



Universidad Nacional Autónoma
de México
Facultad de Ingeniería



Estructuras Discretas

Profesor: Guillermo Anaya

Proyecto Final Cuestionario

Grupo 6

Equipo 6

Integrantes:

Alva Salazar José Luis
Gutierrez Contreras Aldo Aaron
Osnaya Martínez Emmanuel

Semestre 2024-1



Cuestionario:

1. ¿Qué es una función parcial en el contexto de la teoría de la computabilidad?

Una función parcial en la teoría de la computabilidad es una función cuyo dominio no abarca todos los posibles valores de entrada. En otras palabras, hay ciertos valores de entrada para los cuales la función no está definida.

2. ¿Cuáles son las características principales de las funciones parciales en la teoría de la computabilidad?

- **Dominio Restringido:** Las funciones parciales tienen un dominio que no cubre todos los posibles valores de entrada.
- **Valores Indefinidos:** Pueden tener valores no definidos para ciertos elementos del dominio.
- **Posibilidad de Divergencia:** En el caso de funciones computacionales, algunas funciones parciales pueden divergir (no terminar) para ciertos valores de entrada.

3. ¿Cómo se representan las funciones parciales en la teoría de la computabilidad?

Las funciones parciales en la teoría de la computabilidad se pueden representar de varias maneras, siendo las más comunes:

- **Gráficamente:** Mediante gráficos que muestran los pares ordenados (entrada, salida) para las partes definidas de la función.
- **Tablas de Valores:** Una tabla que enumera los valores de entrada y sus correspondientes salidas definidas.
- **Expresiones Algorítmicas:** A través de descripciones algorítmicas que especifican cómo se comporta la función para las entradas definidas.

4. ¿Cuál es la importancia de las funciones parciales en la teoría de la computabilidad?

Modelado de Computación Realista: Muchos problemas del mundo real no se pueden representar con funciones totales, por lo que las funciones parciales permiten un modelado más realista.

- **Manejo de Incertidumbre:** En situaciones donde no se conocen todos los posibles valores de entrada, las funciones parciales proporcionan un enfoque flexible para tratar la incertidumbre.

- **Relevancia en la Programación:** En programación, las funciones parciales son comunes cuando se trata de manejar excepciones o casos especiales en algoritmos y programas.

5. ¿Qué caracteriza a una máquina de Turing?

a) Utiliza lenguaje de alto nivel para operar



- b) Es un modelo abstracto de una computadora con una cinta infinita
 - c) Opera únicamente con números reales
 - d) Ejecuta instrucciones en paralelo
6. ¿Cuál es la principal capacidad de una máquina de Turing?
- a) Resolver problemas de inteligencia artificial
 - b) Realizar operaciones aritméticas complejas
 - c) Simular el comportamiento de cualquier algoritmo computacional
 - d) Almacenar grandes cantidades de información en su memoria
7. ¿Qué tipo de lenguajes son aceptados por una máquina de Turing?
- a) Solo lenguajes formales regulares
 - b) Lenguajes formales libres de contexto
 - c) Lenguajes formales sensibles al contexto
 - d) Cualquier lenguaje formal reconocible por una gramática formal
8. ¿Cuál de las siguientes afirmaciones es verdadera acerca de la decidibilidad en la teoría de la computabilidad?
- a) Todos los problemas son decidibles mediante algoritmos
 - b) Existen problemas indecidibles, imposibles de resolver mediante algoritmos
 - c) Solo los problemas matemáticos son decidibles
 - d) La decidibilidad depende del lenguaje en el que está expresado el problema
9. ¿Qué define la computabilidad de un problema?
- a) La capacidad de ser expresado en términos matemáticos
 - b) La existencia de una solución algorítmica
 - c) La dificultad del problema
 - d) La cantidad de recursos computacionales requeridos para resolverlo
10. ¿Qué representa el concepto de "lenguaje recursivamente enumerable" en teoría de la computabilidad?
- a) Un lenguaje que puede ser resuelto por cualquier algoritmo
 - b) Un lenguaje que puede ser generado por una gramática libre de contexto
 - c) Un lenguaje para el cual existe una máquina de Turing que lo acepta
 - d) Un lenguaje no computable



