NAME	CLASS	SPEAKER	DATE & TIME
Josué Ismael Ruiz P. 2023 0832	Programación en C	Carlos Richardo	6/12/2023

Title: Grafos

Keyword

Grafos

aristas

Vertices

orden del Vertice

Questions

Número de lineas que salen o entran a un Vértice lo llamó "orden del Vertice".

Topic: Introducción

El texto nos introduce a la teoria de grafos, comenzando con la resolución del problema de los puentes de Konigsberg por Leonhard en el siglo XVIII. Euler introdujo la representación de problemas mediante "grafos", donde los puntos son vértices y la lineas son aristas. Estableció que solo es posible establecer un ciclo que pase por todas las aristas una vez si todos los vértices tienen valencia par. Los grafos son útiles para representar relaciones en diversas áreas, como la estructura de empresas, redes eléctricas, telefónicas, carreteras etc. En computación se usen para mostrar relaciones entre archivos, registros, computadoras y redes.

Tambien en el texto se describe a Leonhard Euler, destacado matemático y físico Suizo del siglo XVIII, conocido por contribuciones en áreas como calculo, teoria de grafos y geometria. A pesar de perder la vista en 1738, Euler fue increiblemente productivo, resolviendo el problema de los puentes de Konigsberg en 1735, estableció el primer teorema de la teoria de grafos.

Summary: En estos textos se aborda la figura destacada de Leonhard Euler, un eminente matemático y físico Suizo del siglo XVIII. Euler realizo contribuciones significativas en diversas disciplina como el calculo, la teoria de grafos y la geometria, a pesar de perder la vista en 1738. Su resolución de problema de los puentes de Konigsberg marcó el inicio de la topologia.

NAME	CLASS	SPEAKER	DATE & TIME
José I. Daniel Paulino 2023-0882	Programación en C	Carlos Richardo	6/12/2023

Title: Grafos

Keyword

Topic: Partes y tipos de grafos

El texto introduce el concepto de grafo, compuesto por un conjunto de vertices y un conjunto de lados. Los vertices se representan con circulo y se les asigna un número o letra, mientras que los lados son las lineas que conectan los vertices, tambien con asignación de letra o números. Se describen elementos como lados paralelos, lazo y la valencia de un vértice (número de lados que salen o entran).

Questions

El texto habla sobre los distinto tipos de grafos como son: Grafos simples, grafos completo (Kn), complemento de un grafos (G), grafos Bipartido, grafos Bipartido completo (Kn, m).

- Grafos simples: No tienen lazos ni lados paralelos.
- Grafos completos (Kn): Cada vertice se conecta con todos los demás, sin lazos ni lados paralelos.

Complemento de un grafo (G): El grafo que falta para formar un grafo completo de n vertice. Carece de lazos y ramas paralelas.

Summary: El texto introduce el concepto de grafo, formado por vertices y lados con asignaciones de número o letra. Se destacan elementos como lados paralelos, lazos y valencia de un vertice. Se describen distintos tipos de grafos, como los simples (sin lazos ni lados paralelos entre otros.

NAME	CLASS	SPEAKER	DATE & TIME
Josué I. Suriel Paulino 2023-0882	Programación en C	Carlos Pichardo	6/12/2023

Title: Grafos

Keyword

matricial de grafos

Topic: Representación matricial, camino y circuito

El texto aborda la representación matricial de grafos, destacando la utilidad de matrices para sistemas de ecuaciones y grafos. Se describen dos tipos de matrices: la matriz adyacencia, que utiliza 1 y 0 para indicar la presencia o ausencia de relaciones entre vértice, y la matriz de incidencia, que utiliza 1 y 0 para representar la conexión entre vértice y arista.

Questions

Se introduce el concepto de camino de Euler, que recorre todos los vertices pasando por cada arista una vez, y el circuito de Euler, que realiza un ciclo similar. Se proporciona un algoritmo para determinar circuito de Euler. También se menciona el circuito de Hamilton que pasa por cada vertice una vez.

Se habla de los grafos conexos que es un grafo en el que para cualquier par de vertice w y v distinto, existe un trayecto.

Summary: El texto aborda la representación matricial de grafos mediante matrices de adyacencia e incidencia, resaltando su utilidad en sistema de ecuaciones y analisis de grafos. Se explican conceptos como caminos, circuitos y grafos conexos. Se introduce el camino de Euler y el circuito de Euler con algoritmo para determinar el mismo.

