Lenguajes Formales

Expresiones Regulares

Ing. Fabiana Aragón

Expresiones Regulares

Las ER son una notación especial que se utiliza habitualmente para describir los lenguajes de tipo regular.

La notación más utilizada para especificar patrones son las ***expresiones regulares,*** sirven como nombres para conjuntos de cadenas.

Los usos más habituales de las expresiones regulares son en la creación de analizadores léxicos para compiladores, en la búsqueda de patrones dentro de los editores de texto, en la descripción de redes neuronales y circuitos electrónicos, etc.

Expresiones Regulares

En las expresiones regulares pueden aparecer símbolos de dos tipos:

- **Símbolos base:** son los símbolos del alfabeto del lenguaje que queremos describir, λ para describir la palabra vacía y Ø para describir un lenguaje vacío.

- **Símbolos de operadores:** "+" o unión, "." o concatenación y "*" para la clausura. En ocasiones se utiliza para la unión el símbolo "|" y el punto se omite en la concatenación de expresiones regulares.

Lenguajes de las Expresiones Regulares

El lenguaje de las ER se construye de forma inductiva a partir de símbolos para expresiones regulares elementales y operadores.

Expresiones regulares

Dado un alfabeto Σ y los símbolos: ∅ (lenguaje vacío), λ (palabra vacía), · (concatenación), + (unión), *(clausura), se cumple que:

- ✓ *Los símbolos λ y ∅ son expresiones regulares.*
- ✓ *Cualquier símbolo a ∈ Σ es una expresión regular.*
- ✓ *Si u y v son expresiones regulares, entonces, u+v, u.v, u* y v* son expresiones regulares.*

Lenguajes de las Expresiones Regulares

Sólo son expresiones regulares las que se pueden obtener aplicando las reglas anteriores un número finito de veces.

Se establece la siguiente prioridad en las operaciones:

1. *Paréntesis ( )*
2. *Clausura **
3. *Concatenación ·*
4. *Unión +*

Lenguajes de las Expresiones Regulares

A cada expresión regular le corresponde un **lenguaje regular:**

- Si $\alpha = \emptyset$, $L(\alpha) = \emptyset$
- Si $\alpha = \lambda$, $L(\alpha) = \{\lambda\}$.
- Si $\alpha = a$, $a \in \Sigma$, $L(\alpha) = \{a\}$
- Si α y β son dos expresiones regulares entonces:
- $L(\alpha+\beta) = L(\alpha) \cup L(\beta)$
- $L(\alpha\beta) = L(\alpha)L(\beta)$
- $L(\alpha^*) = (L(\alpha))^*$
- $L((\alpha)) = L(\alpha)$

Ejemplo: sobre el alfabeto $\Sigma = \{ a, b \}$, podemos definir las siguientes expresiones regulares:

Expresiones Regulares	Lenguaje
ab	{ ab }
$a + \emptyset$	{ a }
$(a+b)^*a$	$\{ a^n b^m a \ / \ n, m \in N \}$
$a + \lambda$	$\{ \lambda , a \}$
b^*ab^*	$\{ b^n ab^m / n, m \in N \}$
a^*	$\{ \lambda, a, aa, aaa, \dots \}$

Lenguajes de las Expresiones Regulares

Propiedades de la Unión:

- Asociativa: $\alpha + (\beta + \gamma) = (\alpha + \beta) + \gamma$
- Conmutativa: $\alpha + \beta = \beta + \alpha$
- Elemento neutro la expresión ⟶ vacío: $\varnothing + \alpha = \alpha + \varnothing = \alpha$
- Idempotente: $\alpha + \alpha = \alpha$

Propiedades de la concatenación:

- Asociativa: $\alpha (\beta \gamma) = (\alpha \beta) \gamma$
- <u>No</u> es conmutativa: $\alpha \beta \neq \beta \alpha$
- Elemento neutro la expresión ⟶ lambda: $\lambda \alpha = \alpha \lambda = \alpha$
- Tiene como elemento anulador la expresión ⟶ vacío: $\varnothing \alpha = \alpha \varnothing = \varnothing$
- Distributiva respecto al operador de unión: $\alpha (\beta + \gamma) = \alpha \beta + \alpha \gamma$

Propiedades de la clausura:

