



TECNOLÓGICO
NACIONAL DE MÉXICO®



“TECNOLÓGICO NACIONAL DE
MEXICO”

INSTITUTO TECNOLÓGICO DE
IZTAPALAPA

INTEGRANTES:

ISC-6AM

LENGUAJES Y AUTOMATAS I

M.C. ABIEL TOMÁS PARRA
HERNÁNDEZ

SEP 2020 / FEB 2021

ACTIVIDAD SEMANA 12



Cuanenemi Cuanalo Mario Alberto

3.1 Expresiones regulares Ahora vamos a desviar nuestra atención de las descripciones tipo máquina de los lenguajes, autómatas finitos deterministas y no deterministas, a un tipo de expresión algebraica: la “expresión regular”. Comprobaremos que las expresiones regulares pueden definir de forma exacta los mismos lenguajes que describen los distintos tipos de autómatas: los lenguajes regulares. Sin embargo, las expresiones regulares ofrecen algo que los autómatas no proporcionan: una forma declarativa para expresar las cadenas que deseamos aceptar. Por tanto, las expresiones regulares sirven como lenguaje de entrada de muchos sistemas que procesan cadenas. Algunos ejemplos son los siguientes: 1. Comandos de búsqueda tales como el comando grep de UNIX o comandos equivalentes para localizar cadenas en los exploradores web o en los sistemas de formateo de texto. Estos sistemas emplean una notación de tipo expresión regular para describir los patrones que el usuario desea localizar en un archivo. Los distintos sistemas de búsqueda convierten la expresión regular bien en un AFD o en un AFN y simulan dicho autómata sobre el archivo en que se va a realizar la búsqueda. 72

Introducción a la teoría de autómatas, lenguajes y computación 2. Generadores de analizadores léxicos, como Lex o Flex. Recuerde que un analizador léxico es el componente de un compilador que divide el programa fuente en unidades lógicas o sintácticas formadas por uno o más caracteres que tienen un significado. Entre las unidades lógicas o sintácticas se incluyen las palabras clave (por ejemplo, while), identificadores (por ejemplo, cualquier letra seguida de cero o más letras y/o dígitos) y signos como + o <=. Un generador de analizadores léxicos acepta descripciones de las formas de las unidades lógicas, que son principalmente expresiones regulares, y produce un AFD que reconoce qué unidad lógica aparece a continuación en la entrada. 3.1.1 Operadores de las expresiones regulares Las expresiones regulares denotan lenguajes. Por ejemplo, la expresión regular $0^* + 10^*$ define el lenguaje que consta de todas las cadenas que comienzan con un 0 seguido de cualquier número de 1s o que comienzan por un 1 seguido de cualquier número de 0s. No esperamos que el lector sepa ya cómo interpretar las expresiones regulares, por lo que por el momento tendrá que aceptar como un acto de fe nuestra afirmación acerca del lenguaje de esta expresión. Enseguida definiremos todos los símbolos empleados en esta expresión, de modo que pueda ver por qué nuestra interpretación de esta expresión regular es la correcta. Antes de describir la notación de las expresiones regulares, tenemos que estudiar las tres operaciones sobre los lenguajes que representan los operadores de las expresiones regulares. Estas operaciones son: 1. La unión de dos lenguajes L y M, designada como $L \cup M$, es el conjunto de cadenas que pertenecen a L, a M o a ambos. Por ejemplo, si $L = \{001, 10, 111\}$ y $M = \{\epsilon, 001\}$, entonces $L \cup M = \{\epsilon, 10, 001, 111\}$. 2. La concatenación de los lenguajes L y M es el conjunto de cadenas que se puede formar tomando cualquier cadena de L y concatenándola con cualquier cadena de M. Recuerde la Sección 1.5.2, donde definimos la concatenación de una pareja de cadenas; el resultado de la concatenación es una cadena seguida de la otra. Para designar la concatenación de lenguajes se emplea el punto o ningún operador en absoluto, aunque el operador de concatenación frecuentemente se llama “punto”. Por ejemplo, si $L = \{001, 10, 111\}$ y $M = \{\epsilon, 001\}$, entonces $L.M$, o simplemente LM , es $\{001, 10, 111, 001001, 10001, 111001\}$. Las tres primeras cadenas de LM son las cadenas de L



