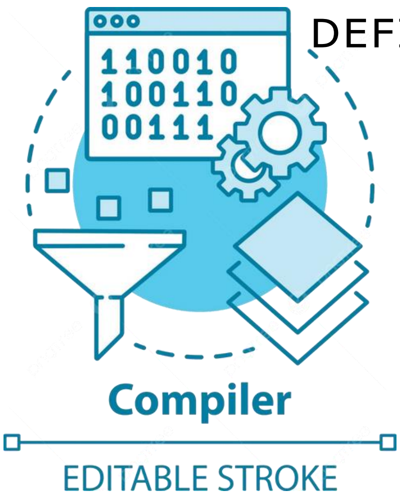
COMPILADORES



DEFINE LOS SIGUIENTES CONCEPTOS Y REALIZAR LOS EJERCICIOS.

ACTIVIDAD 1

LEONARDO MALDONADO GALVEZ
6-M

Definir el concepto de expresión regular.

I.- Explicar los tipos de operadores de expresiones regulares.

Ejemplos básicos:

- **[A-Za-z0-9]+$**: Permite letras (mayúsculas y minúsculas) y dígitos.
- **z{3}**: Requiere exactamente tres "z" consecutivas (ej. "zzz").
- **\d**: Coincide solo con números.
- **go*gle**: La "o" puede aparecer 0 o más veces (ej. "ggle", "gogle", "google").
- **go+gle**: La "o" debe aparecer al menos una vez (ej. "gogle", "google").
- **gray|grey**: Coincide con "gray" o "grey".
- **colou?r**: La "u" es opcional (ej. "color", "colour").

Escapes de caracteres:

- **\t**: Coincide con una tabulación.
- **\n**: Coincide con un salto de línea.
- **\r**: Coincide con un retorno de carro.
- **\d**: Coincide con dígitos.

Clases de caracteres:

- **[grupo_caracteres]**: Coincide con cualquier carácter en el grupo.
- **[^grupo_caracteres]**: Coincide con cualquier carácter que no esté en el grupo.
- **[Primero-Último]**: Coincide con cualquier carácter en el rango.
- **.**: Coincide con cualquier carácter excepto \n.
- **\w**: Coincide con cualquier carácter de palabra.
- **\W**: Coincide con cualquier carácter no perteneciente a una palabra.
- **\s**: Coincide con cualquier espacio en blanco.
- **\S**: Coincide con cualquier carácter que no sea un espacio en blanco.
- **\d**: Coincide con cualquier dígito decimal.
- **\D**: Coincide con cualquier carácter no numérico.

Delimitadores:

- **^**: Coincide con el principio de la cadena o línea.
- **$**: Coincide con el final de la cadena o línea.
- **\A**: Coincide con el principio de la cadena.
- **\Z**: Coincide con el final de la cadena o antes de \n.

- **\z**: Coincide exactamente al final de la cadena.
- **\G**: Coincide donde terminó la última coincidencia.
- **\b**: Coincide en un límite entre un carácter alfanumérico y uno no alfanumérico.
- **\B**: No coincide en un límite de palabra.

Agrupamientos:

- **(subexpresión)**: Captura la subexpresión coincidente y le asigna un número.
- **(?=subexpresión)**: Búsqueda anticipada positiva.
- **(?<=subexpresión)**: Búsqueda retrospectiva positiva.
- **(?<!subexpresión)**: Búsqueda retrospectiva negativa.

Cuantificadores:

- *****: Coincide con el elemento anterior 0 o más veces.
- **+**: Coincide con el elemento anterior 1 o más veces.
- **?**: Coincide con el elemento anterior 0 o 1 vez.
- **{N}**: Coincide exactamente con N repeticiones.
- **{N,}**: Coincide con al menos N repeticiones.
- **{N,M}**: Coincide con al menos N repeticiones, pero no más de M.

II.- Explicar el proceso de conversión de DFA a expresiones regulares.

