Introducción a la Teoría de Autómatas, Lenguajes y Computación

Gustavo Rodríguez Gómez y Aurelio López López

INAOE

Propedéutico 2020

Capítulo 3

Expresiones y Lenguajes Regulares

Outline I

Outline II

- Identidades y Aniquiladores
- Leyes Distributivas
- Ley Idempotente
- Cerraduras

Introducción

- Los autómatas finitos (**AF**) deterministas o no deterministas son "moldes" para construir máquinas que reconocen lenguajes regulares.

Introducción

- Los autómatas finitos (**AF**) deterministas o no deterministas son "moldes" para construir máquinas que reconocen lenguajes regulares.
- Una **expresión regular** es una declaración amigable para describir a un lenguaje regular

Introducción

- Los autómatas finitos (**AF**) deterministas o no deterministas son "moldes" para construir máquinas que reconocen lenguajes regulares.
- Una **expresión regular** es una declaración amigable para describir a un lenguaje regular
  - Ejemplo

$$\mathbf{01}^* + \mathbf{10}^*$$

Introducción

- Los autómatas finitos (**AF**) deterministas o no deterministas son "moldes" para construir máquinas que reconocen lenguajes regulares.
- Una **expresión regular** es una declaración amigable para describir a un lenguaje regular
  - Ejemplo

$$\mathbf{01}^* + \mathbf{10}^*$$

- Las expresiones regulares se utilizan

Introducción

- Los autómatas finitos (**AF**) deterministas o no deterministas son “moldes” para construir máquinas que reconocen lenguajes regulares.
- Una **expresión regular** es una declaración amigable para describir a un lenguaje regular
  - Ejemplo
$$\mathbf{01}^* + \mathbf{10}^*$$
- Las expresiones regulares se utilizan
  - En los comandos grep de UNIX

Introducción

- Los autómatas finitos (**AF**) deterministas o no deterministas son “moldes” para construir máquinas que reconocen lenguajes regulares.
- Una **expresión regular** es una declaración amigable para describir a un lenguaje regular
  - Ejemplo

$$\mathbf{01}^* + \mathbf{10}^*$$

- Las expresiones regulares se utilizan
  - En los comandos grep de UNIX
- Las expresiones regulares son las descripciones algebraicas de los lenguajes de los AF

Operaciones sobre Lenguajes

Sean L y M dos lenguajes

1. **Unión**

$$L \cup M = \{w \mid w \in L \text{ o } w \in M\}$$

2. **Concatenación**

$$L.M = \{w \mid w = xy,\ x \in L,\ y \in M\}$$

3. **Potencias**

$$L^0 = \{\epsilon\}\,,\ L^1 = L,\ L^{k+1} = L.L^k$$

4. **Cerradura de Kleene**

$$L^* = \bigcup_{i=0}^{\infty} L^i$$

Ejemplos de Operaciones sobre Lenguajes

Sea $L = \{001, 10, 111\}$, $M = \{\epsilon, 001\}$ entonces

$$L \cup M = \{\epsilon, 10, 001, 111\} ,$$
$$L.M = \{001, 10, 111, 001001, 10001, 111001\} ,$$

Considere ahora $L = \{0, 11\}$ vamos a determinar su cerradura de Kleene

$L^0 = \{\epsilon\}$, $L^1 = \{0, 11\}$,
$L^2 = L.L = \{00, 011, 110, 1111\}$,
$L^3 = L.L^2 = \{000, 0011, 0110, 01111, 1100, 11011, 11110, 111111\}$,

observe que L^i tiene 2^i miembros pero $|L^*| = |\mathbb{N}|$.

Ejemplos de Operaciones sobre Lenguajes

Cerradura Kleene ϕ

Sea $L = \phi$, entonces

$$\phi^0 = \{\epsilon\},$$
$$\phi^1 = \phi,$$
$$\phi^i = \phi, \; i \geq 1,$$

Luego

$$\phi^* = \{\epsilon\}$$

Algebra de Expresiones Regulares I

Describimos las expresiones regulares recursivamente. Por cada expresión regular E describimos el lenguaje que representa con L(E).