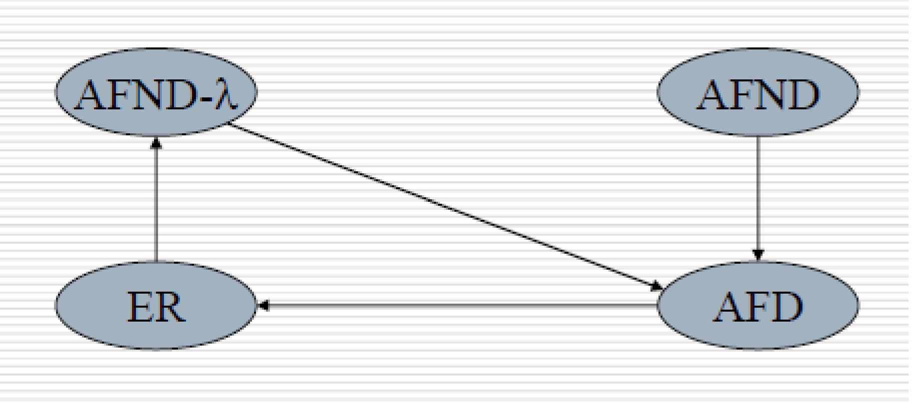
- $\lambda^* = \lambda$
- $\varnothing^* = \lambda$
- $\alpha^* = \lambda + \alpha \alpha^*$
- $\alpha \alpha^* = \alpha^* \alpha$
- $(\alpha^* + \beta^*)^* = (\alpha + \beta)^*$
- $(\alpha + \beta)^* = (\alpha^* \beta^*)^*$

Teorema de Kleene

Kleene enuncio dos teoremas que permiten demostrar la equivalencia entre E.R y A.F, en cuanto a que los dos representan de una forma u otra Lenguajes Regulares.

Los teoremas son:

- **Teorema de Análisis**: Todo lenguaje aceptado por un AF es regular.
- **Teorema de Síntesis**: Todo lenguaje regular es aceptado por un AF.



Teorema de Kleene: Solución al Teorema de Análisis

Teorema de Análisis: para demostrar que a partir de un AF se puede obtener el lenguaje que este acepta y que es regular.

Ecuación característica o fundamental.

1. Se define x_i como el conjunto de palabras que permiten pasar desde el estado q_i a un estado final.
2. Si $q_i \in F \rightarrow \lambda \in x_i$, ya que $\forall\ q \in Q$, $q \in f(q, \lambda)$. También,

   Si $q_i \notin F \rightarrow \lambda$ **no tiene porque estar en** x_i

3. Si $f(q_i, a) = q_j \in F$, entonces el símbolo ***a*** debe pertenecer a x_i

4. Si $f(q_i, a) = q_j$, entonces la concatenación en la entrada del símbolo a, x_j (conjunto de cadenas que permiten pasar desde q_j a un estado final) debe estar en x_i (conjunto de cadenas que permiten transitar desde q_i a un estado final). Así $ax_j \subseteq x_i$

Teorema de Kleene: Solución al Teorema de Análisis

5. Para cada estado q_i se puede definir el denominado *sistema de ecuaciones de conjunto lineales por la derecha* que se calcula como:

$$x_i = C_i \, U \left[\bigcup_{j=1}^{|Q|} D_{ij} \, x_j \right]$$

Donde:

(según 2) $\quad C_i = \begin{cases} \lambda & q_i \in F \\ \varnothing & q_i \notin F \end{cases}$

(según 4) $\quad D_{ij} = \{ a \in \Sigma \,/\, f(q_i, a) = q_j \}$

Teorema de Kleene: Solución al Teorema de Análisis

6. Una forma alternativa de definir este sistema de ecuaciones consiste en utilizar las definiciones 3 y 4 donde solo se cambia el conjunto C_i de la siguiente manera:

$$C_i = \{a \,/\, f(q_i, a) = q_j \in F\} \cup \begin{cases} \lambda & \text{si } i=0 \text{ y } q_0 \in F \\ \varnothing & \text{en caso contrario} \end{cases}$$

- Si el estado q_j es final, añadimos el termino **a**
- Si el estado q_0 es final, añadimos el termino λ para $\mathbf{x_0}$