NAME	CLASS	SPEAKER	DATE & TIME
Josué I. Samuel Paulino 2023-0882	Programación en C	Carlos Richardo	6/12/2023

Title: Grafos

Keyword

Circuitos de Hamilton mediante ajustes manuales

Topic Isomorfismo

El isomorfismo en grafos señala una equivalencia estructural entre dos grafos, G1 y G2, a pesar de sus representaciones visuales distintas. La igualdad se verifica mediante propiedades cruciales: el número de lados, vértices, valencias, la conectividad y características específicas de circuitos. Para demostrar isomorfismo, se compara la información esencial de los grafos evitando la complejidad de igualar matrices de incidencia.

Questions

Se reconoce la dificultad intrínseca de demostrar isomorfismo mediante ajustes manuales o computacionales en matrices y aristas. En lugar de este enfoque, se destaca la utilidad de comparar propiedades clave de los grafos, permitiendo determinar la igualdad si coinciden en todos estos aspectos esenciales. Este método ofrece una alternativa práctica y efectiva.

Summary: El isomorfismo en grafos implica que dos grafos son estructuralmente iguales, a pesar de su apariencia diferente. La igualdad se basa en propiedades clave, como número de lados, vertices, valencias y conectividad. En lugar de igualar matrices.

NAME	CLASS	SPEAKER	DATE & TIME
Josué I. Israel Paulino 2023-0882	Programación 3 C	Carlos Richardo	6/12/2023

Title Grafos

Keyword

Topic Grafos planos

Questions

un grafo se considera plano si se puede dibujarse en un plano sin que sus aristas se crucen. La ecuación de Euler, A = L V + 2, es aplicable a grafos planos y conectados donde A es el número de áreas. L es el número de lados y V es el número de vertices

Se introduce el teorema de Kuratowski, que establece que un grafo es plano si y solo si no contiene subgrafos a K5 o K3,3 K5 implica al menos cinco vertices con valencia cuatro o más mientras que K3,3 requiere al menos seis vertices con valencia tres y una Bipartición.

Se destaca el ejemplo del grafo de Petersen, demostrando que no es plano al contener un subgrafo K3,3. La eliminación de aristas y vertices revela este subgrafo, confirmando la no planaridad del grafo de Petersen. Este enfoque en subgrafos isomorfos a K5 o K3,3 simplifica la determinación de planaridad de grafos complejos.

Summary: un grafo es plano si puede ser dibujado en un plano sin que sus aristas se crucen. y este propiedad se rige por la ecuación de Euler: A = L V + 2, donde A es el número de áreas, L es el número de lados y V es el número de vertice. El teorema de Kuratowski establece que un grafo es plano sino contiene subgrafo isomorfo a K5 o K3,3.

NAME	CLASS	SPEAKER	DATE & TIME
José I. Daniel Rodríguez 2023-0292	Programación de C	Carlos Rodríguez	6/12/2023

Title: Grafos

Keyword

Topic: coloración de grafos

La coloración de grafos es un concepto que implica asignar a los vertices de un grafo, asegurando que vertices adyacentes no comparten el mismo color. Se utiliza el número cromático, simbolizado como $\chi(G)$, que representa la cantidad mínima de colores necesario para lograr esta coloración.

Questions

Características notables del número cromatico incluyen:

un grafo sin aristas tiene un número cromático de 1.

un camino o ciclo de longitud 2 tiene un número cromático de 2.

un grafo completo K_n tiene un número cromatico igual a n.

Grafos bipartitos completos ($K_{n,m}$) tienen número cromático de 2.

El teorema de los cuatro colores establece que cualquier grafo plano puede ser coloreado con cuatro colores diferentes.

Summary: La coloración de grafos implica asignar colores a vértices, evitando colores iguales en vertices adyacentes. El número cromático ($\chi(G)$) es la cantidad minima de colores necesarios. El teorema de los cuatro colores dice que cualquier grafo plano se puede colorear con cuatro colores.

NAME	CLASS	SPEAKER	DATE & TIME
Josué I. Suriel Rubio 2023-0882	Programación en C	Carlos Richardo	6/12/2023

Title: Grafos

Keyword

Topic: Aplicaciones de los grafos

Los grafos, al ser estructuras que representan relaciones entre elementos, encuentran aplicaciones diversas. Los grafos de similaridad, al agrupar información con propiedades semejante, son útiles en reconocimiento de patrones, como en la detección de enfermedades basada en conjuntos de celulas con la característica similares.