concatenadas con ϵ . Puesto que ϵ es el elemento identidad para la concatenación, las cadenas resultantes son las mismas cadenas de L . Sin embargo, las tres últimas cadenas de LM se forman tomando cada una de las cadenas de L y concatenándolas con la segunda cadena de M , que es 001. Por ejemplo, la concatenación de la cadena 10 de L con la cadena 001 de M nos proporciona la cadena 10001 para LM .

3. La clausura (o asterisco, o clausura de Kleene)¹ de un lenguaje L se designa mediante L^* y representa el conjunto de cadenas que se pueden formar tomando cualquier número de cadenas de L , posiblemente con repeticiones (es decir, la misma cadena se puede seleccionar más de una vez) y concatenando todas ellas. Por ejemplo, si $L = \{0,1\}$, entonces L^* es igual a todas las cadenas de 0s y 1s. Si $L = \{0,11\}$, entonces L^* constará de aquellas cadenas de 0s y 1s tales que los 1s aparezcan por parejas, como por ejemplo 011, 11110 y ϵ , pero no 01011 ni 101. Más formalmente, L^* es la unión infinita $\bigcup_{i \geq 0} L^i$, donde $L^0 = \{\epsilon\}$, $L^1 = L$ y L^i , para $i > 1$ es $L L^{i-1}$ (la concatenación de i copias de L).

EJEMPLO 3.1 Dado que la idea de clausura de un lenguaje es algo engañosa, vamos a estudiar algunos ejemplos. Primero, sea $L = \{0,11\}$. $L^0 = \{\epsilon\}$, independientemente de qué lenguaje sea L ; la potencia 0 representa la selección de 1 término “clausura de Kleene” hace referencia a S. C. Kleene, quien ideó la notación de las expresiones regulares y este operador.





Capítulo 3 Lenguajes y expresiones regulares

73

Uso del operador Veamos en primer lugar el operador presentado en la Sección 1.5.2, donde lo aplicamos a un alfabeto, por ejemplo, Σ^* . Dicho operador sirve para formar todas las cadenas cuyos símbolos han sido seleccionados de un alfabeto Σ . El operador de clausura es prácticamente igual, aunque existen algunas diferencias sutiles de tipos. Supongamos que L es el lenguaje que contiene cadenas de longitud 1 y que para cada símbolo a perteneciente a Σ existe una cadena a en L . Luego aunque L y Σ “parezcan” lo mismo, son de tipos diferentes; L es un conjunto de cadenas y Σ es un conjunto de símbolos. Por otro lado, L^* designa el mismo lenguaje que Σ^* . cero cadenas de L . $L^1 = L$, representa la elección de una cadena de L . Por tanto, los dos primeros términos de la expansión de L^* nos da $\{\epsilon, 0, 11\}$. A continuación considere L^2 . Seleccionamos dos cadenas de L , permitiendo repeticiones, de modo que tenemos cuatro posibilidades. Estas cuatro selecciones nos dan $L^2 = \{00, 011, 110, 1111\}$. De forma similar, L^3 es el conjunto de cadenas que se pueden formar eligiendo tres posibilidades de las dos cadenas de L , lo que nos proporciona $\{000, 0011, 0110, 1100, 01111, 11011, 11110, 111111\}$. Para calcular L^* , tenemos que calcular L^i para cada i y hallar la unión de todos estos lenguajes. L^i tiene 2^i miembros. Aunque cada L^i es finito, la unión de un número infinito de términos L^i generalmente es un lenguaje infinito, como en el caso de nuestro ejemplo. Sea ahora L el conjunto de todas las cadenas de 0s. Observe que L es infinito, a diferencia de en el ejemplo anterior en el que era un lenguaje finito. Sin embargo, no es difícil descubrir que es L^* . $L^0 = \{\epsilon\}$, como siempre. $L^1 = L$. L^2 es el conjunto de cadenas que se pueden formar tomando una cadena de 0s y concatenarla con otra cadena de 0s. El resultado sigue siendo una cadena de 0s. De hecho, toda cadena de 0s se puede escribir como la concatenación de dos cadenas de 0s (no olvide que ϵ es una “cadena de 0”; esta cadena siempre puede ser una de las dos cadenas que concatenemos). Por tanto, $L^2 = L$. Del mismo modo, $L^3 = L$, etc. Luego la unión infinita $L^* = L^0 \cup L^1 \cup L^2 \cup \dots$ es L en el caso particular de que el lenguaje L sea el conjunto de todas las cadenas de 0s. Veamos un último ejemplo. $0^* = \{\epsilon\}$. Observe que $0^0 = \{\epsilon\}$, mientras que 0^i , para todo $i \geq 1$, es el conjunto vacío, ya que no podemos seleccionar ninguna cadena en el conjunto vacío. De hecho, 0^i es uno de los dos lenguajes cuya clausura no es infinita.