Bases:

1. Las constantes ϵ y ϕ son expresiones regulares que denotan los lenguajes $\{\epsilon\}$ y ϕ, esto es, $L(\epsilon) = \{\epsilon\}$, $L(\phi) = \phi$.
2. Si $a \in \Sigma$, entonces **a** es una expresión regular y denota el lenguaje $\{a\}$: $L(\mathbf{a}) = \{a\}$.
3. Una variable en mayúsculas e itálicas como L, es una variable que representa cualquier lenguaje

Inducción:

Algebra de Expresiones Regulares II

1. Si E y F son expresiones regulares $\Rightarrow$ $E + F$ es una expresión regular que denota la unión de $L(E)$ con $L(F)$: $L(E+F) = L(E) \cup L(F)$.
2. Si E y F son expresiones regulares $\Rightarrow$ EF es una expresión regular que denota la concatenación de $L(E)$ con $L(F)$: $L(EF) = L(E)L(F)$.
3. Si E es una expresión regular $\Rightarrow$ E^* es una expresión regular que denota la cerradura de $L(E)$: $L(E^*) = (L(E))^*$.
4. Si E es una expresión regular $\Rightarrow$ (E) es una expresión regular que denota el mismo lenguaje de E : $L((E)) = L(E)$.

Ejemplos de Expresiones Regulares

1. $a, b \in \Sigma$, $\mathbf{a}+\mathbf{b} \Rightarrow L(\mathbf{a}+\mathbf{b}) = L(\mathbf{a}) \cup L(\mathbf{b}) = \{a\} \cup \{b\}$.
2. $a, b \in \Sigma$, $\mathbf{ab} \Rightarrow L(\mathbf{ab}) = L(\mathbf{a})L(\mathbf{b}) = \{a\}\{b\} = \{ab\}$.
3. $a \in \Sigma$, $\mathbf{a}^* \Rightarrow L(\mathbf{a}^*) = (L(\mathbf{a}))^* = \{\epsilon, a, aa, aaa, aaaa, \ldots\}$.
4. $\mathbf{aa}^* = \{a\}\{a\}^* = \{a, aa, aaa, aaaa, aaaaa, \ldots\}$
5. $(\mathbf{a}+\mathbf{b})^* = (\{a\} \cup \{b\})^* = (\{a, b\})^* = \{\epsilon, a, b, ab, ba, bb, \ldots\}$.

Ejemplos de Expresiones Regulares

Considere el lenguaje $L = \left\{ w \in \{0,1\}^* \mid 0 \text{ y } 1 \text{ alternan en } w \right\}$

$$(\mathbf{01})^* + (\mathbf{10})^* + 0\,(\mathbf{10})^* + 1\,(\mathbf{01})^*\,,$$

equivalentemente

$$(\epsilon + \mathbf{1})\,(\mathbf{01})^*\,(\epsilon + \mathbf{0})$$

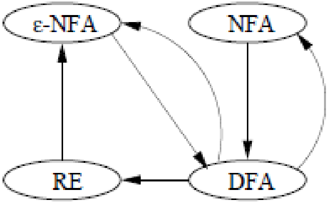
Orden de Precedencia de las Operaciones

1. Estrella
2. Punto
3. Mas

Ejemplo $\mathbf{01}^* + \mathbf{1} = (\mathbf{0}(\mathbf{1})^*) + \mathbf{1}$

Equivalencia entre AF y ER

La manera en que las expresiones regulares y AF describen los lenguajes son radicalmente diferentes. Sin embargo, las dos notaciones representan el mismo lenguaje.



Equivalencia entre AF y ER I

Theorem

Si $L = L(A)$ para algún AFD A, entonces existe una expresión regular R tal que $L(A) = L(R)$.

