

NAME	PAGES	SPEAKER/CLASS	DATE - TIME
Diego Alejandro Vargas Suroso	2-4	Fundamentos de programación	4-12-2023

Title: Introducción a los lenguajes formales

Keyword
 Automates
 finitos
 lenguajes
 Conjuntos
 Cadenas
 Inversa

Topic: Automates finitos

Son una representación gráfica de los lenguajes regulares y constan de estados, alfabeto, estado inicial, estados finales y una función de transición. Se presentan diagramas de transición y tablas de transición para representar AF, y se introducen los autómatas finitos determinísticos (AFD) y no determinísticos (AFN). Mientras que en AFD tiene una transición clara entre estados, un AFN puede tener múltiples transiciones posibles.

Questions

¿Cómo se define un sistema en el contexto de los autómatas finitos?
 ¿Cuáles son los elementos fundamentales del sistema finito?

Los autómatas finitos son sistemas que procesan entradas para producir salidas y su complejidad varía desde sistemas biológicos hasta autómatas finitos.

Se introduce la terminología básica, como Cadenas, C. Vacía, inversa de una Cadena y Cadena Elevada a una potencia. Se exploran operaciones de Conjuntos como unión, intersección y Complemento relativo y positivo.

Summary: Los Automatas finitos desempeñan un papel crucial. Se definen por su capacidad de procesar cadenas de símbolos y determinar si pertenecen a un lenguaje específico.

NAME: Diego Alejandro Vargas Paez
 PAGES: 4-4
 SPEAKER/CLASS: fundamentos de programación
 DATE - TIME: 9-12-2023

Title: Introducción a los lenguajes formales

Keyword

Topic: Teoría de la Computabilidad

Teoría

Computabilidad

Función

tesis

Algoritmos

Complejidad

Es una rama de la informática que se ocupa de analizar y determinar qué problemas pueden resolverse mediante algoritmos o Máquinas de Turing (MT).

La tesis de Church-Turing postula que si una máquina de Turing no puede resolver un problema, ninguna otra computadora podría hacerlo. La teoría de la Complejidad se centra en evaluar los recursos (tiempo y espacio) necesarios para resolver un problema.

Questions

Los algoritmos se clasifican en polinomiales (P) y no polinomiales (NP). Los algoritmos polinomiales tienen una relación polinómica entre el tamaño del problema y el tiempo de ejecución. Los lineales (L) y logarítmicos ($L \log n$) son tipos específicos de algoritmos polinomiales. Los no polinomiales (NP) tienen un crecimiento exponencial en el tiempo de ejecución y se consideran "parcialmente computables" para valores pequeños de entrada.

¿Qué es la teoría de la Complejidad?

¿Cuál es la clasificación de los algoritmos en términos de tiempo y espacio?

Summary:

Aborda la capacidad de resolver problemas mediante algoritmos o Máquinas de Turing (MT). Evalúa los recursos (Tiempo y Espacio) necesarios para resolver problemas.

NAME	PAGES	SPEAKER/CLASS	DATE - TIME
Diego Adolfo Vargas ^{Peñero}	3 - 4	Fundamentos de programación	4-12-2023

Title: Introducción a los lenguajes formales

Keyword

Topic: máquina de estado finito

Finito
 Función
 Turing
 Cinta
 Alfabeto
 Lenguajes
 Expresiones

Es una representación especial de autómatos finitos. Se utiliza en diversas aplicaciones, desde el funcionamiento de elevadores hasta máquinas tragamonedas. Se define como un sistema con un conjunto finito de estados, entradas y salidas, con funciones de estado siguiente y de salida.

La equivalencia entre autómatos finitos y máquinas de estado finito muestra que comportan elementos como alfabetos y conjuntos de estados.

Questions

¿Cómo se define una máquina de estado finito?
 ¿Cuál es la función de la máquina de estado finito en una computadora?

Las Máquinas de Turing son dispositivos más complejos que consisten en una cinta infinita y una cabeza de lectura-escritura. Pueden realizar tareas más arduas y reconocer lenguajes regulares y no regulares. Son más poderosas que los autómatos finitos, ya que admiten lenguajes no regulares. Se presenta un ejemplo de una Máquina Turing que reconoce el lenguaje x^*y .

Summary: Se destaca que las máquinas de Turing son equivalentes a los computadores en términos de su capacidad para realizar procesos complejos. Se compara su capacidad con los autómatos de pila al abordar el lenguaje $\{x^n y^n \mid n \geq 1\}$.