

Modelos de Computación y Complejidad

Grado en Ingeniería Informática. Tecnologías Informáticas

EXAMEN EVALUACIÓN ALTERNATIVA: Junio de 2022

Mario de J. Pérez Jiménez
David Orellana Martín

Dpto. Ciencias de la Computación e Inteligencia Artificial
E.T.S. Ingeniería Informática
Universidad de Sevilla

marper@us.es

<http://www.cs.us.es/~marper/>

Curso 2021-2022



EXAMEN EVALUACIÓN ALTERNATIVA: Duración aproximada: 1 hora

EXAMEN EVALUACIÓN ALTERNATIVA: Duración aproximada: 1 hora

PRIMERA PARTE: Preguntas cortas

EXAMEN EVALUACIÓN ALTERNATIVA: Duración aproximada: 1 hora

PRIMERA PARTE: Preguntas cortas

- * Duración aproximada: 10 minutos (Respuestas directas).

PRIMERA PARTE: Preguntas cortas

- * Duración aproximada: 10 minutos (Respuestas directas).
- * Se realizarán ocho preguntas cortas.

EXAMEN EVALUACIÓN ALTERNATIVA: Duración aproximada: 1 hora

PRIMERA PARTE: Preguntas cortas

- * Duración aproximada: 10 minutos (Respuestas directas).
- * Se realizarán ocho preguntas cortas.

SEGUNDA PARTE: Justificar la veracidad o falsedad de ciertas afirmaciones

EXAMEN EVALUACIÓN ALTERNATIVA: Duración aproximada: 1 hora

PRIMERA PARTE: Preguntas cortas

- * Duración aproximada: 10 minutos (Respuestas directas).
- * Se realizarán ocho preguntas cortas.

SEGUNDA PARTE: Justificar la veracidad o falsedad de ciertas afirmaciones

- * Duración aproximada: 20 minutos

PRIMERA PARTE: Preguntas cortas

- * Duración aproximada: 10 minutos (Respuestas directas).
- * Se realizarán ocho preguntas cortas.

SEGUNDA PARTE: Justificar la veracidad o falsedad de ciertas afirmaciones

- * Duración aproximada: 20 minutos
- * Al comenzar esta parte, se enviará por correo, un fichero PDF con las afirmaciones propuestas, y se proporcionará un tiempo prudencial, antes de responder.

PRIMERA PARTE: Preguntas cortas

- * Duración aproximada: 10 minutos (Respuestas directas).
- * Se realizarán ocho preguntas cortas.

SEGUNDA PARTE: Justificar la veracidad o falsedad de ciertas afirmaciones

- * Duración aproximada: 20 minutos
- * Al comenzar esta parte, se enviará por correo, un fichero PDF con las afirmaciones propuestas, y se proporcionará un tiempo prudencial, antes de responder.
- * Se propondrán seis afirmaciones, de las cuales el alumno ha de responder **sólo cuatro** de ellas.

PRIMERA PARTE: Preguntas cortas

- * Duración aproximada: 10 minutos (Respuestas directas).
- * Se realizarán ocho preguntas cortas.

SEGUNDA PARTE: Justificar la veracidad o falsedad de ciertas afirmaciones

- * Duración aproximada: 20 minutos
- * Al comenzar esta parte, se enviará por correo, un fichero PDF con las afirmaciones propuestas, y se proporcionará un tiempo prudencial, antes de responder.
- * Se propondrán seis afirmaciones, de las cuales el alumno ha de responder **sólo cuatro** de ellas.

TERCERA PARTE: Ejercicios

PRIMERA PARTE: Preguntas cortas

- * Duración aproximada: 10 minutos (Respuestas directas).
- * Se realizarán ocho preguntas cortas.

SEGUNDA PARTE: Justificar la veracidad o falsedad de ciertas afirmaciones

- * Duración aproximada: 20 minutos
- * Al comenzar esta parte, se enviará por correo, un fichero PDF con las afirmaciones propuestas, y se proporcionará un tiempo prudencial, antes de responder.
- * Se propondrán seis afirmaciones, de las cuales el alumno ha de responder **sólo cuatro** de ellas.