Demostración. Supongamos que los estados de A son denotados por $\{1, 2, \ldots, n\}$ donde 1 es el estado de inicio. Nuestro objetivo es construir una colección de ER que describan conjuntos de caminos progresivamente más amplios en el diagrama de transición de A.

- Sea R_{ij}^{k} una ER cuyo lenguaje es el conjunto de todas las cadenas w tal que w es la etiqueta de un camino del estado i al estado j de A que va a través de los estados intermedios $\{1, 2, \ldots, k\}$ únicamente.

Equivalencia entre AF y ER II

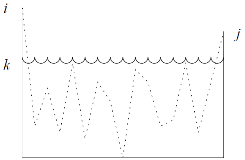


Figure: Caminos representados por R_{ij}^{k}. Dimensión vertical: estado 1 en la parte inferior y el n en la parte superior. Dimensión horizontal representa el recorrido a lo largo del camino.

Construcción de R_{ij}^{k} por medio de definición inductiva.
Base: $k = 0$ no hay estados intermedios. Hay dos clases de caminos que satisfacen la condición

Equivalencia entre AF y ER III

1. Un arco del nodo i al nodo j.
2. Un camino de longitud 0 que consiste únicamente del nodo i.

Si $i \neq j$ (1) es la única posibilidad. Examinamos el AFD A para encontrar los símbolos de entrada a tal que haya una transición del estado i al estado j por medio de a.

- Si el símbolo a no existe hacemos $R_{ij}^{0} = \phi$.
- Si hay únicamente un símbolo a hacemos $R_{ij}^{0} = \mathbf{a}$.
- Si hay $a_1, a_2, \ldots, a_k$ símbolos que etiquetan los arcos del estado i al j hacemos $R_{ij}^{0} = \mathbf{a}_1 + \mathbf{a}_2 + \cdots + \mathbf{a}_k$

Si $i = j$ los caminos asociados son los de longitud 0 y todos los lazos de i consigo mismo entonces

- Camino de longitud 0, $R_{ij}^{0} = \epsilon$.

Equivalencia entre AF y ER IV

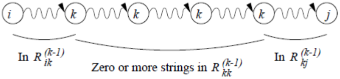
- Hay un símbolo a, $R_{ij}^0 = \epsilon + \mathbf{a}$.
- Múltiples símbolos $R_{ij}^0 = \epsilon + \mathbf{a}_1 + \mathbf{a}_2 + \cdots + \mathbf{a}_k$.

Inducción: Supongamos que existe un camino del estado i al estado j que no pasa por un estado superior a k. Hay dos posibles casos

- El camino no pasa a través del estado k en ningún caso.
- Pasa por k una o más veces. Luego

$$R_{ij}^k = R_{ij}^{k-1} + R_{ik}^{k-1}(R_{kk}^{k-1})^* R_{kj}^{k-1}$$

Equivalencia entre AF y ER V

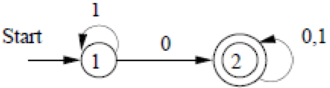


- **Paso final**: La ER para el lenguaje del AFD es la unión de todas las expresiones R_{1j}^{n} tal que j es uno de los estados aceptados.

AFD a ER

- Encuentre la expresión regular del AFD A donde

$$L(A) = \{x0y \mid x \in \{1\}^* \text{ y } y \in \{0,1\}^*\}$$



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

Reglas de Simplificación

1. $(\epsilon + R)^* = R^*$
2. $R + RS^* = RS^*$
3. $\phi R = R\phi = \phi$
4. $\phi + R = R + \phi = R$

AFD a ER (continuación)

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

$$R_{ij}^{(1)} = R_{ij}^{(0)} + R_{i1}^{(0)}\left(R_{11}^{(0)}\right)^* R_{1j}^{(0)}$$

