



COLEGIO GNET

INFORMATICA

GRADO: 5TO COMPUTACION

INVESTIGACION

TEORIA DE AUTOMATAS

Y COMPILADORES

NOMBRE: Sharon Venetza Martinez Donis

GRADO: 5to Bachillerato en Computación

FECHA: 31/03/2026

INTRODUCCION

Esta investigación trata sobre la teoría de autómatas, gramáticas formales y el funcionamiento de los compiladores. Estos temas son importantes en la informática por que permiten entender como se procesan y analizan los lenguajes de programación.

Los autómatas permiten representar de manera abstracta el comportamiento de sistemas que reconoce patrones, mientras que las gramáticas formales establecen las reglas necesarias para construir leguajes de forma correcta. Por otro lado, los compiladores e interpretes son herramientas clave que traducen el código escrito por los programadores a un lenguaje que la maquina pueda entender.

Además, se estudian procesos importantes como el análisis léxico y sintáctica, los cuales permiten identificar y organizar correctamente los elementos de un programa. También se incluyen algoritmos de búsqueda de cadenas que facilitan el procedimiento de información dentro de textos.

En conjunto, estos temas tienen una gran importancia en el desarrollo de software y en áreas como la inteligencia artificial y el análisis de datos. Por ello, esta investigación tiene como objetivo comprender su funcionamiento y reconocer sus aplicaciones en la vida real.

1. Teoría de autómatas

La teoría de autómatas estudia modelos matemáticos que representan máquinas capaces de procesar información y reconocer patrones. Los autómatas finitos deterministas (AFD) son los más utilizados, ya que siguen reglas definidas para aceptar o rechazar cadenas de texto. Se aplican en programas para validar datos y analizar información.

2. Gramáticas formales

Las gramáticas formales son un conjunto de reglas que permiten construir lenguajes de forma correcta. Las gramáticas regulares son simples y se usan en programación básica, mientras que las gramáticas recursivas permiten estructuras más complejas, como las que se utilizan en lenguajes de programación completos.

3. Análisis léxico

El análisis léxico es la primera fase del proceso de compilación. Se encarga de leer el código fuente y dividirlo en unidades llamadas tokens. Utiliza expresiones regulares para identificar palabras clave, operadores y símbolos, facilitando el análisis del programa.

4. Análisis sintáctico

El análisis sintáctico verifica que el código esté bien estructurado según las reglas del lenguaje. Se apoya en árboles sintácticos para representar la organización del programa. Puede realizarse de forma descendente o ascendente, dependiendo del método utilizado.

5. Algoritmos de análisis de cadenas

Estos algoritmos permiten buscar patrones o palabras dentro de textos. Un ejemplo es el algoritmo Boyer-Moore, que es muy eficiente porque reduce la cantidad de comparaciones necesarias, haciendo la búsqueda más rápida.

6. Compiladores e intérpretes

Los compiladores e intérpretes son herramientas que traducen el código fuente a lenguaje máquina. Este proceso incluye varias fases como el análisis léxico, sintáctico, la generación de código y su optimización. Son fundamentales para ejecutar programas correctamente.

7. Aplicaciones de autómatas

Los autómatas tienen muchas aplicaciones en la vida real, como el reconocimiento de patrones, el procesamiento de lenguaje natural, la inteligencia artificial y el análisis de datos. Gracias a ellos, es posible desarrollar sistemas más inteligentes y eficientes.

Conclusión

En conclusión, la teoría de autómatas, las gramáticas formales y los compiladores son elementos fundamentales en el campo de la informática, ya que permiten comprender cómo las computadoras procesan, interpretan y ejecutan la información. A través de estos conceptos, es posible analizar la estructura de los lenguajes de programación y garantizar que el código funcione de manera correcta.

El estudio del análisis léxico y sintáctico demuestra la importancia de organizar y verificar cada parte de un programa, mientras que los algoritmos de búsqueda de cadenas facilitan el manejo eficiente de la información dentro de los sistemas. Todo esto contribuye al desarrollo de software más preciso y confiable.

Además, los autómatas tienen aplicaciones en diversas áreas como la inteligencia artificial, el reconocimiento de patrones y el análisis de datos, lo que demuestra su utilidad en la vida real. En general, estos temas no solo son importantes a nivel teórico, sino que también tienen un impacto directo en la tecnología que utilizamos diariamente.

Por lo tanto, comprender estos conceptos permite tener una base sólida en programación y valorar la importancia de los procesos que hacen posible el funcionamiento de los sistemas informáticos.

RESUMEN EN EL CUADERNO

31/03/26

Resumen 2

La teoría de autómatos estudia máquinas abstractas que procesan información y reconocen patrones. Un ejemplo importante son los autómatas finitos deterministas (AFD), que se usan en programas para analizar cadenas de texto.

Las gramáticas formales son reglas que definen cómo se construyen los lenguajes. Las gramáticas regulares son más simples y se usan en programación básica, mientras que las recursivas permiten estructuras más complejas.

El análisis léxico es la primera etapa de un compilador, donde el código fuente se divide en partes llamadas tokens. Aquí se utilizan expresiones regulares para identificar palabras clave y símbolos.

El análisis sintáctico revisa el código para tener una estructura correcta según las reglas del lenguaje. Para esto se utiliza un árbol sintáctico y puede hacerse de forma descendente o ascendente.

Los compiladores e intérpretes son herramientas que traducen el código a lenguaje máquina. El compilador pasa por varias fases como análisis léxico, sintáctico, generación y optimización de código.

Finalmente, los autómatos tienen muchas aplicaciones prácticas como el reconocimiento de patrones, la inteligencia artificial, el procesamiento de lenguaje natural y el análisis de datos.