

**UNIVERSIDAD AUTÓNOMA DE CHIAPAS
FACULTAD DE CONTADURÍA Y ADMINISTRACIÓN**



**LICENCIATURA EN INGENIERÍA EN DESARROLLO
Y TECNOLOGÍAS DE SOFTWARE**

**CATEDRÁTICO EN TURNO:
DR. LUIS GUTIERREZ ALFARO**

**MATERIA A CURSAR:
COMPILADORES**

• VÁZQUEZ MORALES BRIAN ISAAC

SEXTO SEMESTRE GRUPO “M”

**“DEFINE LOS SIGUIENTES CONCEPTOS Y REALIZAR LOS
EJERCICIOS.- ACTIVIDAD I, II.”**

TUXTLA GUTIÉRREZ, CHIAPAS

25 DE ENERO DEL AÑO 2024

Contenido

- DEFINIR EL CONCEPTO DE EXPRESIÓN REGULAR.3
- EXPLICAR LOS TIPOS DE OPERADORES DE EXPRESIONES REGULARES..... 3
- EXPLICAR EL PROCESO DE CONVERSIÓN DE DFA A EXPRESIONES REGULARES. 3
- EXPLICAR LEYES ALGEBRAICAS DE EXPRESIONES REGULARES. 4
- EJEMPLO 4
- BIBLIOGRAFIA5
- ANEXO DE IMAGEN5

DEFINIR EL CONCEPTO DE EXPRESIÓN REGULAR.

Las expresiones regulares son un equivalente algebraico para un autómata. Utilizado en muchos lugares como un lenguaje para describir patrones en texto que son sencillos pero muy útiles. Pueden definir exactamente los mismos lenguajes que los autómatas pueden describir: Lenguajes regulares.

EXPLICAR LOS TIPOS DE OPERADORES DE EXPRESIONES REGULARES.

Según Regulares, 1. Expresiones. (n.d.) Comúnmente existen tres operadores de las expresiones regulares: Unión, concatenación y cerradura.

Si L y M son dos lenguajes, su unión se denota por $L \cup M$ e.g. $L = \{001, 10, 111\}$, $M = \{\epsilon, 001\}$, entonces la unión será $L \cup M = \{\epsilon, 10, 001, 111\}$.

La concatenación de lenguajes se denota como LM o $L.M$ e.g. $L = \{001, 10, 111\}$, $M = \{\epsilon, 001\}$, entonces la concatenación será $LM = \{001, 10, 111, 001001, 10001, 111001\}$.

Finalmente, la cerradura (o cerradura de Kleene) de un lenguaje L se denota como L^* . Representa el conjunto de cadenas que pueden formarse tomando cualquier número de cadenas de L , posiblemente con repeticiones y concatenando todas ellas e.g. si $L = \{0, 1\}$, L^* son todas las cadenas con 0's y 1's. Si $L = \{0, 11\}$, entonces L^* son todas las cadenas de 0's y 1's tal que los 1's están en pareja.

Para calcular L^* se debe calcular L_i para cada i y tomar la unión de todos estos lenguajes. L_i tiene 2^i elementos. Aunque cada L_i es finito, la unión del número de términos de L_i es en general un conjunto infinito e.g. $\emptyset^* = \{\epsilon\}$ o $\emptyset 0 = \{\epsilon\}$. Generalizando, para toda i mayor o igual que uno, \emptyset^i es el conjunto vacío, no se puede seleccionar ninguna cadena del conjunto vacío.

EXPLICAR EL PROCESO DE CONVERSIÓN DE DFA A EXPRESIONES REGULARES.

Para convertir expresiones regulares a ϵ -NFA se realizan pruebas por inducción en los diferentes operadores (+, concatenación, *) en la expresión regular. Siempre se construye un autómata de una forma especial.

Se mostrara que un NFA con transiciones- ϵ puede aceptar el lenguaje de una RE. Después, se mostrara que un RE puede describir el lenguaje de un DFA (la misma construcción funciona para un NFA). Los lenguajes aceptados por DFA, NFA, ϵ -NFA, RE son llamados lenguajes regulares. Regulares, 1. Expresiones. (n.d.)

EXPLICAR LEYES ALGEBRAICAS DE EXPRESIONES REGULARES.

- Leyes distributivas

Esta implica a dos operadores y establece que un operador puede aplicarse por separado a cada argumento del otro operador. Existe una ley análoga para las expresiones regulares, que tenemos que establecer de dos formas

$L(M + N) = LM + LN$. Ésta es la ley distributiva por la izquierda de la concatenación respecto de la unión.

$(M + N)L = ML + NL$. Ésta es la ley distributiva por la derecha de la concatenación respecto de la unión.

- Ley de idempotencia

Se dice que un operador es idempotente si el resultado de aplicarlo a dos valores iguales es dicho valor. Los operadores aritméticos habituales no son idempotentes.

$L + L = L$. Ésta es la ley de idempotencia para la unión, que establece que si tomamos la unión de dos expresiones idénticas, podemos reemplazarla por una copia de la de la expresión.

EJEMPLO

The screenshot shows a web-based regular expression testing interface. At the top, it says "REGULAR EXPRESSION" and "1 match, 302 steps (~0ms)". The regular expression entered is `/([a-zA-Z0-9._%+-]+@[a-zA-Z0-9.-]+\.[a-zA-Z]{2,6})/g`. Below this, the "TEST STRING" section contains the text: "Hola Juan: Evitemos las comisiones de esta plataforma. Envíame directamente un correo a mi dirección fulanito@gmail.com.". The "SUBSTITUTION" section shows the result of replacing the matched email address with the text "<correo eliminado>". The final output string is: "Hola Juan: Evitemos las comisiones de esta plataforma. Envíame directamente un correo a mi dirección <correo eliminado>."

BIBLIOGRAFIA

Regulares, 1. Expresiones. (n.d.). *Propedeutico: Teoría de Autómatas y Lenguajes Formales Expresiones regulares y lenguajes*. Inaoep.Mx. Recuperado Enero 25, 2024, de https://posgrados.inaoep.mx/archivos/PosCsComputacionales/Curso_Propedeutico/Automatas/03_Automatas_ExpresionesRegularesLenguajes/CAPTUL1.PDF

ANEXO DE IMAGEN

Expresión	Explicación y ejemplos de coincidencias
'[abc]'	El patrón coincide con la cadena si en esta hay ya sea una a, una b, o una c
'[a-c]'	Es equivalente a la expresión anterior
'c[ao]sa'	Coincide con casa y con cosa
'[^abc]'	Coincide con la cadena si en esta NO hay ninguna a, b, ni c.
'[0-9]'	Coincide con una cadena que contenga cualquier número entre el 0 y el 9
'[^0-9]'	Coincide con una cadena que NO contenga ningún número
'[a-zA-Z]'	Coincide con cualquier letra mayúscula o minúscula