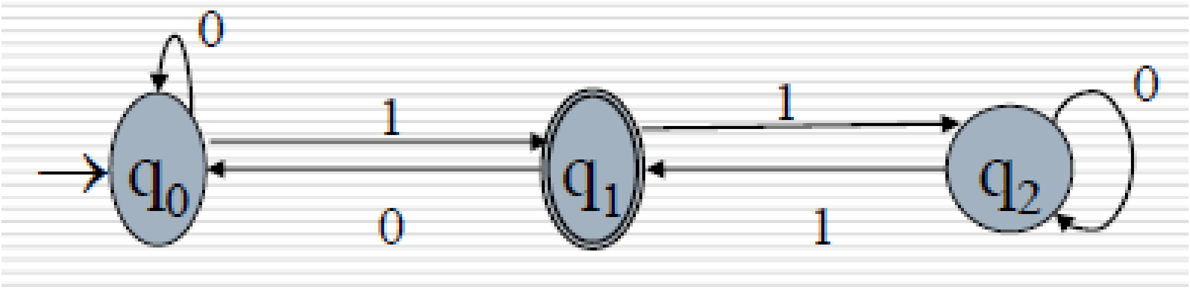
Teorema de Kleene: Solución al Teorema de Análisis

7. Una vez que se obtienen los sistemas de ecuaciones para cada estado, se debe intentar resolver las ecuaciones de cada estado, de manera que tenga la forma $x_i = Ax_i + B$ **donde** $\lambda \notin A$, $x_i \notin B$ que es a lo que se denomina **ecuaciones características.** Para cada estado en el que se tenga despejado x_i la solución es $x_i = A^* B$. <u>Nota:</u> Al efecto de que se cumpla $\lambda \notin A$, lo mas sencillo es eliminar las transiciones λ del autómata, cuando sea no determinista.

8. Cuando se calcule el valor de x_0 (ecuación característica del estado inicial), **esta E.R. es la que describe el lenguaje aceptado por el autómata**.

Teorema de Kleene. Solución al Teorema de Análisis

Ejemplo: Sea el AFD



Se crea las ecuaciones características para cada estado:

$x_0 = 0x_0 + 1x_1 + 1$ (1)

$x_1 = 0x_0 + 1x_2$ (2)

$x_2 = 0x_2 + 1x_1 + 1$ (3)

NOTA: Utilizamos la forma alternativa del punto 6

Teorema de Kleene. Solución al Teorema de Análisis

Resolvemos aplicando la regla de inferencia

*X=αX+β ⇔ X=α*β*

comenzando por la ecuación (3):

$x_2= 0x_2+ 1x_1+1$

$x_2= 0*(1x_1+1)$

$\boxed{x_2 = \mathbf{0*1x_1+0*1}}$ (4)

Sustituyendo (4) en (2)

$x_1= 0x_0+ 1x_2$

$x_1= 0x_0+ 1(0*1x_1+0*1)$

$x_1= 0x_0+ 10*1x_1+10*1)$ *Reordenando los términos*

$x_1 =10*1x_1+ 0x_0 +10*1$

$x_1 =(10*1)*(0x_0+ 10*1)$

$\boxed{x_1 = \mathbf{(10*1)*0x_0+ (10*1)*10*1}}$ (5)

Teorema de Kleene. Solución al Teorema de Análisis

Sustituyendo (5) en (1):

$x_0 = 0x_0 + 1x_1 + 1$

$x_0 = 0x_0 + 1[\mathbf{(10^*1)^*0x_0 + (10^*1)^*10^*1}]+1$

$x_0 = 0x_0 + 1(10^*1)^*0x_0 + 1(10^*1)^*10^*1+1$

$x_0 = (0 + 1(10^*1)^*0)x_0 + 1(10^*1)^*10^*1+1$ *Reordenando los términos*

$x_0 = (0+1(10^*1)^*0)^*(1(10^*1)^*10^*1+1)$ *Aplicamos la regla de inferencia:* $X=\alpha X+\beta \Leftrightarrow X=\alpha^*\beta$

$x_0 = (0+1(10^*1)^*0)^*1[(10^*1)^*(10^*1)+ \lambda]$ *Aplicamos las propiedades:* $(ab^*b+a) = a(b^*b+\lambda) = ab^*)$

$$x_0 = (0+1(10^*1)^*0)^*1(10^*1)^*$$

Expresión regular que describe el lenguaje L(A).

