**Teoría de autómatas**

Un autómata finito determinista (AFD) se define como un **modelo matemático que consta de un conjunto de estados, un estado inicial, un alfabeto, un conjunto de estados de aceptación y una función de transición**. Utiliza la función de transición para determinar el siguiente estado en función de la entrada que recibe.

**Ejemplos y aplicaciones** en informática y ciencias de la computación.

Las **ciencias de la computación** o **ciencias de la informática** son las [ciencias formales](https://es.wikipedia.org/wiki/Ciencias_formales) que abarcan las bases teóricas de la [información](https://es.wikipedia.org/wiki/Informaci%C3%B3n) y la [computación](https://es.wikipedia.org/wiki/Teor%C3%ADa_de_la_computaci%C3%B3n), así como su aplicación en los [sistemas informáticos](https://es.wikipedia.org/wiki/Sistemas_de_informaci%C3%B3n).[1](https://es.wikipedia.org/wiki/Ciencias_de_la_computaci%C3%B3n#cite_note-1)​[2](https://es.wikipedia.org/wiki/Ciencias_de_la_computaci%C3%B3n#cite_note-2)​[3](https://es.wikipedia.org/wiki/Ciencias_de_la_computaci%C3%B3n#cite_note-3)​ El cuerpo de conocimiento de las ciencias de la computación es frecuentemente descrito como el estudio sistemático de los procesos algorítmicos que describen y transforman información: su teoría, análisis, diseño, eficiencia, implementación, algoritmos sistematizados y aplicación

La informática, ​ también llamada computación, ​ es el área de la ciencia que se encarga de estudiar la administración de métodos, técnicas y procesos con el fin de almacenar, procesar y transmitir información y datos en formato digital. La informática abarca desde disciplinas teóricas hasta disciplinas prácticas.

**Ramas de las ciencias de la computación**

* Desarrollo de algoritmos.
* Inteligencia artificial.
* **Ciencia** de datos.
* Ingeniería de software.
* Teoría de la **computación**.
* Programación.
* Robótica.
* Tecnologías de la información (TIC)

**Entre otros ejemplos, podemos nombrar:**

* Google Chrome.
* Windows Movie Maker.
* Audacity.
* Adobe Photoshop.
* MS Project.
* Avast.
* MSN Messenger.
* Paint.

### ****Gramáticas formales****

En [informática](https://es.wikipedia.org/wiki/Inform%C3%A1tica) una **gramática regular** es una [gramática formal](https://es.wikipedia.org/wiki/Gram%C3%A1tica_formal) (V, Σ, R, S) que puede ser clasificada como regular izquierda o regular derecha. Las gramáticas regulares sólo pueden generar a los [lenguajes regulares](https://es.wikipedia.org/wiki/Lenguaje_regular) de manera similar a los autómatas finitos y las expresiones regulares.

EJEMPLOS DE GRAMÁTICAS REGULARES:

            1.- V={S,A}, T={0,1}, las producciones son

S→0A

A→ 10A

A→

**Gramáticas recursivas**

En [informática](https://en-m-wikipedia-org.translate.goog/wiki/Computer_science?_x_tr_sl=en&_x_tr_tl=es&_x_tr_hl=es&_x_tr_pto=rq) , una [gramática](https://en-m-wikipedia-org.translate.goog/wiki/Formal_grammar?_x_tr_sl=en&_x_tr_tl=es&_x_tr_hl=es&_x_tr_pto=rq) se denomina informalmente **gramática recursiva** si contiene [reglas de producción](https://en-m-wikipedia-org.translate.goog/wiki/Formal_grammar?_x_tr_sl=en&_x_tr_tl=es&_x_tr_hl=es&_x_tr_pto=rq#The_syntax_of_grammars) que son [recursivas](https://en-m-wikipedia-org.translate.goog/wiki/Recursion_%28computer_science%29?_x_tr_sl=en&_x_tr_tl=es&_x_tr_hl=es&_x_tr_pto=rq) , lo que significa que expandir un no terminal de acuerdo con estas reglas puede eventualmente conducir a una cadena que incluya el mismo no terminal nuevamente. De lo contrario, se denomina **gramática no recursiva**

Una regla recursiva por la izquierda **se produce cuando una regla gramatical hace referencia a sí misma en su inicio, lo que crea bucles en el análisis** .

