Universidad Autónoma de Chiapas
Facultad de contaduría y administración C-I

Carrera:
Lic. En ing en desarrollo y tecnologías de software.

Nombre del alumno:
González Aguilar Eduardo - A211154

Semestre: 6. **Grupo:** M.

Nombre de la actividad:
Conceptos.

Fecha de entrega:
07/08/2023.

Definición de expresión regular.

Expresiones Regulares (Regular Expresión) o simplemente conocidas como REGEX son cadenas de caracteres que forman patrones de búsqueda o de sustitución. El uso de REGEX está ampliamente difundido debido a su ventaja para el manejo de cadenas en lenguajes como Java, JavaScript, C++11, Python y PERL.

Definición de expresión regular.

Operadores aritméticos

Transforman y manipulan datos, principalmente cuando son números (tanto con decimales como sin decimales).

Se pueden distinguir de dos tipos de operadores aritméticos: los unarios y los binarios.

Operadores unarios

Se llaman así porque sólo necesitan una única variable o argumento para poder calcular (recuerda, transformar) su valor.

- La **negación**. Se utiliza la palabra reservada `not`, o también el símbolo `!`.
- El **negativo**. Con el símbolo `-`, si ponemos `-a` devuelve el valor negativo de la variable. ¡No cambia el valor que tiene!
- El **positivo**. También se puede escribir `+a`.
- Operador de **incremento**. Sirve para aumentar una variable en uno su valor (cambia su valor), su símbolo es `++`. Si ponemos `a++`, y `a` valía **2**, ahora valdrá **3**.
- Operador de **decremento**. Lo mismo que el anterior, pero para reducir en **1**. Su símbolo es `--`.

Operadores binarios

Los operadores binarios necesitan dos variables o argumentos. Ejemplos típicos de todos los lenguajes de programación:

- La **suma**. Con el símbolo `+`.
- La **resta**. Su símbolo es el `-`.
- La **multiplicación**. El símbolo es `*`.
- La **división**. La mayoría de lenguajes usan el símbolo `/`.
- La **división entera**. La diferencia con la anterior es que no devuelve decimales. Su símbolo suele ser la palabra `div`. Por ejemplo `5 / 2 = 2,5`, pero `5 div 2 = 2`.

- El **resto** o **módulo**. Este operador devuelve el resto de dos números. Suele usarse el símbolo `%` o la palabra `mod`. `5 mod 2 = 1`.

Hay lenguajes que usan el símbolo + para juntar cadenas. Por ejemplo, en Java `'Hola ' + 'Mundo'` sería `'Hola Mundo'`.

Operadores booleanos

Sirven para combinar `true` y `false`, así que sólo puede devolver uno de estos valores.

Operadores de asignación

Un operador de asignación da un valor a una variable. Si la variable tenía un valor, se sustituye por el nuevo. Ejemplo:

```
variable = 4;
```

Ahora variable almacena el número 4. Si tenía un valor antes se ha perdido. También podemos asignar el valor de otra variable:

```
variable = var_2;
```

O incluso de una expresión:

```
variable = 2/5;
```

Veamos una lista de operadores de asignación.

- `=`: Para asignar una variable.
- `+=`: Para acumular una variable. Por ejemplo `variable+=2` añade `2` a lo que ya tenía en esa variable. Si la variable valía **5**, ahora vale **3**. Sería lo mismo que escribir `variable = variable + 2`.
- `-=`: Para decrementar o restar el valor de la variable. `variable = variable - 2`.
- `*=`: Para multiplicar su valor. `variable = variable * 2`.
- `/=`: Para dividir su valor. `variable = variable / 2`.

Operadores relacionales

Con los operadores lógicos podemos comparar dos valores, diciéndonos si el resultado es **verdadero** o **falso**.

Como habrás deducido, este tipo de operador devuelve los valores `true` o `false`. Por ejemplo, la siguiente comparación:

```
5 > 2
```

nos devuelve `true`. En cambio:

```
5 < 2
```

nos devuelve `false`. Recuerda que puedes usar variables y expresiones también, como los siguientes ejemplos:

```
a < b
a < b + 2
```

Los operadores relacionales son:

- `==`: Igual.
- `!=` o `<>`: Distinto.
- `<`: Menor.
- `<=`: Menor o igual.
- `>`: Mayor.
- `>=`: Mayor o igual.

Hemos visto ejemplos con números, ya sean números reales o números enteros, pero también es posible usar los operadores relaciones con otros tipos de datos, como puede ser las cadenas de texto.

Operadores lógicos

Hemos comentado que los operadores relacionales sirve para evaluar una expresión. Si quieres evaluar varias operaciones a la vez, debes juntarlas con **operadores lógicos**.

Estos operadores lógicos puedes encontrarlos con la palabra `and` y `&&`, depende del lenguaje, y también el operador `or` o `||`.

Con `&&` te devuelve verdadero si las dos expresiones son verdaderas, si uno es falsa ya devuelve falsa. Por ejemplo:

```
a > b && b > c
```

Con `||` devuelve verdadero si sólo uno de los dos valores lo es.

Al estar hablando de expresiones booleanas, también te puedes encontrar como operador lógico el símbolo `!`, que ya hemos comentado que sirve como negación, así que pasaría de `true` a `false`, o de `false` a `true`.

Operadores condicionales

En algunos lenguajes de programación, puedes encontrarte con la posibilidad de usar los operadores condicionales.

Es posible que conozcas la estructura condicional `if()...else()`, que devuelve un valor si es verdadero, y otro valor si es false.

Pues es posible usar un **operador ternario** (porque necesitas tres operadores), que tienen esta sintaxis:

```
resultado = x > 10 ? true: false
```

Si la condición `x > 10` se cumple, la variable resultado valdrá `true`, y sino se cumple, valdrá `false`.

Orden de operaciones en informática

Aunque parece que hemos hablado de muchos **tipos de operadores**, seguramente la gran mayoría ya los conoces, y con un poco de práctica sabrás usarlos correctamente.

Para finalizar, comentar que existe un orden de prioridad a la hora de usar los operadores.

Si tienes dudas, usa siempre paréntesis para establecer la prioridad. Por defecto, la mayoría de lenguajes utiliza esta regla:

1. Primero los **paréntesis**.
2. Luego la **negación**.
3. **Multiplicar**, **dividir** y **resto** de división.
4. **Sumas** y **restas**.
5. Y si tienen igualdad de prioridad, de izquierda a derecha.

Conversión de DFA a expresiones regulares.

1. Convertir el DFA en un NFA.
2. Convertir el NFA en una expresión regular.

El primer paso se puede realizar utilizando el algoritmo de construcción de subconjuntos. El segundo paso se puede realizar utilizando el algoritmo de eliminación de estados.

Leyes algebraicas de expresiones regulares.

Las expresiones regulares tienen un conjunto de leyes algebraicas que se pueden utilizar para simplificarlas[1]:

- Ley conmutativa para la unión: L+M = M+L.
- Ley asociativa para la unión (L+M) + N: L+ (M+N).
- Ley asociativa para la concatenación: (LM)N = L(MN).
- Elemento identidad y Elemento nulo.

¿Qué es un token?

Es una secuencia de caracteres que actúa individualmente dentro del patrón, representa un elemento particular que trata de coincidir o analizar.

¿Qué es un lexema?

Es una secuencia específica de caracteres en el código fuente que se empareja con una regla definida por una expresión regular.

¿Qué es un patrón?

Descripción formal que define una secuencia de caracteres. La cual compara las cadenas de texto, y se puede utilizar para buscar, extraer, reemplazar o manipular texto.