Teorema de Kleene: Solución al problema de Síntesis

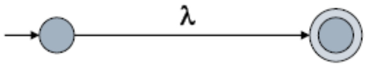
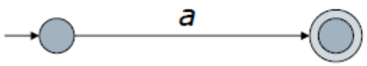
Existen 2 algoritmos para pasar de una E.R., que representa a un L.R., a un autómata que reconozca dicho lenguaje.

1° Algoritmo: es el que asocia a cada regla de descripción de una E.R, un autómata que la reconoce. El problema es que se generan autómatas muy grandes y es necesario luego minimizarlos para poder trabajar con ellos

2° Algoritmo: esta basado en el concepto de derivada.

Teorema de Kleene: Solución al problema de Síntesis

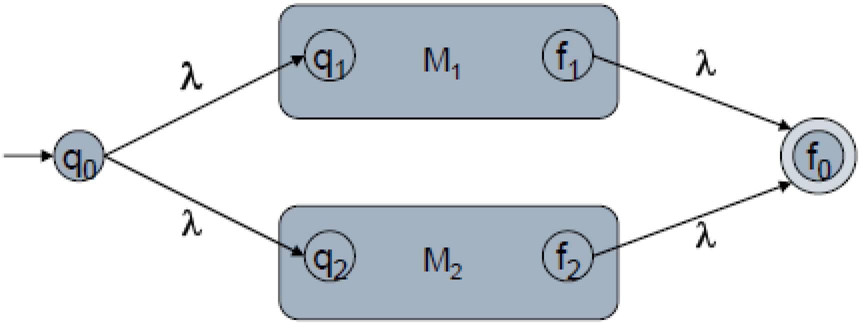
Algoritmo Recursivo: asocia a cada E.R. un autómata que reconoce el lenguaje descrito por dicha E.R.

Expresión regular α	Autómata
λ	
$\varnothing$	
a	

Teorema de Kleene: Solución al problema de Síntesis

$\alpha = \alpha_1 + \alpha_2$

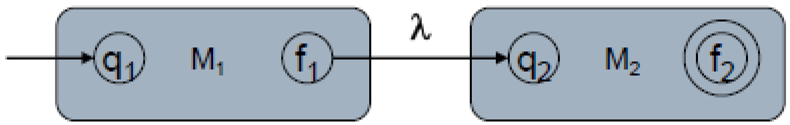
A partir de M_1 y M_2 construimos otro autómata A(la unión), el autómata que acepta el mismo lenguaje es:



Teorema de Kleene: Solución al problema de Síntesis

$\alpha = \alpha_1 . \alpha_2$

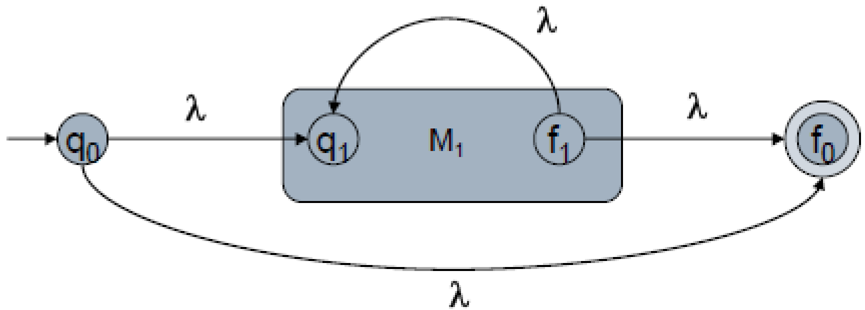
A partir de M_1 y M_2 construimos otro autómata A(la concatenación), el autómata que acepta el mismo lenguaje es:



Teorema de Kleene: Solución al problema de Síntesis

$\alpha = (\alpha_1)^*$

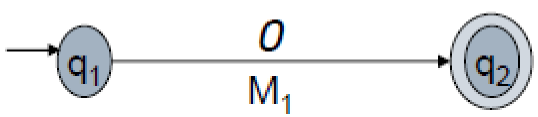
A partir de M_1 construimos otro autómata A(la clausura), el autómata que acepta el mismo lenguaje es:



