TECNOLÓGICO NACIONAL DE MEXICO

INSTITUTO TECNOLÓGICO DE IZTAPALAPA

INTEGRANTES:

GUTIERREZ ARELLANO RAFAEL
181080022

ISC-6AM

LENGUAJES Y AUTOMATAS I

M.C. ABIEL TOMÁS PARRA HERNÁNDEZ

SEP 2020 / FEB 2021

ACTIVIDAD SEMANA 12

Gutierrez Arellano Rafael

Autómatas finitos y expresiones regulares

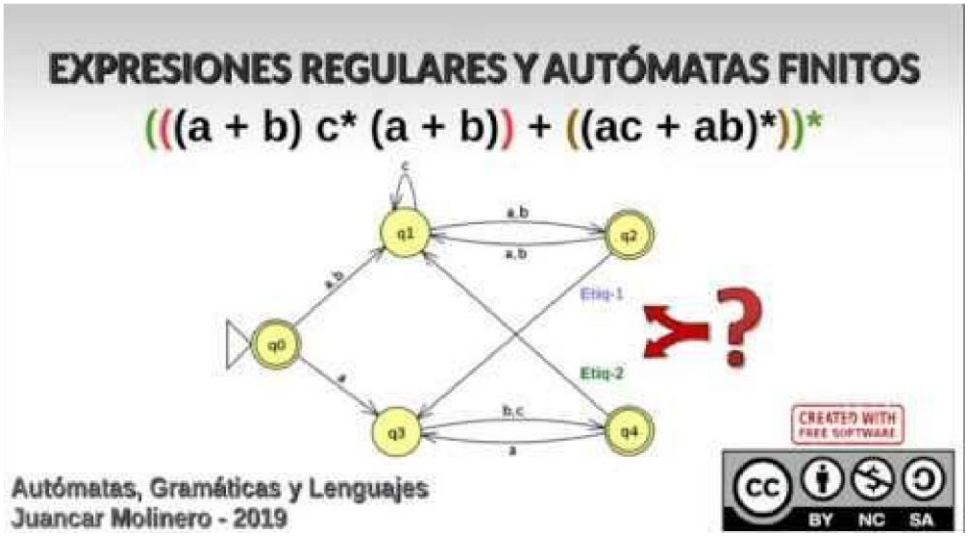
Expresiones regulares

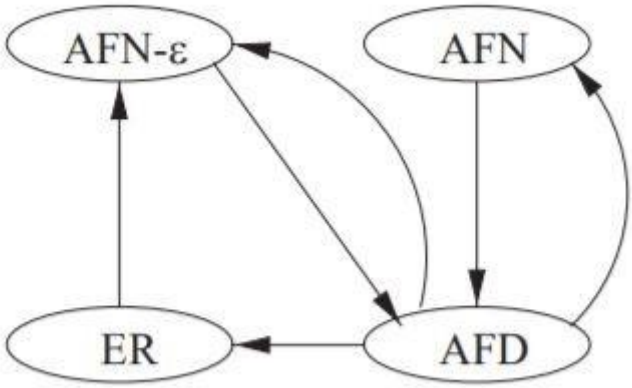
Es un equivalente algebraico para un autómata.
Utilizado en muchos lugares como un lenguaje para describir patrones en texto que son sencillos pero muy útiles.
Pueden definir exactamente los mismos lenguajes que los autómatas pueden describir: Lenguajes regulares
Ofrecen algo que los autómatas no: Manera declarativa ´ de expresar las cadenas que queremos aceptar

Ejemplos de sus usos

- Comandos de búsqueda, e.g., grep de UNIX ´
- Sistemas de formateo de texto: Usan notación de tipo expresión regular • Convierte la expresión regular a un DFA o un NFA y simula el autómata en el archivo de búsqueda
- Generadores de analizadores-lexicos. Como Lex o Flex.
- Los analizadores léxicos son parte de un compilador. Dividen el programa fuente en unidades lógicas (tokens), como while, números, signos (´ +, −, <, etc.)
- Produce un DFA que reconoce el token

Aunque las expresiones regulares describen los lenguajes de manera completamente diferente a como lo hacen los autómatas finitos, ambas notaciones representan exactamente el mismo conjunto de lenguajes, que hemos denominado “lenguajes regulares”. Ya hemos demostrado que los autómatas finitos deterministas y los dos tipos de autómatas finitos no deterministas (con y sin transiciones ε) aceptan la misma clase de lenguajes. Para demostrar que las expresiones regulares definen la misma clase, tenemos que probar que: 1. Todo lenguaje definido mediante uno de estos autómatas también se define mediante una expresión regular. Para demostrar esto, podemos suponer que el lenguaje es aceptado por algún AFD. 2. Todo lenguaje definido por una expresión regular puede definirse mediante uno de estos autómatas. Para esta parte de la demostración, lo más sencillo es probar que existe un AFN con transiciones-ε que acepta el mismo lenguaje.





Vamos a convertir el AFD de la Figura 3.4 en una expresión regular. Este AFD acepta todas las cadenas que tienen al menos un 0. Para comprender por qué, observe que el autómata va desde el estado inicial 1 al estado de aceptación 2 tan pronto como recibe una entrada 0. Después, el autómata permanece en el estado 2 para todas las secuencias de entrada.

A continuación se especifican las expresiones básicas de la construcción del Teorema 3.4.

$R_{11}^{(0)}$	$\varepsilon + \mathbf{1}$
$R_{12}^{(0)}$	$\mathbf{0}$
$R_{21}^{(0)}$	$\emptyset$
$R_{22}^{(0)}$	$(\varepsilon + \mathbf{0} + \mathbf{1})$

Por ejemplo, R(0) 11 tiene el término ε porque los estados inicial y final son el mismo, el estado 1. Contiene el término 1 porque existe un arco desde el estado 1 al estado 1 sobre la entrada 1. Otro ejemplo, R(0) 12 es 0 porque hay un arco etiquetado como 0 desde el estado 1 hasta el estado 2. No existe el término ε porque los estados inicial y final son diferentes. Como tercer ejemplo, tenemos R(0) 21 = 0, porque no existe un arco desde el estado / 2 al estado 1. Ahora tenemos que abordar la parte inductiva, construyendo expresiones más complejas que la primera teniendo en cuenta los caminos que pasan por el estado 1, luego los caminos que pasan por los estados 1 y 2, es decir, cualquier camino. La regla para calcular las expresiones(1) i j es un caso particular de la regla general dada en la parte inductiva del Teorema 3.4:

	Por sustitución directa	Simplificada
$R_{11}^{(1)}$	$\varepsilon + \mathbf{1} + (\varepsilon + \mathbf{1})(\varepsilon + \mathbf{1})^*(\varepsilon + \mathbf{1})$	$\mathbf{1}^*$
$R_{12}^{(1)}$	$\mathbf{0} + (\varepsilon + \mathbf{1})(\varepsilon + \mathbf{1})^*\mathbf{0}$	$\mathbf{1}^*\mathbf{0}$
$R_{21}^{(1)}$	$\emptyset + \emptyset(\varepsilon + \mathbf{1})^*(\varepsilon + \mathbf{1})$	$\emptyset$
$R_{22}^{(1)}$	$\varepsilon + \mathbf{0} + \mathbf{1} + \emptyset(\varepsilon + \mathbf{1})^*\mathbf{0}$	$\varepsilon + \mathbf{0} + \mathbf{1}$

Figura 3.5. Expresiones regulares para caminos que sólo pueden pasar a través del estado 1.