TERCERA PARTE: Ejercicios

- * Duración aproximada: 30 minutos

PRIMERA PARTE: Preguntas cortas

- * Duración aproximada: 10 minutos (Respuestas directas).
- * Se realizarán ocho preguntas cortas.

SEGUNDA PARTE: Justificar la veracidad o falsedad de ciertas afirmaciones

- * Duración aproximada: 20 minutos
- * Al comenzar esta parte, se enviará por correo, un fichero PDF con las afirmaciones propuestas, y se proporcionará un tiempo prudencial, antes de responder.
- * Se propondrán seis afirmaciones, de las cuales el alumno ha de responder **sólo cuatro** de ellas.

TERCERA PARTE: Ejercicios

- * Duración aproximada: 30 minutos
- * Al comenzar esta parte, se enviará por correo, un fichero PDF con los ejercicios propuestos y se proporcionará un tiempo prudencial, antes de responder.

PRIMERA PARTE: Preguntas cortas

- * Duración aproximada: 10 minutos (Respuestas directas).
- * Se realizarán ocho preguntas cortas.

SEGUNDA PARTE: Justificar la veracidad o falsedad de ciertas afirmaciones

- * Duración aproximada: 20 minutos
- * Al comenzar esta parte, se enviará por correo, un fichero PDF con las afirmaciones propuestas, y se proporcionará un tiempo prudencial, antes de responder.
- * Se propondrán seis afirmaciones, de las cuales el alumno ha de responder **sólo cuatro** de ellas.

TERCERA PARTE: Ejercicios

- * Duración aproximada: 30 minutos
- * Al comenzar esta parte, se enviará por correo, un fichero PDF con los ejercicios propuestos y se proporcionará un tiempo prudencial, antes de responder.
- * El alumno escribirá las soluciones que habrán de ser enviadas (en fichero JPG, PNG o PDF) por correo, al final de la clase, **antes de cerrar la conexión**.

PRIMERA PARTE: Preguntas cortas

- * Duración aproximada: 10 minutos (Respuestas directas).
- * Se realizarán ocho preguntas cortas.

SEGUNDA PARTE: Justificar la veracidad o falsedad de ciertas afirmaciones

- * Duración aproximada: 20 minutos
- * Al comenzar esta parte, se enviará por correo, un fichero PDF con las afirmaciones propuestas, y se proporcionará un tiempo prudencial, antes de responder.
- * Se propondrán seis afirmaciones, de las cuales el alumno ha de responder **sólo cuatro** de ellas.

TERCERA PARTE: Ejercicios

- * Duración aproximada: 30 minutos
- * Al comenzar esta parte, se enviará por correo, un fichero PDF con los ejercicios propuestos y se proporcionará un tiempo prudencial, antes de responder.
- * El alumno escribirá las soluciones que habrán de ser enviadas (en fichero JPG, PNG o PDF) por correo, al final de la clase, **antes de cerrar la conexión**.
- * Se propondrán tres ejercicios, de los cuales el alumno ha de resolver **sólo dos** de ellos.

Un ensayo de EXAMEN ONLINE

PRIMERA PARTE: Preguntas cortas



Un ensayo de EXAMEN ONLINE

PRIMERA PARTE: Preguntas cortas

1. ¿Existe algún programa GOTO singular, relacionado con un conjunto **recursivamente enumerable**?



Un ensayo de EXAMEN ONLINE

PRIMERA PARTE: Preguntas cortas

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?



Un ensayo de EXAMEN ONLINE

PRIMERA PARTE: Preguntas cortas

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.



Un ensayo de EXAMEN ONLINE

PRIMERA PARTE: Preguntas cortas

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 $\text{IF } V \neq 0 \text{ GOTO } L$ a una configuración $\sigma = (j, s)$ de un programa GOTO.
4. ¿Qué es el **código** de una **instrucción** GOTO?

Un ensayo de EXAMEN ONLINE

PRIMERA PARTE: Preguntas cortas

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?

