EDUCACIÓN
SECRETARÍA DE EDUCACIÓN

TECNOLÓGICO NACIONAL DE MÉXICO

INSTITUTO TECNOLÓGICO DE PACHUCA

NOMBRE DE LA CARRERA:

INGENIERÍA EN SISTEMAS COMPUTACIONALES

GRUPO: B

MATERIA: LENGUAJES Y AUTÓMATAS I

ING. BAUME LAZCANO RODOLFO

TAREA 2.1

NOMBRE:

CITLALI MARTÍNEZ SÁNCHEZ 21200614

FECHA: 18 DE MARZO DE 2024

EDUCACIÓN
SECRETARÍA DE EDUCACIÓN

TECNOLÓGICO NACIONAL DE MÉXICO

INTRODUCCIÓN

Las expresiones regulares son una herramienta importante para trabajar con texto de manera eficiente y precisa en una amplia gama de aplicaciones informáticas, desde el desarrollo de software hasta la administración de sistemas. Su capacidad para definir y buscar patrones en cadenas de texto las convierte en una herramienta indispensable para cualquier persona que trabaje con datos textuales en el ámbito informático.

Su importancia radica en su versatilidad y eficiencia para realizar tareas relacionadas con el procesamiento de texto, como validar datos de entrada, buscar y reemplazar cadenas, analizar archivos de texto, y mucho más. Permiten realizar búsquedas avanzadas y manipular cadenas de manera precisa y flexible, lo que las convierte en una herramienta fundamental en el desarrollo de software y en la administración de sistemas.

En esta investigación veremos la que son las expresiones regulares, su importancia y los casos de uso de esta.

EXPRESIONES REGULARES

¿QUÉ SON LAS EXPRESIONES REGULARES?

Una expresión regular es una forma de representar los lenguajes regulares, y se construye utilizando caracteres del alfabeto sobre el cual se define el lenguaje.
Las expresiones regulares son una relajación en la notación de lenguajes regulares. El objetivo es hacer más visible el patrón de concatenación, para lograr esto se omiten las llaves de la notación de conjuntos y se cambia el símbolo de la unión por un símbolo +.
En el contexto de la teoría de la computación y los autómatas, son patrones formales que describen conjuntos de cadenas de texto. Estos patrones se utilizan para buscar, filtrar o manipular cadenas de texto de manera eficiente y precisa. La importancia de las expresiones regulares radica en su capacidad para representar patrones complejos de una manera concisa, lo que las hace indispensables en una amplia gama de aplicaciones informáticas.
En un sentido más formal, las expresiones regulares pueden definirse como un tipo de gramática formal que describe un lenguaje regular, es decir, un conjunto de cadenas de texto que pueden ser reconocidas por un autómata finito determinista (AFD) o un autómata finito no determinista (AFND). Desde un punto de vista práctico, las expresiones regulares se componen de caracteres literales (como letras y números) y metacaracteres (como "*", ".", "^", "$"), que representan clases de caracteres, repeticiones, posiciones específicas en una cadena, entre otros.

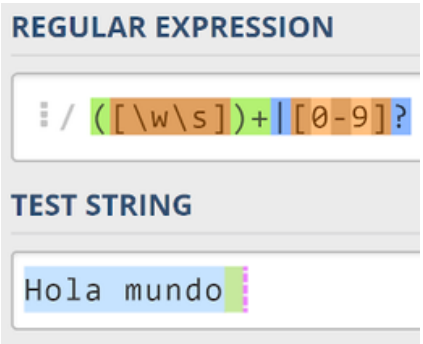
Las expresiones regulares están estrechamente relacionadas con los autómatas finitos no deterministas y pueden considerarse una alternativa, que el usuario puede comprender fácilmente, a la notación de los AFN para describir componentes de software. Por tanto, las expresiones regulares sirven como lenguaje de entrada de muchos sistemas que procesan cadenas. Algunos ejemplos son los siguientes:

- Comandos de búsqueda tales como el comando grep de UNIX o comandos equivalentes para localizar cadenas en los exploradores web o en los sistemas de formateo de texto.
- Generadores de analizadores léxicos, como Lex o Flex. Recuerde que un analizador léxico es el componente de un compilador que divide el programa fuente en unidades lógicas o sintácticas formadas por uno o más caracteres que tienen un significado.