11. ¿Cuál es el papel de la función de transición en una máquina de Turing?
 - a) Determina el número de estados de la máquina
 - b) Define la entrada de la máquina
 - c) Especifica el comportamiento de la máquina ante un estado y un símbolo leído
 - d) Controla la velocidad de ejecución de la máquina
12. ¿Qué caracteriza a un lenguaje recursivo en teoría de la computabilidad?
 - a) Es un lenguaje reconocido por una máquina de Turing que siempre se detiene
 - b) No es aceptado por ninguna máquina de Turing
 - c) Puede ser resuelto por cualquier algoritmo
 - d) Es equivalente a un lenguaje regular
13. ¿Qué implica el problema de la detención en la teoría de la computabilidad?
 - a) Determinar si un programa se ejecuta de manera correcta
 - b) Saber si una máquina de Turing se detendrá o entrará en un bucle infinito
 - c) Resolver un problema de optimización
 - d) Calcular el tiempo de ejecución de un algoritmo
14. ¿Cuál es la relación entre la teoría de la computabilidad y la complejidad computacional?
 - a) La teoría de la computabilidad estudia la dificultad intrínseca de los problemas computacionales
 - b) La complejidad computacional no tiene relación con la teoría de la computabilidad
 - c) La complejidad computacional se enfoca en la resolución de problemas prácticos
 - d) La teoría de la computabilidad se concentra en el rendimiento de los algoritmos en situaciones reales
15. ¿Cuál es el objetivo principal de una máquina de Turing?
 - a) Resolver problemas de inteligencia artificial
 - b) Ejecutar programas de alto nivel
 - c) Modelar algoritmos computacionales
 - d) Analizar estructuras de datos complejas
16. ¿Cuál es la función principal de la cinta en una máquina de Turing?
 - a) Almacenar instrucciones de operación
 - b) Almacenar datos de entrada y salida
 - c) Almacenar números reales
 - d) Controlar el flujo de ejecución



17. ¿Qué representa el concepto de "computabilidad" en la teoría de la computación?
 - a) Capacidad de resolver problemas complejos
 - b) Existencia de una solución algorítmica para un problema
 - c) Complejidad intrínseca de un algoritmo
 - d) Facilidad de implementación de un algoritmo
18. ¿Qué caracteriza a un problema indecidible en la teoría de la computabilidad?
 - a) No existe un algoritmo que pueda resolverlo
 - b) Todos los problemas son decidibles
 - c) Puede ser resuelto por cualquier computadora
 - d) Depende de la codificación del problema
19. ¿Cuál es la función principal de una gramática en la teoría de lenguajes formales?
 - a) Generar lenguajes recursivos
 - b) Definir la estructura sintáctica de un lenguaje
 - c) Describir el comportamiento de una máquina de Turing
 - d) Establecer operaciones de entrada y salida
20. ¿Cuál es la diferencia entre un lenguaje recursivamente enumerable y un lenguaje recursivo?
 - a) Los dos términos son sinónimos
 - b) Los lenguajes recursivos no son aceptados por una máquina de Turing
 - c) Los lenguajes recursivamente enumerables pueden no ser decidibles
 - d) Los lenguajes recursivos siempre son finitos
21. ¿Cuál es el concepto central detrás del teorema de la indecidibilidad de Turing?
 - a) Demuestra que ciertos problemas no son resolubles por una máquina de Turing
 - b) Establece que todos los problemas son resolubles por una máquina de Turing
 - c) Define un nuevo modelo de computación más poderoso que una máquina de Turing
 - d) Indica que los problemas deben ser expresados en lenguajes formales
22. ¿Qué describe el concepto de "lenguaje formal" en teoría de la computabilidad?
 - a) Un lenguaje hablado por humanos
 - b) Un conjunto de cadenas sobre un alfabeto dado
 - c) Un lenguaje comprensible por una máquina de Turing
 - d) Un lenguaje utilizado en la programación de sistemas



23. ¿Qué representa el problema de la parada en la teoría de la computabilidad?
- a) Determinar si un programa tiene errores
 - b) Saber si un programa finalizará su ejecución o entrará en un bucle infinito
 - c) Calcular la complejidad temporal de un algoritmo
 - d) Establecer la eficiencia de un programa
24. ¿Qué estudia la teoría de la complejidad computacional?
- a) La dificultad de resolver problemas en tiempo polinomial
 - b) La cantidad de recursos necesarios para resolver problemas
 - c) La eficiencia de los algoritmos en el mundo real
 - d) La capacidad de resolver problemas complejos en lenguajes formales