	By direct substitution	Simplified
$R_{11}^{(1)}$	$\epsilon + 1 + (\epsilon + 1)(\epsilon + 1)^*(\epsilon + 1)$	1^*
$R_{12}^{(1)}$	$0 + (\epsilon + 1)(\epsilon + 1)^*0$	1^*0
$R_{21}^{(1)}$	$\emptyset + \emptyset(\epsilon + 1)^*(\epsilon + 1)$	$\emptyset$
$R_{22}^{(1)}$	$\epsilon + 0 + 1 + \emptyset(\epsilon + 1)^*0$	$\epsilon + 0 + 1$

AFD a ER (continuación)

	Simplified
$R_{11}^{(1)}$	1^*
$R_{12}^{(1)}$	1^*0
$R_{21}^{(1)}$	$\emptyset$
$R_{22}^{(1)}$	$\epsilon + 0 + 1$

$$R_{ij}^{(2)} = R_{ij}^{(1)} + R_{i2}^{(1)} \left(R_{22}^{(1)}\right)^* R_{2j}^{(1)}$$

	By direct substitution
$R_{11}^{(2)}$	$1^* + 1^*0(\epsilon + 0 + 1)^*\emptyset$
$R_{12}^{(2)}$	$1^*0 + 1^*0(\epsilon + 0 + 1)^*(\epsilon + 0 + 1)$
$R_{21}^{(2)}$	$\emptyset + (\epsilon + 0 + 1)(\epsilon + 0 + 1)^*\emptyset$
$R_{22}^{(2)}$	$\epsilon + 0 + 1 + (\epsilon + 0 + 1)(\epsilon + 0 + 1)^*(\epsilon + 0 + 1)$

AFD a ER (continuación)

	By direct substitution
$R_{11}^{(2)}$	$1^* + 1^*0(\epsilon + 0 + 1)^*\emptyset$
$R_{12}^{(2)}$	$1^*0 + 1^*0(\epsilon + 0 + 1)^*(\epsilon + 0 + 1)$
$R_{21}^{(2)}$	$\emptyset + (\epsilon + 0 + 1)(\epsilon + 0 + 1)^*\emptyset$
$R_{22}^{(2)}$	$\epsilon + 0 + 1 + (\epsilon + 0 + 1)(\epsilon + 0 + 1)^*(\epsilon + 0 + 1)$

	Simplified
$R_{11}^{(2)}$	1^*
$R_{12}^{(2)}$	$1^*0(0 + 1)^*$
$R_{21}^{(2)}$	$\emptyset$
$R_{22}^{(2)}$	$(0 + 1)^*$

The final regex for A is

$$R_{12}^{(2)} = 1^*0(0 + 1)^*$$

Observaciones

1. Hay n^3 expresiones R_{ij}^k

Observaciones

1. Hay n^3 expresiones R_{ij}^k
2. Cada paso inductivo aumenta la expresión 4 veces

Observaciones

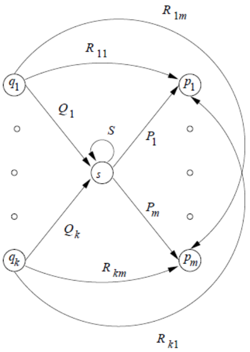
1. Hay n^3 expresiones R_{ij}^{k}
2. Cada paso inductivo aumenta la expresión 4 veces
3. Para toda $\{i, j\} \subseteq \{1, 2, \ldots, n\}$ emplea R_{kk}^{k-1}, hay que escribir n^2 la ER R_{kk}^{k-1}.

Observaciones

1. Hay n^3 expresiones R_{ij}^{k}
2. Cada paso inductivo aumenta la expresión 4 veces
3. Para toda $\{i, j\} \subseteq \{1, 2, \ldots, n\}$ emplea R_{kk}^{k-1}, hay que escribir n^2 la ER R_{kk}^{k-1}.
4. Es necesario una estrategia más eficiente.