Un ensayo de EXAMEN ONLINE

PRIMERA PARTE: Preguntas cortas

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 $\text{IF } V \neq 0 \text{ GOTO } L$ 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 ?

Un ensayo de EXAMEN ONLINE

PRIMERA PARTE: Preguntas cortas

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 $\text{IF } V \neq 0 \text{ GOTO } L$ 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**?

Un ensayo de EXAMEN ONLINE

PRIMERA PARTE: Preguntas cortas

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 $IF V \neq 0 \text{ GOTO } L$ 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**?



Un ensayo de EXAMEN ONLINE

SEGUNDA PARTE: Justificar la veracidad o falsedad de las siguientes afirmaciones
(responder sólo cuatro de ellas)



Un ensayo de EXAMEN ONLINE

SEGUNDA PARTE: Justificar la veracidad o falsedad de las siguientes afirmaciones
(responder sólo cuatro de ellas)

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**.



Un ensayo de EXAMEN ONLINE

SEGUNDA PARTE: Justificar la veracidad o falsedad de las siguientes afirmaciones
(responder sólo cuatro de ellas)

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.



Un ensayo de EXAMEN ONLINE

SEGUNDA PARTE: Justificar la veracidad o falsedad de las siguientes afirmaciones
(responder sólo cuatro de ellas)

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 $\llbracket U_1 \rrbracket^{(2)}(1, 2) = 3$.



Un ensayo de EXAMEN ONLINE

SEGUNDA PARTE: Justificar la veracidad o falsedad de las siguientes afirmaciones
(responder sólo cuatro de ellas)

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 $\llbracket U_1 \rrbracket^{(2)}(1, 2) = 3$.
4. La cuantificación universal de un predicado GOTO-computable **no** tiene por qué ser un predicado GOTO-computable.



Un ensayo de EXAMEN ONLINE

SEGUNDA PARTE: Justificar la veracidad o falsedad de las siguientes afirmaciones
(responder sólo cuatro de ellas)

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 $\llbracket U_1 \rrbracket^{(2)}(1, 2) = 3$.
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**.

Un ensayo de EXAMEN ONLINE

SEGUNDA PARTE: Justificar la veracidad o falsedad de las siguientes afirmaciones
(responder sólo cuatro de ellas)

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 $\llbracket U_1 \rrbracket^{(2)}(1, 2) = 3$.
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* a X_2 y, además, $X_1 \in \mathbf{P}$, entonces $X_2 \in \mathbf{P}$.



Un ensayo de EXAMEN ONLINE

TERCERA PARTE: Ejercicios (resolver sólo dos de ellos)

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.

Un ensayo de EXAMEN ONLINE

TERCERA PARTE: Ejercicios (resolver sólo dos de ellos)

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 ; (b) determinar si la configuración de código 803 es una configuración de parada de P .

Un ensayo de EXAMEN ONLINE

TERCERA PARTE: Ejercicios (resolver sólo dos de ellos)

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 ; (b) determinar si la configuración de código 803 es una configuración de parada de P .
3. Consideremos el siguiente problema de decisión:

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

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



Otro ensayo de EXAMEN ONLINE

PRIMERA PARTE: Preguntas cortas



Otro ensayo de EXAMEN ONLINE

PRIMERA PARTE: Preguntas cortas

1. ¿Existe alguna relación entre conjuntos GOTO-computables y conjuntos recursivamente enumerables?

Otro ensayo de EXAMEN ONLINE

PRIMERA PARTE: Preguntas cortas

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?

Otro ensayo de EXAMEN ONLINE

PRIMERA PARTE: Preguntas cortas

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?

Otro ensayo de EXAMEN ONLINE

PRIMERA PARTE: Preguntas cortas

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)?

Otro ensayo de EXAMEN ONLINE

PRIMERA PARTE: Preguntas cortas

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?

Otro ensayo de EXAMEN ONLINE

PRIMERA PARTE: Preguntas cortas

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 U_1 ¿puede **simular** el comportamiento del programa universal U_2 ?