IMPORTANCIA

las expresiones regulares desempeñan un papel fundamental en el ámbito de los autómatas y la teoría de la computación al proporcionar una forma poderosa y eficiente de representar y manipular patrones en cadenas de texto.
En pocas palabras las expresiones regulares son de vital importancia en el contexto de los autómatas y la teoría de la computación por varias razones fundamentales:

Representación de patrones: Las expresiones regulares proporcionan una forma compacta y precisa de representar patrones en cadenas de texto. Esto es esencial para describir conjuntos de cadenas que siguen ciertas reglas o formatos, lo que facilita la especificación de los comportamientos que se desean reconocer o manipular en un autómata.
Flexibilidad y concisión: Las expresiones regulares permiten expresar patrones complejos de manera concisa y elegante. Esto es crucial en la implementación de algoritmos para la búsqueda, manipulación y análisis de texto, ya que reduce la complejidad y la cantidad de código necesario para lograr el mismo resultado.
Eficiencia computacional: La capacidad de representar patrones de manera concisa y eficiente hace que las expresiones regulares sean una herramienta poderosa en términos de rendimiento computacional. Esto es especialmente importante en aplicaciones donde el tiempo de ejecución y el uso de recursos son críticos, como en motores de búsqueda, sistemas de filtrado de correo electrónico y análisis de registros.
Aplicaciones en la industria: Las expresiones regulares se utilizan en una amplia gama de aplicaciones industriales, incluyendo la construcción de compiladores, analizadores léxicos, procesadores de texto, sistemas de búsqueda y filtrado de texto, análisis de datos y más. Su versatilidad las hace indispensables en el desarrollo de software y en la administración de sistemas informáticos.
Abstracción de patrones comunes: Las expresiones regulares proporcionan una forma estándar y portátil de representar patrones comunes en cadenas de texto, lo que facilita la reutilización y el intercambio de algoritmos y técnicas entre diferentes aplicaciones y plataformas.



EDUCACIÓN
SECRETARÍA DE EDUCACIÓN

REGLAS SINTÁCTICAS SON VÁLIDAS PARA LAS EXPRESIONES REGULARES

Las expresiones regulares se pueden aplicar **en diversos lenguajes**, tales como Perl, Python, Ruby, JavaScript, XML o HTML, por lo que los usos o funciones pueden llegar a ser muy diferentes. En JavaScript los patrones regex se utilizan, por ejemplo, en los métodos de cadena search(), match() o replace(), mientras que las expresiones en documentos XML sirven para limitar elementos de contenido. En lo que respecta a la **sintaxis**, entre los diferentes lenguajes de programación y lenguajes de marcado **apenas hay diferencias** en cuanto a las expresiones regulares:

Caracteres especiales regex de sintaxis	Función
[]	Los corchetes identifican a una clase de caracteres que siempre representa a un único carácter en un patrón de búsqueda.
()	Los paréntesis identifican un grupo de caracteres formado por uno o varios caracteres y que pueden operarse unos dentro de los otros.
-	Funciona a modo de especificación del área (de [...] hasta [...]) cuando se sitúa entre dos caracteres normales.
^	Limita la búsqueda al inicio de una línea (otra función: elemento de negación en clases de caracteres).
$	Limita la búsqueda al final de una línea.
.	Equivale a cualquier carácter.
*	El número del carácter, de la clase o del grupo situado antes del asterisco puede ser aleatorio (cero incluido).
+	El carácter, la clase o el grupo antes de un signo más debe aparecer como mínimo una vez.
?	El carácter, la clase o el grupo antes del signo de interrogación es opcional y puede aparecer como máximo una vez.
{n}	El carácter, la clase o el grupo anteriores aparecen exactamente n veces.
{n,m}	El carácter, la clase o el grupo anteriores aparecen como mínimo n veces y como máximo m veces.
{n,}	El carácter, la clase o el grupo anteriores aparecen como mínimo n veces o con frecuencia.
\b	Tiene en cuenta el límite de palabra durante la búsqueda.
\B	Ignora el límite de palabra durante la búsqueda.
\d	Cualquier dígito; abreviatura para la clase de caracteres [0-9].
\D	Cualquier no dígito; abreviatura para la clase de caracteres [^0-9].
\w	Cualquier carácter alfanumérico; abreviatura para la clase de caracteres [a-zA-Z_0-9].
\W	Cualquier carácter no alfanumérico; abreviatura para la clase de caracteres [^\w].