Questions

En el ámbito de la optimización, los grafos ponderados permiten abordar el problema de encontrar la ruta más corta, crucial en logística o redes de transporte.

Grafos de Similaridad: Los grafos de similaridad se utilizan para agrupar información con características afines.

Ruta más corta en grafos ponderados: En contextos logísticos o de redes de transporte, los grafos ponderados ofrecen una representación eficaz.

Summary: Los grafos tienen aplicaciones extensas y diversas: Los grafos de similaridad son útiles para reconocimiento de patrones, agrupando datos con características comunes. En la determinación de rutas más cortas en grafos ponderados, se utiliza el algoritmo de Dijkstra, fundamental para optimizar recursos.

NAME	CLASS	SPEAKER	DATE & TIME
Jesús I. Daniel Rubio 2023-0882	Programación en C	Carlos Richardo	6/12/2025

Title: Grafos

Keyword

Topic: Resumen

Los grafos, constituidos por vertices y aristas se clasifican en tipos como simples, completos y bipartidos, representados visualmente mediante circulos y lineas. Características clave incluyen valencia, la presencia de lazos y lados paralelos.

Questions

La teoría del grafos explora conceptos como conexidad, isomorfismo y grafos ponderados. En el reconocimiento de patrones, los grafos de similaridad agrupan información afin. El algoritmo de Dijkstra optimiza rutas en grafos ponderados, beneficiando la logistica y el transporte.

En sintesis, los grafos, aplicados en redes, quimica, organización y computación, destacan como herramienta versátil para modelar y resolver problema del mundo real.

Summary: Los grafos, estructuras de vertices y aristas, clasificados en tipos como conexos y ponderados, encuentran aplicaciones en redes y optimización de rutas. La teoría de grafos abarca conceptos como la coloración y el reconocimiento de patrones. En resumen son herramientas versátiles

NAME	CLASS	SPEAKER	DATE & TIME
José I. Daniel Pardo 2022-0722	Programación III C	Carlos Richards	6/12/2023

Title: Árboles

Keyword

Topic: Introducción

El texto podemos ver los desafíos en el tratamiento de grafos en informática debido a su falta de estructura y reglas definidas. Se puede destacar la preferencia por el uso de árboles, que son grafos con características específicas para una mejor organización de la información. En el ámbito computacional, se busca tanto el desarrollo de equipos con mayor capacidad de almacenamiento como la entrega más rápida y ordenada de resultados.

Questions

Se habla de aplicaciones históricas de árboles en redes eléctricas y expresiones matemáticas. y se destaca su uso actual en la clasificación de información, bases de datos, codificación, estructuras de datos y reconocimiento de patrones.

nos define a lo último lo que es un árbol como un grafo conexo sin ciclos.

Summary: El texto destaca los desafíos en el tratamiento de grafos en informática y aboga por el uso de árboles para una mejor organización de la información. Se busca aumentar la capacidad de almacenamiento en hardwares y mejorar la capacidad y rapidez y orden en la entrega de result

NAME	CLASS	SPEAKER	DATE & TIME
José I. Suriel Rubino 2023-0682	Programación en C	Carlos Richardo	6/12/2023

Title: Árboles

Keyword

Topic: tipos y propiedades de los árboles

Questions

El texto destaca las propiedades fundamentales de los árboles en la teoría de grafos. un árbol se define como un grafo conexo sin ciclos ni lados paralelos. Se introducen términos como nodos, ramas, niveles y raíz, con la altura o peso del árbol como la medida de su nivel más bajo.

Luego se exploran clasificaciones de árboles según el número de nodos y la altura. Se describen árboles binarios, trinarios, cuaternarios etc. Destaca la importancia de los árboles binarios en sistemas digitales debido a su naturaleza de valores (0, 1).

En término de clasificación por altura, se diferencian árboles balanceados y desbalanceados. un árbol se considera balanceado si la diferencia de altura entre sus hojas es máximo 1.

Summary: El texto habla sobre las propiedades y clasificaciones de los árboles en la teoría de grafos. Se mencionan términos clave como nodos, ramas, hojas, y se abordan clasificaciones según el número de nodos y la altura. Se destaca la importancia de los árboles binarios en sistemas digitales.