Otro ensayo de EXAMEN ONLINE

PRIMERA PARTE: Preguntas cortas

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 U_1 ¿puede **simular** el comportamiento del programa universal U_2 ?
7. ¿Qué es la **potencia computacional** de un **modelo de computación**?

Otro ensayo de EXAMEN ONLINE

PRIMERA PARTE: Preguntas cortas

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 U_1 ¿puede **simular** el comportamiento del programa universal U_2 ?
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**?



Otro ensayo de EXAMEN ONLINE

SEGUNDA PARTE: Justificar la veracidad o falsedad de las siguientes afirmaciones
(responder sólo cuatro de ellas)



Otro ensayo de EXAMEN ONLINE

SEGUNDA PARTE: Justificar la veracidad o falsedad de las siguientes afirmaciones
(responder sólo cuatro de ellas)

1. Un programa GOTO calcula funciones de distinta aridad, pero **no puede calcular** dos funciones distintas y con la **misma aridad**.

Otro ensayo de EXAMEN ONLINE

SEGUNDA PARTE: Justificar la veracidad o falsedad de las siguientes afirmaciones
(responder sólo cuatro de ellas)

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 definida por $f(0) = 1$ y $f(n + 1) = (n + 1) \cdot f(n)$, para cada $n \in \mathbb{N}$, es GOTO-computable.

Otro ensayo de EXAMEN ONLINE

SEGUNDA PARTE: Justificar la veracidad o falsedad de las siguientes afirmaciones
(responder sólo cuatro de ellas)

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 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**.

Otro ensayo de EXAMEN ONLINE

SEGUNDA PARTE: Justificar la veracidad o falsedad de las siguientes afirmaciones (responder sólo cuatro de ellas)

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 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 U_2 es un programa universal (correspondiente al número natural 2), entonces el resultado de la computación $U_2(1, 2, 3, 4, 5, 6)$ coincide con el de la computación del programa de código 3 con dato de entrada (1, 2).

Otro ensayo de EXAMEN ONLINE

SEGUNDA PARTE: Justificar la veracidad o falsedad de las siguientes afirmaciones
(responder sólo cuatro de ellas)

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 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 U_2 es un programa universal (correspondiente al número natural 2), entonces el resultado de la computación $U_2(1, 2, 3, 4, 5, 6)$ 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 decidable **no tienen** por qué ser un predicado parcialmente decidable.



Otro ensayo de EXAMEN ONLINE

SEGUNDA PARTE: Justificar la veracidad o falsedad de las siguientes afirmaciones
(responder sólo cuatro de ellas)

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 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 U_2 es un programa universal (correspondiente al número natural 2), entonces el resultado de la computación $U_2(1, 2, 3, 4, 5, 6)$ 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 decidable **no tienen** por qué ser un predicado parcialmente decidable.
6. El teorema de Rice proporciona un método para establecer que un conjunto de números naturales **no es GOTO-computable**.



Otro ensayo de EXAMEN ONLINE

TERCERA PARTE: Ejercicios (Resolver sólo dos de ellos)



Otro ensayo de EXAMEN ONLINE

TERCERA PARTE: Ejercicios (Resolver sólo dos de ellos)

1. Dado el programa GOTO, P , de código 153055007, se pide: (a) describir explícitamente P ; (b) determinar si la configuración de código 803 es una configuración de parada de P



Otro ensayo de EXAMEN ONLINE

TERCERA PARTE: Ejercicios (Resolver sólo dos de ellos)

1. Dado el programa GOTO, P , de código 153055007, 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 $A = \{x \in \mathbb{N} : \text{el programa GOTO de código } x + 5 \text{ posee, a lo sumo, 5 instrucciones}\}$ 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.

Otro ensayo de EXAMEN ONLINE

TERCERA PARTE: Ejercicios (Resolver sólo dos de ellos)

1. Dado el programa GOTO, P , de código 153055007, 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 $A = \{x \in \mathbb{N} : \text{el programa GOTO de código } x + 5 \text{ posee, a lo sumo, 5 instrucciones}\}$ 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.