

UNIVERSIDAD AUTONOMA DE CHIAPAS

Facultad de contaduría y administración | Campus 1

Diego González Carpio
Compiladores

Diego Gonzalez Carpio

08/Agosto

Tarea 1

➤ Ejemplo de Expresión Regular

$([a-z]\{2,3}[0-9]\{3,5\})$

- Dos o mas letras seguidas por 3 a 5 numeros.

➤ Definir los conceptos:

Expresión Regular, tipos de Operadores de expresiones regulares,
Proceso de conversión de DFA a expresiones regulares,
Explicar las leyes algebraicas de expresiones regulares.
También: Tokens / Lexema / Patron.

Expresiones Regulares:

Podemos definir como expresion regular a (regular Expression) como una cadena de texto generica, que se utiliza a modo de patron y que sirve para localizar trozos de texto dentro de otro mayor.

Tipos de Operadores

* Operador de cuantificación

- '*' : coincide con 0 o mas repeticiones del elemento anterior
- '+' : coincide con una o mas repeticiones del elemento anterior
- '?' : coincide con 0 o una repeticion del elemento anterior
- '{n}' : coincide con n repeticiones del elemento anterior
- '{n,}' : coincide con al menos n repeticiones del elemento anterior
- '{n,m}' : coincide con entre n y m repeticiones del elemento anterior

* Operadores de agrupación y alternancia:

- '(')' : Agrupa elementos y permite aplicar operadores al conjunto
- '\' : Escapa un caracter especial, permitiendo que se interprete literalmente

* Operadores de conjuntos y negación

- '[']' : Define un conjunto de caracteres posibles para una posición en la cadena
- '^' : Coincide con cualquier caracter que no este en el conjunto negado

* Anclajes y limites

- '^' : Coincide con el inicio de una linea o cadena
- '\$' : Coincide con el final de una linea o cadena

Proceso de conversión de DFA a expresión Regular

Este implica traducir el diagrama en un patrón de búsqueda que pueda ser utilizado para reconocer y extraer ciertas cadenas de texto.

1. Entender el diagrama de flujo de Automatización (FDA) ya que es crucial para traducir correctamente la lógica del diagrama en expresión Regular.
2. Identificar patrones, ya que estos pueden contener palabras clave, números, formatos específicos, etc.
3. Mapeo de símbolos a operadores; Asigna símbolos y estados del diagrama a los operadores y elementos correspondientes en expresiones regulares.
4. Traducción de lógica de las transiciones del diagrama en expresiones regulares.
5. Crea grupos y capturas. Utilizando parentesis para agrupar partes de la expresión regular.
6. Añadir metacaracteres de escape ('\') para tratar los caracteres especiales que deben interpretarse literalmente.
7. Pruebas y Ajustes.
8. Documentación.

Lejes Algebraicas de las expresiones Regulares.

Son las reglas que se pueden aplicar para simplificar y manipular las expresiones regulares.

1. Ley de Identidad: ' $r + \epsilon = r$ '
Cualquier expresión regular ' r ' unida a la expresión de cadena vacía (' ϵ ') es igual a ' r ' sola.
2. Ley de Anulación: ' $r \cdot \emptyset = \emptyset$ '
La concatenación de cualquier expresión regular ' r ' con la expresión regular vacía (' \emptyset ') resulta en la expresión vacía.
3. Ley de Cierre: ' $r^* = \epsilon + r \cdot r^*$ '
El cierre de Kleene de una expresión regular ' r ' es igual a la expresión vacía (' ϵ ') más la concatenación de ' r ' con el cierre de Kleene de ' r '.
4. Ley de Absorción: ' $r + r^* = r^*$ '
La unión de una expresión regular ' r ' con su cierre de Kleene ' r^* ' es igual al cierre de Kleene ' r^* '.

5- Ley de Distribución: $r \cdot (s + t) = r \cdot s + r \cdot t$

La concatenación de una expresión regular r con la unión de expresiones regulares s y t es igual a la unión de concatenaciones $r \cdot s$ y $r \cdot t$.

6- Ley de Complemento: $\neg(\neg r) = r$

La negación doble de una expresión regular es r es igual a r misma.

Token | Lexema | Patron

Token: es una secuencia de caracteres que representan un elemento lexico o un componente basico del lenguaje. Los tokens son unidades indivisibles que se utilizan en el proceso de analisis lexico para identificar partes de un programa o texto.

Lexema: Es una secuencia de caracteres en el codigo fuente que corresponde a un solo token. Es la representación textual exacta de un elemento lexico en el codigo.

Patron: Es una descripción de una secuencia de caracteres que define una clase de cadenas que se espera encontrar en un texto. Estos se utilizan en las expresiones regulares y en el analisis lexico para buscar y coincidir con fragmentos específicos de texto. Estos pueden tener caracteres especiales, literales y operadores especiales que permiten definir reglas para buscar coincidencias.

Tokens: P. Reservada, op

Lexemas \rightarrow Printf