NAME	CLASS	SPEAKER	DATE & TIME
Jesús I. Suriel Paulino 2023-0382	Programación en C	Carlos Pichardo	6/12/2023

Title: árboles

Keyword

Topic: Bosques

El texto explora la relación entre árboles y bosques en la teoría de grafos. Se establece que un bosque es un conjunto de árboles, donde un árbol a su vez, puede considerarse un bosque conectado. Se enfatiza que el árbol más pequeño consiste en al menos dos nodos conectados por una arista. Se ilustra la distinción entre árboles y bosques mediante ejemplos gráficos, resaltando la condición de conexidad y la ausencia de ciclos como características clave para clasificar un grafo como árbol. El análisis se complementa con ejemplos específicos que demuestran como un grafo puede ser clasificado como árbol o bosque según sus propiedades.

Questions

Summary: El texto explora la relación entre árboles y bosque en teoría de grafos. Un bosque es un conjunto de árboles, y un árbol puede considerarse un bosque conectado. La condición clave es que un árbol tiene al menos dos nodos conectados por una arista.

NAME	CLASS	SPEAKER	DATE & TIME
José I. Daniel Paulino 2023-0882	Programación en C	Carlos Rodríguez	6/12/2023

Title: Árboles

Keyword

Topic: Árboles con Pesos

El texto explora el concepto de árboles con pesos en la codificación de caracteres, particularmente a través del "código de Huffman". Este método busca mejorar la eficiencia en el uso de memoria y la velocidad de procesamiento al asignar cadenas de bits más cortas a caracteres frecuentes. Se describe el proceso de codificación y decodificación mediante un árbol binario completo, donde los caracteres más frecuentes están más cerca de la raíz.

Questions

Se destaca la importancia de la frecuencia de uso en la determinación cadenas más pequeñas.

Se explica el proceso de obtención de un árbol optimo, ordenando pesos, combinando nodos de menor peso y construyendo un árbol binario con altura minima

Summary: El texto explora el uso de árboles ponderados, como el "código de Huffman", para optimizar la codificación de caracteres. Busca asignar cadenas de bits cortas a caracteres frecuentes, destacando la importancia de la frecuencia de uso.

NAME	CLASS	SPEAKER	DATE & TIME
Josué I. Suriel Paulino 2023-0882	Programación en C	Carlos Richardo	6/12/2023

Title: Árboles

Keyword

Topic: Árboles generadores

El texto aborda la obtención de árboles generadores en grafos conexos. Un árbol generador es un subgrafo que mantiene todos los nodos del grafo original, eliminando aristas redundantes. Se describen dos metodos para obtener árboles generadores: busqueda a lo ancho y busqueda en profundida.

Questions

En la busqueda a lo ancho, se inicia desde la raiz, explorando nodos por niveles de izquierda a derecha. La busqueda en la profundidad comienza en la raíz, explorando los nodos en profundidad antes de pasar al siguiente nivel.

Se introduce el concepto de árbol generador minimo, que es aquel con la menor suma de pesos en las aristas. Se presentan dos metodos para obtenerlo: el de prim y el de Kruskal. Se concluye que ambos metodos pueden dar arboles generadores.

Summary: El texto aborda la obtención de árboles generadores en grafos conexos, utilizando metodos como busqueda a lo ancho y en profundida. Se introducen los conceptos de árboles generador minimo, destacando los metodos de prim y Kruskal.

NAME	CLASS	SPEAKER	DATE & TIME
José I. Juriel Rubio 2023-0852	Programación en C	Carlos Pichardo	6/12/2023

Title: árboles

Keyword

Topic: Recorrido de un árbol

El texto se centra en los recorridos de árboles, ofreciendo tres enfoques distinto: en orden primero, en orden segundo y en orden final. Estos recorrido definen el orden en que se procesan los nodos y sus descendientes en un árbol, además, se explora el uso de árboles etiquetadas en el contexto de la evaluación de expresiones matemáticas, donde las hojas representan variables y constantes y los nodos intermedios actúan como operadores.

Questions

Recorridos de árboles:

En orden primero: Comienza con el nodo raíz, segudo del hijo izquierdo y, posteriormente, de los demás hijos.

En orden segundo: Inicia con el hijo izquierdo, luego el padre y finalmente los demás hijos.

En orden Final: Comienza con el hijo izquierdo.

Summary: El texto aborda tres tipos de recorridos en arboles: en orden primero, en orden segundo y en orden final. Se destaca el uso de árboles etiquetados en la evaluación de expresiones matemática, donde las variables y constantes son hojas y los operadores son nodos intermedios.