Teorema de Kleene: Solución al problema de Síntesis

Ejemplo: AF construido para la expresión regular 01* + 1:

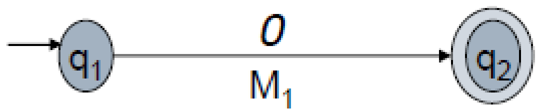
M_1 representa el autómata para la expresión regular 0



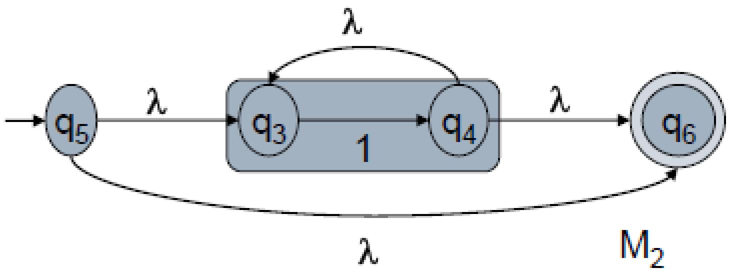
Teorema de Kleene: Solución al problema de Síntesis

Ejemplo: AF construido para la expresión regular 01* + 1:

M_1 representa el autómata para la expresión regular 0



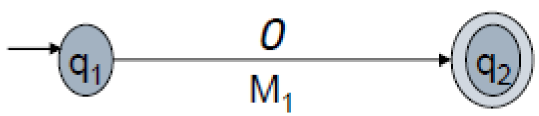
M_2 representa el autómata para la expresión regular 1*



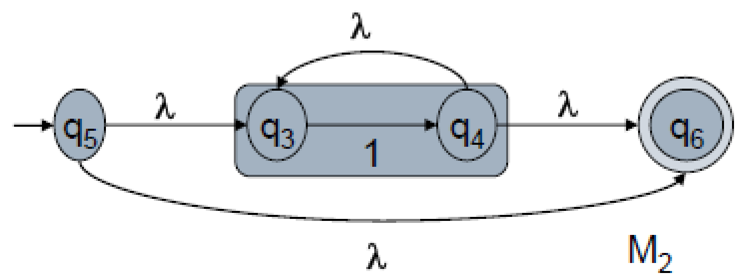
Teorema de Kleene: Solución al problema de Síntesis

Ejemplo: AF construido para la expresión regular 01* + 1:

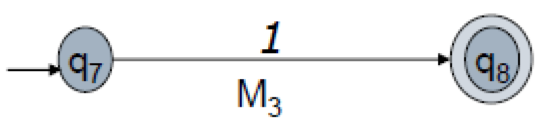
M_1 representa el autómata para la expresión regular 0



M_2 representa el autómata para la expresión regular 1*



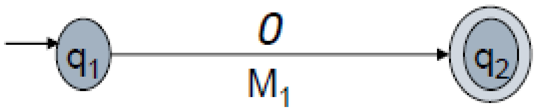
M_3 representa el autómata para la expresión regular 1



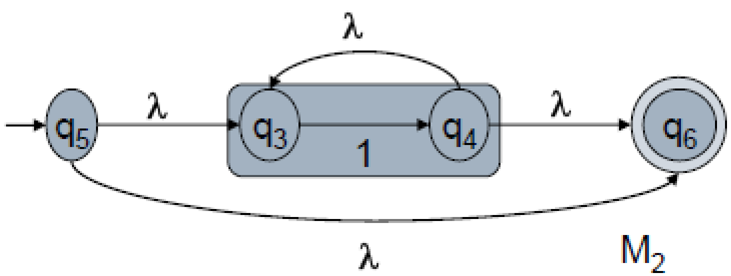
Teorema de Kleene: Solución al problema de Síntesis

Ejemplo: AF construido para la expresión regular 01* + 1:

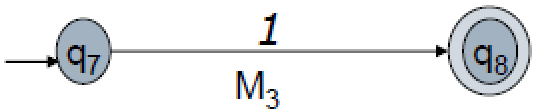
0:



1*:



1:



En el Autómata final se integran los autómatas para:

- la concatenación (0 con 1*)
- y la suma de expresiones regulares **01* + 1**