Por ejemplo, en "A -> A algo", la "A" no terminal aparece más a la izquierda.

* **Tokenización**: qué es y cómo se usa en el proceso de compilación.

La tokenización, cuando se aplica a la seguridad de los datos, es el proceso de sustitución de un elemento de datos confidenciales por un equivalente. La tokenización tiene una importancia enorme en el proceso de compilación de lenguajes de programación. **Actúa como un paso preliminar antes del análisis, que convierte los tokens en una representación estructurada, lo que permite la ejecución precisa de instrucciones**.

* **Expresiones regulares**: su función en el análisis léxico.

Las expresiones regulares **son una forma de especificar patrones, entendiendo por patrón la forma de describir cadenas de caracteres**. Es la forma de definir los tokens o componentes léxicos y, como veremos, cada patrón concuerda con una serie de cadenas.

* **Reconocimiento de palabras clave**: cómo se identifican en los lenguajes de programación.

es una parte de la [inteligencia artificial](https://es.wikipedia.org/wiki/Inteligencia_artificial) que consiste en identificar palabras específicas dentro de una locución

* **Identificación de tokens**: clasificación y ejemplos prácticos.
* La clasificación de tokens se refiere al **proceso de asignar una etiqueta o categoría a cada token o elemento en una secuencia de texto**



* **Tipos de análisis sintáctico**:
	+ Análisis sintáctico descendente. Se puede considerar el análisis sintáctico descendente como un **intento de encontrar una derivación por la izquierda para una cadena de entrada**
	+ Análisis sintáctico ascendente. **Es un tipo de análisis que construye el árbol sintáctico desde las hojas (la parte inferior) hasta la raíz (la parte superior**
* **Árboles de análisis sintáctico**: qué son. **es un** [árbol](https://en-m-wikipedia-org.translate.goog/wiki/Tree_%28data_structure%29?_x_tr_sl=en&_x_tr_tl=es&_x_tr_hl=es&_x_tr_pto=tc) ordenado y con raíz que representa la estructura [sintáctica de una](https://en-m-wikipedia-org.translate.goog/wiki/Syntax?_x_tr_sl=en&_x_tr_tl=es&_x_tr_hl=es&_x_tr_pto=tc) [cadena](https://en-m-wikipedia-org.translate.goog/wiki/String_%28computer_science%29?_x_tr_sl=en&_x_tr_tl=es&_x_tr_hl=es&_x_tr_pto=tc) de acuerdo con alguna [gramática independiente del contexto](https://en-m-wikipedia-org.translate.goog/wiki/Context-free_grammar?_x_tr_sl=en&_x_tr_tl=es&_x_tr_hl=es&_x_tr_pto=tc)
* y cómo ayudan en la estructura de los programas. Los árboles de análisis son útiles en varias aplicaciones, incluido el procesamiento del lenguaje, el diseño del compilador y el análisis de sintaxis. **Proporcionan una representación estructural de la cadena de entrada, lo que permite el análisis y la manipulación de sus propiedades sintácticas.**



### ****Algoritmos de análisis de cadenas****

* **Algoritmo de Boyer-Moore**: concepto, funcionamiento y su aplicación en la búsqueda de cadenas dentro de textos o programas.

CONCEPTO: El **algoritmo de Boyer**-**Moore** utiliza información recopilada durante el paso de preprocesamiento para omitir secciones del texto, lo que da como resultado un factor constante más bajo que muchos otros **algoritmos** de búsqueda de cadenas.

FUNCIONAMIENTO: 