NAME	CLASS	SPEAKER	DATE & TIME
José I. Daniel Paulino 2023-0382	Programación en C	Carlos Pichardo	6/12/2023

Title: Árboles

Keyword

Topic: Busquedas

El texto aborda la importancia de la eficiencia en la búsqueda de información en computadoras, proponiendo el uso de estructuras de datos como árboles, con enfasis en los árboles de búsqueda binarios (ABB), árboles AVL y árboles B.

Questions

Árboles de busqueda Binarios (ABB):

- Proporciona una solución para organizar y acceder eficientemente a la información almacenada.
- Destaca la simplicidad en la creación y manipulación de ABB.

operaciones en arboles de Busqueda:

- Detalla las operaciones de eliminación de nodos en ABB.
- Prioriza al hijo izquierdo para ocupar el lugar de un nodo eliminado.

Explora cómo se reorganiza el árbol después de eliminarlo.

Summary: El texto destaca la importancia de usar árboles, especificamente arboles de busqueda binarios (ABB), para optimizar la búsqueda de información en computadora. Proporciona un ejemplo de creación y búsqueda en un ABB, aborda la eliminación de nodos y sugiere el uso de arboles.

NAME	CLASS	SPEAKER	DATE & TIME
Josué I. Daniel Paulino 2023-0882	Programación en C	Carlos Richardo	6/12/2023

Title: árboles

Keyword

Topic: Aplicación de los árboles

El texto aborda la aplicación de diversas estructuras de árboles, como los árboles binarios, AVL y B, en la gestión eficiente de información en entornos computacionales. Se destaca la importancia de organizar datos de manera que el acceso sea rápido y eficaz, considerando la elección de tipo de árbol en función de la estructura y el volumen de la información.

Questions

Se proporciona un ejemplo específico de la creación de un árbol de búsqueda binaria con números dados, ilustrando cómo se pueden organizar los datos de manera que la búsqueda y recuperación de información sean eficientes.

El texto resalta como el uso de árboles, ya sean binarios, AVL o B, se adapta a diferentes escenarios según el tamaño y estructura.

Summary: El texto profundiza en cómo la elección cuidadosa de estructuras de árbol puede mejorar significativamente las operaciones de búsqueda y manipulación de datos en sistemas informáticos, considerando factores como el equilibrio, la eficiencia y la distribución en memoria secundaria.

NAME	CLASS	SPEAKER	DATE & TIME
José I. Daniel Paulino 2023-0882	Programación Sn C	Carlos Pichardo	6/12/2023

Title: árboles

Keyword

Topic: Resumen

El texto presenta conceptos fundamentales sobre árboles en teoría de grafos. un árbol es definido como un grafo conexo sin ciclos, lazos ni lados paralelos, con jerarquía determinada por niveles, donde la cúspide es la raíz y la altura es el valor del nivel más bajo. Se describen términos como "hojas", "descendientes", "antecesores" y "vértices interno" para clasificar los elementos del árbol.

Questions

Los árboles pueden ser binarios, trinarios, cuaternarios etc., y se clasifican como balanceados o desbalanceados según su altura.

Se subraya la importancia de utilizar árboles de busqueda binarios (ABB), árboles AVL y árboles B para organizar datos de manera eficiente en la memoria de una computadora.

Summary: El texto proporciona una vision integral de los conceptos básicos de árboles en teoría de grafos, su aplicación en la organización eficiente de datos y su relevancia en diversos campos, incluyendo la computación y la optimización de sistema a bajo costo.

NAME	CLASS	SPEAKER	DATE & TIME
José I. Daniel Rubio 2023-0882	Programación en C	Carlos Richards	6/12/2023

Title Introducción a los lenguajes formales

Keyword

Topic: Introducción

Questions

La introducción explora la naturaleza de los lenguajes formales. Se define un lenguaje como un conjunto de símbolos o palabras junto con métodos para estructurar y combinar esos símbolos. El lenguaje natural abarca todas las palabras y reglas para estructurar palabras, frases y oraciones. Se señala que los lenguajes formales, como el binario, Java, C, Basic o Pascal, son utilizados en la interacción con computadoras pero son más limitados en capacidad para simular y modelar lenguajes naturales.

Se hace enfasis en que la dificultad no solo radica en la estructuración correcta de las palabras (sintaxis) en los lenguaje formales, sino también en el significado de las palabras o frases (semántica). Este significado puede variar según la persona.