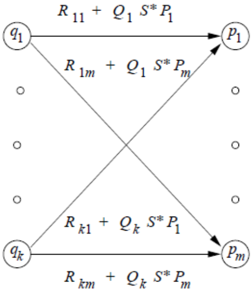
Técnica de Eliminación de Estados I

Etiquete las aristas con ER's en lugar de símbolos



Siguiente paso elimine el estado s

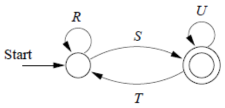
Técnica de Eliminación de Estados II



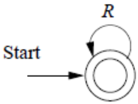
por cada estado aceptado q elimine del autómata original todos los estados excepto q_0 y q.

Para cada $q \in F$ nos quedaremos con una A_q que se parezca a

Técnica de Eliminación de Estados III



que corresponde a la ER $E_q = (R + SU^*T)SU^*$ o con una A_q que se parezca a



que corresponde a la ER $E_q = R^*$.

Técnica de Eliminación de Estados IV

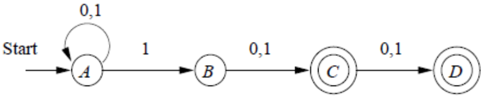
- La expresión final es la unión de todas las E_q con $q \in F$

$$\bigoplus_{q \in F} E_q$$

Ejemplo Eliminación de Estados I

Sea A el AFND con

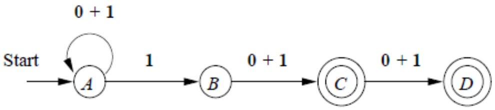
$$L(A) = \{w : w = x1b \text{ ó } w = x1bc, x\{0,1\}^*, \{b, c\} \subseteq \{0,1\}\}$$



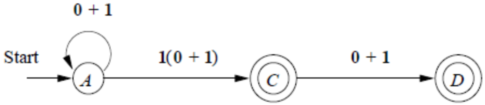
Convertiremos a A en una expresión regular por medio de la estrategia de Eliminación de Estados.

Ejemplo Eliminación de Estados II

- Primer paso convertimos a A en un autómata cuyas etiquetas sean ER

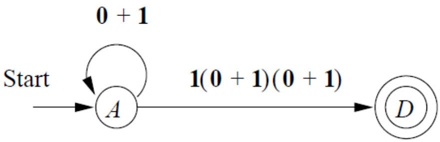


- Elimínanos el estado B



Ejemplo Eliminación de Estados III

- Eliminamos C y obtenemos el autómata A_D

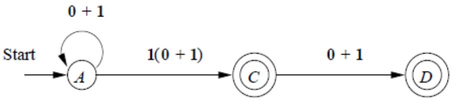


cuya ER es

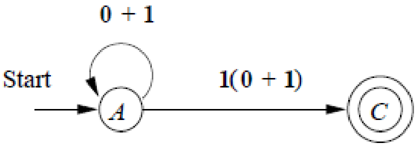
$$(\mathbf{0}+\mathbf{1})^*\mathbf{1}(\mathbf{0}+\mathbf{1})(\mathbf{0}+\mathbf{1})$$

Ejemplo Eliminación de Estados IV

- Del diagrama



eliminamos D y obtenemos el autómata A_C



Ejemplo Eliminación de Estados V

cuya ER es

$$(\mathbf{0}+\mathbf{1})^*(\mathbf{0}+\mathbf{1})$$

- Finalmente obtenemos la expresión final de la ER como la suma de las dos expresiones anteriores

$$(\mathbf{0}+\mathbf{1})^*\mathbf{1}(\mathbf{0}+\mathbf{1})(\mathbf{0}+\mathbf{1})+(\mathbf{0}+\mathbf{1})^*(\mathbf{0}+\mathbf{1})$$

ER a Autómata

Theorem

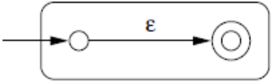
Para cada ER R que define el lenguaje $L(R)$ se puede construir un $\in -AFND$ A tal que $L(A) = L(R)$.

Demostración ER a Autómata I

Demostración. Supongamos que $L(R)$ es un lenguaje para la ER R. Demostraremos que el $L(E)$ asociado a algún $\in -AFND$ coincide con $L(R)$ si se satisface las condiciones

1. $\in -AFND$ tiene un único estado aceptado
2. No hay arcos en el estado inicial.
3. No hay arcos fuera del estado aceptado.