El proceso de conversión de un Autómata Determinista Finito (DFA, por sus siglas en inglés) a una expresión regular implica transformar el autómata en una representación algebraica que describe el mismo lenguaje que reconoce el DFA. Aquí te explico los pasos básicos del proceso:

1. Definir el DFA:

- Un DFA se define por un conjunto de estados, un alfabeto de entrada, una función de transición, un estado inicial y un conjunto de estados finales.

2. Crear una Tabla de Transiciones:

- Se utiliza una tabla para representar las transiciones entre los estados del DFA para cada símbolo del alfabeto.

3. Introducir Estados Inicial y Final Únicos:

- Si el DFA tiene múltiples estados finales, se agrega un nuevo estado final único y se crean transiciones epsilon (que no consumen símbolos) desde cada uno de los

antiguos estados finales a este nuevo estado.
- Si el estado inicial tiene transiciones entrantes, se agrega un nuevo estado inicial con transiciones epsilon hacia el estado inicial original.

4. Eliminar Estados Intermedios:

- Se eliminan los estados del DFA uno por uno, empezando por aquellos que no son iniciales ni finales.
- Al eliminar un estado, se ajustan las transiciones entre los estados restantes para que el lenguaje reconocido no cambie.
- Esto se hace combinando las transiciones entrantes y salientes del estado eliminado mediante expresiones regulares que representan estas combinaciones.

5. Formar la Expresión Regular:

- Una vez que solo quedan el estado inicial y el estado final, la expresión regular que describe el lenguaje aceptado por el DFA se deriva de las transiciones entre estos dos estados.

6. Simplificación (Opcional):

- La expresión regular obtenida puede simplificarse utilizando reglas algebraicas para que sea más concisa y fácil de interpretar.

Este proceso es sistemático y garantiza que la expresión regular obtenida describa exactamente el mismo lenguaje que reconocía el DFA original.

III.- Explicar leyes algebraicas de expresiones regulares.

1. Ley de Identidad:

- **Epsilon (ε)**: ε representa la cadena vacía.
- **Rε = εR = R**: Concatenar cualquier expresión regular con ε (la cadena vacía) no cambia la expresión.
- **RØ = ØR = Ø**: Concatenar cualquier expresión regular con el conjunto vacío (Ø) siempre da como resultado el conjunto vacío.

2. Ley de Conmutatividad (Solo para unión):

- **R | S = S | R**: Cambiar el orden en una unión no altera el resultado. Por ejemplo, a | b es lo mismo que b | a.

3. Ley de Asociatividad:

- **Unión:** R | (S | T) = (R | S) | T (El orden en que se agrupan no afecta la unión).
- **Concatenación:** (RS)T = R(ST) (El orden en que se agrupan no afecta la concatenación).

4. Ley de Distributividad:

- **R(S | T) = RS | RT**: Una expresión regular que concatena con una unión puede distribuirse.
- **(S | T)R = SR | TR**: Esto es válido tanto si la concatenación ocurre antes como después.

5. Ley de Idempotencia:

- **R | R = R**: Unir una expresión regular consigo misma no la cambia.
- **R*R* = R***: La cerradura de Kleene aplicada dos veces es lo mismo que aplicarla una vez.

6. Ley de Anulabilidad:

- **RØ = ØR = Ø**: Cualquier expresión regular multiplicada (concatenada) con el conjunto vacío (Ø) sigue siendo Ø.
- **R | Ø = R**: Unir una expresión regular con el conjunto vacío no cambia la expresión.

7. Ley de Cerradura de Kleene:

- **R* = ε | RR***: La cerradura de Kleene de una expresión regular R incluye la cadena vacía y cualquier número de repeticiones de R.

8. Ley de Absorción:

- **R | RS = R**: Si puedes generar todas las cadenas de RS usando R, entonces unir R con RS da como resultado solo R.
- **R | R* = R***: Unir una expresión regular con su cerradura de Kleene da como resultado la cerradura de Kleene.

Estas leyes son útiles para simplificar expresiones regulares y encontrar formas más eficientes de escribir patrones de búsqueda.