

Modelos de Computación y Complejidad

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

Algunos ejemplos de preguntas cortas para el EXAMEN ONLINE INDIVIDUAL

PARTE PRIMERA : Preguntas cortas y directas

1. ¿Qué se puede asegurar de un programa GOTO en el que **no aparece ninguna variable de entrada**?
2. ¿Qué se puede asegurar de un programa GOTO en el que **sólo aparecen variables de entrada**?
3. ¿Qué se puede asegurar de un programa GOTO en el que **no aparece la variable de salida**?
4. ¿Qué se puede asegurar de un programa GOTO en el que **sólo aparecen variables auxiliares o de trabajo**?
5. ¿Qué se puede asegurar de un programa GOTO en el que **todas las instrucciones son del tipo condicional**?
6. ¿Qué se puede asegurar de un programa GOTO en el que **ninguna** de sus instrucciones está etiquetada a la izquierda?
7. ¿Qué se puede asegurar de un programa GOTO en el que **todas las instrucciones son del mismo tipo**?
8. ¿Qué se puede asegurar de un programa GOTO que consta de **una única instrucción**?
9. ¿Cuántas **funciones distintas** se pueden calcular con **un** programa GOTO?
10. ¿Puede existir **un** programa GOTO que calcule **dos** funciones **distintas** de la misma aridad?
11. ¿Pueden existir **dos** programas GOTO distintos que calculen una misma función?
12. ¿Pueden existir **dos** programas GOTO **distintos** que calculen las **mismas funciones**, para cualquier aridad?
13. ¿Puede calcular un programa GOTO la función suma y, también, la función identidad en \mathbb{N} ?
14. ¿Puede calcular un programa GOTO la función producto y, también, la función idénticamente nula en \mathbb{N} de aridad 1?
15. ¿Qué es una **macro GOTO**? ¿Podría citar algún ejemplo sencillo?
16. ¿Qué es un **índice** de una función GOTO-computable?
17. ¿Qué significa que un **conjunto de tuplas** (de la misma aridad) **de números naturales** sea GOTO-computable?
18. ¿Podría indicar algún método para determinar que un **conjunto no es** GOTO-computable?
19. ¿Qué es un **predicado GOTO-computable**? ¿Podría indicar algún método para determinar que un **predicado no es** GOTO-computable?
20. ¿Qué es un **predicado parcialmente decidable**? ¿Podría indicar algún método para determinar que un **predicado es parcialmente decidable**?

21. ¿Podría indicar algún método para determinar que un **predicado no es parcialmente decidable**?
22. ¿Existe algún programa GOTO “singular”, relacionado con un conjunto recursivamente enumerable?
23. ¿Podría indicar alguna caracterización de conjuntos recursivamente enumerables?
24. ¿Podría indicar algún método para determinar que un **conjunto** es recursivamente enumerable?
25. ¿Podría indicar algún método para determinar que un **conjunto no es** recursivamente enumerable?
26. ¿Existe alguna relación entre conjuntos GOTO-computables y conjuntos recursivamente enumerables?
27. ¿Cómo se define la función **característica** de un conjunto de tuplas de números naturales (de la misma aridad)? ¿Y la función **característica parcial**? ¿Qué propiedades interesantes verifican esas funciones?
28. ¿Qué es un **estado** de un programa GOTO? ¿Cómo se define el **código** de un **estado** de un programa GOTO?
29. ¿Qué es una **configuración** de un programa GOTO? ¿Qué es una **computación** de un programa GOTO?
30. ¿Qué es una **configuración inicial**? ¿Y una **configuración de parada** de un programa GOTO?
31. ¿Podría indicar explícitamente cuál sería la configuración inicial de un programa GOTO asociada al dato de entrada $(5, 1, 0, 2)$?
32. ¿Podría indicar cómo se ejecuta una instrucción **condicional** $\text{IF } V \neq 0 \text{ GOTO } L$ a una configuración $\sigma = (j, s)$ de un programa GOTO?
33. ¿Cómo se define la función de Gödel? ¿Cómo se utiliza dicha función para la codificación en el modelo de computación GOTO?
34. ¿Cómo se define la función par? ¿Cómo se pueden codificar las ternas ordenadas a partir de la función par?
35. ¿Qué es un **estado** de un programa GOTO? ¿Cómo se define el **código** de un **estado** de un programa GOTO? ¿Qué información relevante proporciona?
36. ¿Qué se puede asegurar de un **programa GOTO** cuyo **código** es un **número par**?
37. ¿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 de un programa GOTO?
38. ¿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?
39. ¿Podría describir el enunciado del **problema de la parada**? ¿Es resoluble mecánicamente dicho problema?
40. ¿Podría describir el método que proporciona el **teorema de Rice** para establecer la **no** GOTO-computabilidad de un conjunto de números naturales?

41. ¿Qué afirma el teorema del complemento? ¿Para qué se puede utilizar?
42. ¿Podría describir algún método para establecer la **indecidibilidad** de un problema de decisión?
43. ¿Podría describir algún método para establecer la **no semidecidibilidad** de un problema de decisión?
44. ¿Podría describir cómo se comportan los predicados GOTO-computables con la cuantificación universal y existencial (tanto si es acotada como si no lo es)?
45. ¿Podría describir cómo se comportan los predicados parcialmente decidibles con la cuantificación universal y existencial (tanto si es acotada como si no lo es)?
46. ¿Qué se puede asegurar de la **función de aridad n** que calcula el programa universal **U_n** ?
47. ¿Qué es la **potencia computacional** de un modelo de computación? ¿Qué significa que cualquier modelo de computación tenga **limitaciones**?
48. ¿Existe alguna relación entre **modelos de computación universales** y programas GOTO universales?
49. ¿Porqué la negación del predicado $\text{HALT}(x, y) \equiv \varphi_x^{(1)}(y) \downarrow$ **no es** parcialmente decidable?
50. ¿Podría citar alguna diferencia importante entre una **MTD** y una **MTND**?
51. ¿Qué significa que una MTD **decida un lenguaje**? ¿Qué significa que una MTD **resuelva un problema de decisión**?
52. ¿Qué es una medida de complejidad? ¿Qué propiedades interesante verifica una medida de complejidad?
53. ¿Cómo se define el tiempo de ejecución de una MTD? ¿Proporciona alguna información interesante?
54. ¿Qué significa que una MTND **decida un lenguaje**? ¿Qué significa que una MTND **resuelva un problema de decisión**?
55. ¿Podría describir un método para establecer que un problema de decisión pertenece a la clase de complejidad **NP**?
56. ¿Podría citar algunas propiedades que satisface las reducibilidades en tiempo polinomial?
57. ¿Qué quiere decir que una clase de complejidad es **estable** bajo reducibilidad en tiempo polinomial?
58. ¿Qué diferencia existe entre problemas **decidibles** y problemas **tratables**?
59. ¿Porqué suele afirmarse que los problemas de la clase de complejidad **NP** son problemas **tratables** en **modo no determinista**?
60. ¿Cómo se pueden obtener nuevos problemas **NP-completos** a partir de otros problemas cuya **NP-completitud** haya sido demostrada previamente?