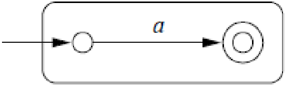
Base: Autómatas para $\in, \phi, a$



(a)

Demostración ER a Autómata II

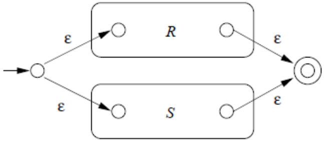
(b)



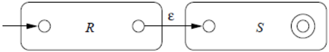
(c)

Demostración ER a Autómata III

Inducción: Autómatas para $R + S$, RS y R^*

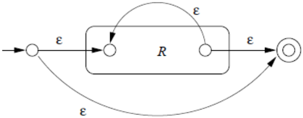


(a)



(b)

Demostración ER a Autómata IV

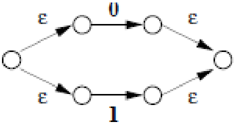


(c)

Ejemplo ER a Autómata I

Convertir la ER $(\mathbf{0}+\mathbf{1})^*\mathbf{1}(\mathbf{0}+\mathbf{1})$

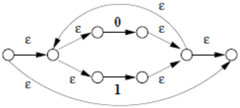
- Suma $\mathbf{0}+\mathbf{1}$



(a)

Ejemplo ER a Autómata II

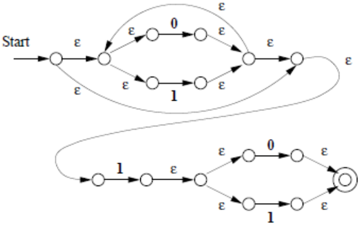
- Estrella $(\mathbf{0} + \mathbf{1})^*$



(b)

Ejemplo ER a Autómata III

- Concatenación y suma $\mathbf{1}(\mathbf{0}+\mathbf{1})$



(c)

Leyes Algebraicas ERs

- La unión es conmutativa $L + M = M + L$
- La unión es asociativa $(L + M) + N = L + (M + N)$
- Concatenación es asociativa $(LM)N = L(MN)$
- La concatenación no es conmutativa $LM \neq ML$, ejemplo $\mathbf{01} \neq \mathbf{10}$

Identidades y Aniquiladores

- ϕ es la identidad (neutro) para la unión $\phi + L = L + \phi$
- ϵ es la identidad para la concanetación $\epsilon L = L\epsilon = L$
- ϕ es el aniquilador para la concatenación $\phi L \neq L\phi = \phi$

Identidades y Aniquiladores

- Ley distributiva izquierda de la concatenación sobre la unión $L(M + N) = LM + LN$.
- Ley distributiva derecha de la concatenación sobre la unión $(M + N)L = ML + NL$.

Idempotente

Un operador se dice idempotente si tiene la propiedad que al aplicarse a si mismo vuelve a obtenerse el mismo operador.

- La unión es idempotente $L + L = L$.

Cerraduras

- $(L^*)^* = L^*$.
- $\phi^* = \{\epsilon\}$, $\{\epsilon\}^* = \{\epsilon\}$
- $L^+ = LL^* = L^*L$, $L^* = L^* + \epsilon$

Demostración Idempotente

- Demostraremos que $(L^*)^* = L^*$.

Demostración. Sea

$$
\begin{aligned}
w \in (L^*)^* &\Leftrightarrow w \in \bigcup_{i=0'}^{\infty} \left(\bigcup_{j=0}^{\infty} L^j \right)^i \\
&\Leftrightarrow \exists k, m \in \mathbb{N} \text{ tal que } w \in (L^m)^k \\
&\Leftrightarrow \exists p \in \mathbb{N} \text{ tal que } w \in L^P \\
&\Leftrightarrow w \in \bigcup_{i=0}^{\infty} L^i \\
&\Leftrightarrow w \in L^*
\end{aligned}
$$