(Resolver sólo dos ejercicios))

1. Dado el programa GOTO, P , de código $153055007 = 2^5 \cdot 3^{14} - 1$, se pide: (a) describir explícitamente P ; (b) determinar si la configuración de código 803 es una configuración de parada de P
2. Probar que el conjunto $A = \{x \in \mathbb{N} : \text{el programa GOTO de código } x + 5 \text{ posee, a lo sumo, } 5 \text{ instrucciones}\}$ es GOTO-computable y que, en cambio, el conjunto $B = \{\langle x, y \rangle \in \mathbb{N} : \text{el programa de código } x \text{ para sobre } y \text{ en, exactamente, un número impar de pasos y el resultado es un número par}\}$ es recursivamente enumerable.
3. Diseñar una MTD cuyos datos de entrada sean números naturales expresados en binario y de tal manera que: (a) si el dato de entrada es **par**, entonces **para y devuelve 1**; y (b) si es **impar** entonces **no para**. Ilustrar el diseño realizado, considerando **dos** datos de entrada de tamaño 3 y detallando las correspondientes computaciones.