3.1.2 Construcción de expresiones regulares Todos los tipos de álgebras se inician con las expresiones elementales, que normalmente son



constantes y/o variables. Las álgebras nos permiten construir más expresiones aplicando un cierto conjunto de operadores a las expresiones elementales y a las expresiones previamente construidas. Normalmente, también se necesitan algunos métodos que permitan agrupar operadores con sus operandos, tales como los paréntesis. Por ejemplo, la familiar álgebra de la aritmética se inicia con constantes, como los números enteros y reales, más las variables y se construyen expresiones más complejas utilizando operadores aritméticos como $+$ y \times . El álgebra de las expresiones regulares sigue también este patrón, utilizando constantes y variables que representan lenguajes y operadores para las tres operaciones mencionadas en la Sección 3.1.1 (la unión, el punto y el asterisco). Podemos describir las expresiones regulares recursivamente del siguiente modo. En esta     74 Introducción a la teoría de autómatas, lenguajes y computación definición, no sólo describimos lo que son las expresiones regulares válidas, sino que para cada expresión regular E , describimos el lenguaje que representa, al que denominaremos $L(E)$. BASE. El caso básico consta de tres partes: 1. Las constantes ϵ y $/0$ son expresiones regulares, que representan a los lenguajes $\{\epsilon\}$ y $/0$, respectivamente. Es decir, $L(\epsilon) = \{\epsilon\}$ y $L(/0) = /0$. / 2. Si a es cualquier símbolo, entonces a es una expresión regular. Esta expresión representa el lenguaje $\{a\}$. Es decir, $L(a) = \{a\}$. Observe que utilizamos la fuente en negrita para indicar la expresión correspondiente a un símbolo. La correspondencia, por ejemplo, que a hace referencia a a , es obvia. 3. Una variable, normalmente escrita en mayúsculas e itálicas, como L , representa cualquier lenguaje. PASO INDUCTIVO. Existen cuatro partes en el paso de inducción, una para cada uno de los tres operadores y otra para la introducción de paréntesis. 1. Si E y F son expresiones regulares, entonces $E + F$ es una expresión regular que representa la unión de $L(E)$ y $L(F)$. Es decir, $L(E + F) = L(E) \cup L(F)$. 2. Si E y F son expresiones regulares, entonces EF es una expresión regular que representa la concatenación de $L(E)$ y $L(F)$. Es decir, $L(EF) = L(E)L(F)$. Observe que el punto puede utilizarse opcionalmente para explicitar el operador de concatenación, bien como una operación sobre lenguajes o como el operador en una expresión regular. Por ejemplo, 0.1 es una expresión regular que significa lo mismo que 01 y que representa el lenguaje $\{01\}$. Sin embargo, nosotros vamos a evitar el uso del punto en la concatenación de expresiones regulares.2 3.

Si E es una expresión regular, entonces E^* es una expresión regular, que representa la clausura de $L(E)$. Es decir, $L(E^*) = L(E)^*$. 4. Si E es una expresión regular, entonces (E) , una E encerrada entre paréntesis, es también