Summary: La introducción explora entre el lenguaje natural y los lenguajes formales utilizados en computación. Destacada la limitación de los lenguajes formales, como el binario, para modelar la complejidad de los lenguajes naturales.

NAME	CLASS	SPEAKER	DATE & TIME
José I. Ruiel Paulino 2023-0882	Programación 2 n1	Carlos Richardo	6/12/2023

Title: Introducción a los lenguajes formales

Keyword

Topic: Gramáticas y lenguaje formales
El texto aborda la relación entre gramáticas y lenguajes formales, centrandose en el concepto de lenguaje ($L(G)$) basado en la gramática G. Se describen los elementos de una gramática, como el alfabeto (Σ). Los simbolos no terminales (N), los simbolos terminales (T), el estado inicial (S), y las composiciones (C) que definen las reglas, y se proporciona un ejemplo con $\Sigma = (,\ldots,)$ $\Sigma = (a, b, c, \ldots, z)$ y $() = h, \ldots)$ $L(G) = (hola, conejo, pera, piña, \ldots)$.

Questions

La clasificación de las gramáticas se presenta en cuatro tipos (0, 1, 2, 3) según las restricciones impuestas a las composiciones.

Luego, se aborda la representación de las gramáticas, dividiendolas en gramáticas regulares, libres de contexto y sensibles al contexto.

Summary: El texto explora la relación entre gramática y lenguajes formales, destacando la importancia de gramática libres de contexto en la creación de lenguajes formales. Se discuten las clasificaciones de gramáticas y sus representaciones mediante automatas finitos árboles de derivación, notación BNF.

NAME	CLASS	SPEAKER	DATE & TIME
Jesué Ismael Jurado 2023-0882	Programación en C	Carlos Richardo	6/12/2023

Title Introducción a los lenguajes formales

Keyword Topic máquinas de estado finito

El texto aborda la teoría de autómatas, destacando las máquinas de estado finito y las maquinas de turing.

1 - Máquinas de estado finito:
Representan automátas finitos sin estados aceptados.
usadas en diversas aplicaciones con estados, entrada y salidas definidas.

Questions

2 - Máquinas de turing:
Propuestas por Alan turing, fundamentales en la teoría de computación.
Reconocen lenguajes regulares y no regulares superando a los autómatas finitos.

3 - Ejemplo de máquina de turing:
Ejemplo práctico donde una máquina de turing reconoce el complemento de una cadena binaria.

4 - Lenguaje y máquina de turin: Explora ejemplo de lenguaje reconocido por máquina.

Summary: El texto explora máquina de estado finito y máquina de turing. Destacada la versatilidad de la máquina de turing para reconocer lenguajes regulares y noregulares. utiliza ejemplo para ilustrar su aplicación en el reconocimiento de cadenas binarias y subraya su potencial en la resolución de problemas complejos.

NAME	CLASS	SPEAKER	DATE & TIME
José I. Daniel Paulino 2023-0382	Programación en C	Carlos Pichardo	6/12/2023

Title: Introducción a los lenguajes formales

Keyword

Topic: Autómatas finitos

El texto inicia explorando los conceptos fundamentales de teoría de autómatas, abordando gramáticas libres de contexto, arboles de derivación y diversas representaciones como la notación BNF. Se destaca la relación entre gramáticas y autómatas, incluyendo autómatas de pila y máquinas de turing. Luego, se profundiza en autómatas finitos, describiendo su función como sistemas que procesan entradas para producir salidas, en contraste con sistemas infinitos como el comportamiento humano.

Questions

Se discuten autómatas finitos determinístico (AFD) y no determinísticos (AFN), y se demuestra la conversión de un AFD. El texto finaliza destacando la importancia de estos conceptos en la teoría de la computación.

Summary: El texto abarca desde los fundamentos teóricos hasta aplicaciones práctica, proporcionando una comprensión integral de la teoría de autómatas y sus diversas representaciones.

NAME	CLASS	SPEAKER	DATE & TIME
Josué Ismael Suriel P. 2023-0882	Programación en C	Carlo Pichardo	6/12/2023

Title: Introduccion a los lenguajes formales

Keyword

Topic: máquinas de estado finito

El texto aborda la teoría de autómatas, destacando la máquinas de estado finito y la maquinas de turing.