CASOS DE USO

- Las expresiones regulares permiten especificar patrones de texto que pueden ser reconocidos por autómatas. Estos patrones pueden representar desde cadenas simples hasta estructuras más complejas, como números de teléfono, direcciones de correo electrónico o formatos de fecha.
- Construcción de compiladores, las expresiones regulares se utilizan en la fase de análisis léxico para reconocer los componentes básicos del lenguaje, como identificadores, palabras clave, operadores y literales. Esto es fundamental para descomponer el código fuente en una secuencia de tokens que pueden ser procesados por etapas posteriores del compilador.
- Pueden utilizar expresiones regulares para buscar patrones específicos dentro de grandes conjuntos de datos. Esto es útil en aplicaciones como motores de búsqueda, análisis de registros y sistemas de monitorización, donde se necesita identificar y extraer información relevante de manera eficiente.
- También se utilizan para realizar transformaciones en el texto, como la eliminación de caracteres no deseados, la normalización de formatos y la extracción de información estructurada. Esto es útil en aplicaciones de procesamiento de texto y análisis lingüístico, donde se necesita manipular cadenas de texto de manera eficiente.

Expresión Regular	Equivalencia
a.b	axb aab abb aSb a#b ...
a..b	axxb aaab abbb a4$b ...
[abc]	a b c (cadenas de un caracter)
[aA]	a A (cadenas de un caracter)
[aA][bB]	ab Ab aB AB (cadenas de dos caracteres)
[0123456789]	0 1 2 3 4 5 6 7 8 9
[0-9]	0 1 2 3 4 5 6 7 8 9
[A-Za-z]	A B C ... Z a b c ... z
[0-9][0-9][0-9]	000 001 .. 009 010 .. 019 100 .. 999
[0-9]*	cadena_vacía 0 1 9 00 99 123 456 999 9999 ...
[0-9][0-9]*	0 1 9 00 99 123 456 999 9999 99999 99999999 ...
^.*$	cualquier línea completa

CONCLUSIÓN

Las expresiones regulares son una herramienta fundamental en el ámbito de los autómatas, permitiendo reconocer, manipular y analizar patrones en cadenas de texto de manera eficiente y precisa.
Estas son un componente esencial e importante en una amplia gama de aplicaciones informáticas, incluyendo la construcción de compiladores, el análisis léxico, la búsqueda y filtrado de texto, la validación de datos de entrada, y la transformación de texto.

¿Qué hace?	Expresión Regular	Ejemplo de resultado
Palabras que llevan tilde, el filtro es sensible a las tildes!	c[óo]mo	cómo y como
Tipicas preguntas	(cu[aá]ndo)|(d[oó]nde)|(c[óo]mo)	Cualquiera de esas KWs
Una única palabra	^([^\s]+)$	palabra1 pero no palabra1 palabra2
Dos palabras	^(\S+)\s\S+$	palabra1 palabra2
Tres palabras	^(\S+)\s\S(\S+)\s\S{1,}$	palabra1 palabra2 palabra3
Cuatro palabras	^(\S+)\s\S(\S+)\s\S\s\S+$	palabra1 palabra2 palabra3 palabra4
Al menos una palabra dos caracteres y toda la frase teremina en dos digitos	^[a-z]{2,}\s\d\d$	palabra1 47

Bibliografía

Expresiones regulares. (s. f.). Lenguajes Formales y Autómatas. https://ivanvladimir.gitlab.io/lfya_book/docs/02lam%C3%A1quinasinmemoria/03expresionesregulares/

IBM documentation. (s. f.). https://www.ibm.com/docs/es/i/7.3?topic=expressions-regular

Equipo editorial de IONOS. (2019, 30 diciembre). Regex o expresiones regulares: la manera más sencilla de describir secuencias de caracteres. IONOS Digital Guide. https://www.ionos.mx/digitalguide/paginas-web/creacion-de-paginas-web/regex/

Uruñuela, L. (2021, 8 julio). Expresiones regulares para SEO. Mecagoenlos. https://www.mecagoenlos.com/Posicionamiento/expresiones-regulares-en-google.php

Equipo editorial de IONOS. (2019b, diciembre 30). Regex o expresiones regulares: la manera más sencilla de describir secuencias de caracteres. IONOS Digital Guide. https://www.ionos.mx/digitalguide/paginas-web/creacion-de-paginas-web/regex/