

I. Explicar los tipos de operadores de expresiones regulares:

Expresión regular: Una expresión regular es una secuencia de caracteres que define un patrón de búsqueda. Se utiliza para realizar operaciones de búsqueda y manipulación de cadenas de texto, permitiendo encontrar patrones específicos dentro de ellas.

Tipos de operadores de expresiones regulares:

Concatenación: Representada por la yuxtaposición de dos expresiones. Por ejemplo, si se tiene la expresión regular AB, esta buscará la secuencia de caracteres "AB" en el texto.

Alternancia (o |): Permite especificar alternativas entre dos expresiones. Por ejemplo, la expresión regular A|B buscará tanto "A" como "B" en el texto.

***Cierre (o *):** Indica que la expresión anterior puede aparecer cero o más veces. Por ejemplo, la expresión regular A* buscará la secuencia "A", "AA", "AAA", etc.

Opción (o ?): Indica que la expresión anterior puede aparecer cero o una vez. Por ejemplo, la expresión regular A? buscará tanto "A" como la cadena vacía.

Agrupación (o paréntesis): Permite agrupar expresiones para aplicar operadores sobre ellas como un todo. Por ejemplo, (AB)* buscará la repetición de la secuencia "AB".

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

DFA (Autómata Finito Determinista): Un DFA es un modelo matemático que describe un sistema que puede estar en uno de un número finito de estados, y que cambia de un estado a otro en respuesta a una entrada específica.

Proceso de conversión de DFA a expresiones regulares:

Eliminar estados inalcanzables: Eliminar los estados que no son accesibles desde el estado inicial.

Eliminar estados muertos: Eliminar los estados que no tienen transiciones a otros estados.

Reducción del DFA: Simplificar el DFA mediante la combinación de estados equivalentes.

Asignación de expresiones regulares a transiciones: Asignar una expresión regular a cada transición del DFA, representando la cadena de entrada que lleva de un estado a otro.

Eliminación de estados intermedios: Reemplazar los estados intermedios con expresiones regulares que representen las rutas posibles entre los estados.

Al final de este proceso, se obtiene una expresión regular que describe el lenguaje aceptado por el DFA original.

III. Explicar leyes algebraicas de expresiones regulares:

Las leyes algebraicas de expresiones regulares son reglas y propiedades que facilitan la manipulación y simplificación de expresiones regulares. Algunas leyes algebraicas comunes incluyen:

1. **Ley de la identidad:** $E + O = E$
2. **Ley del Aniquilador:** $E * O = O$
3. **Ley del Cero:** $E^* \times O = O$
4. **Ley del Absorbente:** $E + E * F = E$
5. **Ley de la Unidad:** $E + \epsilon * E = E$

Estas leyes permiten simplificar expresiones regulares, combinarlas de manera eficiente y entender las propiedades algebraicas que rigen la manipulación de patrones en texto.

1. Realice una expresión regular de todas las cadenas con símbolos a y b, que terminan con el sufijo abb. Ejemplos de estas cadenas con: abb, aabb, babb, aaabb, ababb, baabb, bbab, etc:

R: $^{[ab]^*abb\$}$

Se indica que **a** y **b** puedan repetirse, con condición de que el final debe ser **abb**

2. Realice una expresión regular de todas las cadenas de con símbolos 0 y 1, que primero tengan los símbolos 1's con longitud impar y después aparezcan los 0's con longitud par. Ejemplo de éstas cadenas son: 100, 10000, 1000000, 11100, 1110000, 111110000.

R: $^{(1(11)^*+(00)^*\$}$

1. Para la expresión regular $([+|-]?)\d+\.\d+\$$ Indique las cadenas correctas de los siguientes incisos (Nota. En esta expresión el "." es un símbolo no el operador concatenación y "d" representa los dígitos del 0 al 9). Los incisos a revisar son los siguientes:

- a) -20.43 = **Cumple.**
- b) 0.3216 = **Cumple.**
- c) 329 = **No cumple.**
- d) 217.92 = **Cumple.**
- e) +2019 = **No cumple.**
- f) +.762 = **No cumple.**
- g) -.4555 = **No cumple.**