1. máquinas de Estado Finito:

- Representan autómatas finitos sin estados aceptado.
- usadas en diversas aplicaciones con estados, entrada y salidas definidas.

2. máquinas de turing:

Propuestas por Alan turing, fundamentales en la teoría de la computación. Reconocen lenguajes regulares y no regulares, superando a los autómatas finito.

3. Ejemplo de máquina de turing:

Explora ejemplos de lenguajes reconocidos por máquinas de turing, demostrando su capacidad para aceptar o rechazar palabras.

Questions

Summary: El texto explora máquinas de estado finito y máquinas de turing. Destacada la versatilidad de las máquinas de turing para reconocer lenguajes regulares y no regulares. Utiliza ejemplos para ilustrar su aplicación en el reconocimiento de cadenas binarias y subraya su potencial en Resolución de prob.

NAME	CLASS	SPEAKER	DATE & TIME
José I. Suriel Paulino 2023-0882	Programación en C	Carlos Richardo	6/12/2023

Title: Introducción a los lenguajes formales

Keyword

Topic: teoría de la computabilidad

La teoría de la computabilidad aborda la capacidad de la computadoras para resolver problemas mediante algoritmos o máquina de turing (MT). Iniciada en la decada de 1930, surge del problema Entscheidungsproblem de Hilbert, buscando un algoritmo para validar proposiciones matemáticas. Godel y Church contribuyen estableciendo limitaciones y funciones recursivas.

Questions

La complejidad computacional considera los recursos (tiempo y espacio) necesarios para resolver problemas. Se clasifican algoritmos en polinomiales (P) y no polinomiales (NP). Los polinomiales tienen complejidad manejable, mientras que los no polinomiales, como aquellos con crecimiento exponencial (NP), pueden ser problemáticos para valores grandes de entrada. Aunque se prefieren algoritmos polinomiales.

Summary: El texto explora la teoría de la computabilidad y complejidad, originada en los años 30 en la busqueda de un algoritmo para validar proposiciones matemática. Destaca las constribuciones de Goldel, Church y turing, culminando en la tesis de church-turing.

NAME	CLASS	SPEAKER	DATE & TIME
Josué I. Suriel P. 2023-0882	Programación en C	Carlos Richardo	6/12/2023

Title: Introducción a los lenguajes formales

Keyword

Topic: aplicación de los lenguajes formales

El texto comienza explicando la aplicación de lenguajes formales mediante el uso de máquinas de estado finito y autómatas finitos deterministas (AFD). Se ejemplifica este concepto con una máquina expendedora de cerveza que acepta monedas y tiene botones para seleccionar entre cerveza clara y obscura. Se presenta un conjunto de estados, entrada y salidas, y se describe como la máquina transita entre estados en función de las entradas.

Questions

En la segunda parte, se aborda el uso de lenguajes libres de contexto en la creación de intérpretes y compiladores para programas de computadora.

Finalmente, se menciona la aplicación de lenguajes sensibles al contexto en campos más avanzados, como la inteligencia artificial (IA).

Summary: El texto aborda la aplicación de lenguajes formales, enfocandose en máquinas de estado finito y autómatas. Se utiliza el ejemplo de una máquina expendedora. también se explora el uso de lenguajes libre de contexto en la creación de intérpretes y compiladores para programas.

NAME	CLASS	SPEAKER	DATE & TIME
Josué I. Suriel Pimbe 2023-0882	Programación en C	Carlos Richardo	6/12/2023

Title: Introducción a los lenguajes formales

Keyword

Topic: Resumen

El texto explora la naturaleza de los lenguajes, dividiendolos en lenguajes naturales y formales, siendo estos ultimos utilizados en la programacion, como el binario o la Java. Se introduce el concepto de gramáticas (representadas por $G = \{\Sigma, N, T, S, C\}$), que definen las reglas de un lenguaje. Se clasifican las gramáticas en cuatro tipos (0,1,2,3) según sus restricciones, destacando las gramáticas libres de contexto como fundamentales para la creación de lenguajes formales y su aplicación en la construcción de compiladores.

Questions

Se mencionan los autómatas finitos deterministicos y no deterministicos, así como la máquinas de estado finito, detallando su estructura y funciones.

Se destaca el desafío actual de diseñar algoritmos más eficientes y mejorar los recursos.

Summary: El texto explora lenguaje formales gramáticas y su aplicación en programación. Destaca la importancia de la gramáticas libres de contexto y menciona conceptos como autómatas finito y máquinas de turing.