5. Lenguajes Regulares

Grado Ingeniería Informática
Teoría de Autómatas y Lenguajes Formales

AUTÓMATAS FINITOS Y G3

Gramática asociada a un AF

Sea el AF, A = (Σ, Q, q0, f, F), existe una G3 LD tal que

L(G3LD) = L(A)

Es decir, el lenguaje que genera la gramática es el mismo que reconoce el Autómata

Veamos como se obtiene la gramática G={ΣT, ΣN, S, P} a partir del AF= {Q, Σ, q0, f, F}.

3

Gramática asociada a un AF

Se construye la **gramática G3LD** (G={Σ_T, Σ_N, S, P}) de la siguiente forma, **a partir del Autómata** (AF= {Σ, Q, q_0, f, F}):

- $\Sigma_T = \Sigma$; $\Sigma_N = Q$; $S = q_o$
- P= { … }
  1. transición f(p,a) = q → si q ∉ F→ **p::= a q**
  2. Si q ∈ F y f(p,a) = q → **p::= a** y **p::= a q**
  3. Si q_0 ∈ F → **q_0 ::= λ (es axioma para dar** λ)
  4. si f(p, λ) = q → si q ∉ F → p ::= q (redenominación);
  5. q ∈ F y f(p, λ) = q → p ::= q y p::= λ (redenominación y no generativa)

4

AF asociado a una G3 (cuando es LD)

De AF → G3: Ejemplo

Sea el AF descrito por la siguiente tabla, hallar la G3LD que genera el lenguaje por ella descrito. Comprobar que los lenguajes son iguales

	0	1
→A	A	C
B	A	C
C*	C	B

5

AF asociado a una G3LD

Sea la G3LD, G=(Σ_T, Σ_N S, P), existe un AF, A, tal que:

L(G3LD) = L(A)

Cómo se construye **AF a partir de G3LD**:

- $\Sigma = \Sigma_T$
- $Q = \Sigma_N \cup \{F\}$, con $F \notin \Sigma_N$
- $q_0 = S$
- F = {F}
- f:

  Si A ::= a B → f(A,a) = B

  Si A ::= a → f(A,a) = F

  Si S ::= λ → f(S, λ) = F (equivalente a hacer $q_0 = S \in F$)

6

AF asociado a una G3 (cuando es LD)

Se ha visto el procedimiento para obtener el AF que aceptaba el lenguaje descrito por una G3LD, sin embargo, ese procedimiento no siempre conduce a un AFD.

Lo habitual es: G3LD → AFND → AFD

1. Ejemplo: Sea la G3LD hallar el AF correspondiente.

G = ({d,c}, {A,S,T}, A, {A ::= cS, S::= d|cS|dT, T::= dT|d})

AF asociado a una G3

¿Y si queremos obtener un AF a partir de una G3LI?

G3LI → G3LD → AF

¿Y si queremos obtener una G3LI a partir de un AF?

AF → G3LD → G3LI

EXPRESIONES REGULARES

- Definición
- Expresiones y Lenguajes Regulares
- Equivalencias
- Teoremas

Definición de ER (I)

" Metalenguaje para expresar el conjunto de palabras aceptadas por un AF (es decir, para expresar lenguajes de tipo 3 o regulares)"

Kleene, 1956

Definición de ER(I)

Ejemplo

Dado el alfabeto Σ= {0,1},

La ER 0*10* es una palabra del metalenguaje que representa las infinitas palabras del lenguaje regular formado por un 1, precedido y seguido de 0, 1 o infinitos 0s.

El lenguaje Σ* puede representarse mediante la ER:

(0+1)*

El lenguaje {01, 101} puede representarse mediante la ER:

01 + 101

La ER 1(1+0)* representa todas las cadenas que empiezan por el símbolo 1.

Definición de ER(II)

Dados los elementos/símbolos:

Σ, $\varnothing$ (lenguaje vacío), λ (palabra vacía)

y las operaciones:

+ (unión), •(concatenación), *(cierre o clausura)

se cumple que:

- $\varnothing$ es una ER
- λ es una ER
- cualquier $a \in \Sigma$ es una ER
- si α y β son EERR entonces $\alpha+\beta$ y $\alpha \bullet \beta$ son EERR
- si α es una ER entonces α^* es una ER, donde $\alpha^* = \bigcup_{i=0}^{\infty} \alpha^i$

13

Definición de ER(III)

Solo son EERR las que se obtienen de aplicar las reglas anteriores **un número finito de veces** sobre símbolos de Σ, $\varnothing$, λ

La prioridad de las operaciones es la siguiente:

$* > \bullet > +$

14

- Definición
- Expresiones y Lenguajes Regulares
- Equivalencias
- Teoremas

15

EERR y LR

Cada Expresión Regular (ER) describe o expresa un lenguaje regular

A cada ER α, se le asocia un subconjunto de Σ^*, $L(\alpha)$, que es el LR descrito por α. Este lenguaje se define con:

- si $\alpha = \varnothing$, $L(\alpha) = \varnothing$
- si $\alpha = \lambda$, $L(\alpha) = \{\lambda\}$
- si $\alpha = a$, $a \in \Sigma$, $L(\alpha) = \{a\}$
- si α y β son EERR $\Rightarrow L(\alpha + \beta) = L(\alpha) \cup L(\beta)$
- si α y β son EERR $\Rightarrow L(\alpha \bullet \beta) = L(\alpha)\, L(\beta)$
- si α^* es una ER $\Rightarrow L(\alpha^*) = L(\alpha)^*$

16

- Definición
- Expresiones y Lenguajes Regulares
- Equivalencias
- Teoremas

Equivalencia de EERR (I)

Dos EERR son equivalentes, $\alpha = \beta$, si describen el mismo lenguaje regular, si $L(\alpha) = L(\beta)$

Se cumple:

1) $(\alpha + \beta) + \sigma = \alpha + (\beta + \sigma)$ (+ es asociativa)
2) $\alpha + \beta = \beta + \alpha$ (+ es conmutativa)
3) $(\alpha \bullet \beta) \bullet \sigma = \alpha \bullet (\beta \bullet \sigma)$ (• es asociativa)
4) $\alpha \bullet (\beta + \sigma) = (\alpha \bullet \beta) + (\alpha \bullet \sigma)$ (+ es distributiva
   $(\beta + \sigma) \bullet \alpha = (\beta \bullet \alpha) + (\sigma \bullet \alpha)$ respecto de •)
5) $\alpha \bullet \lambda = \lambda \bullet \alpha = \alpha$ (• tiene elemento neutro)
6) $\alpha + \varnothing = \varnothing + \alpha = \alpha$ (+ tiene elemento neutro)
7) $\lambda^* = \lambda$
8) $\alpha \bullet \varnothing = \varnothing \bullet \alpha = \varnothing$

Equivalencia de EERR (II)

9) $\varnothing^* = \lambda$

10) $\alpha^* \bullet \alpha^* = \alpha^*$

11) $\alpha \bullet \alpha^* = \alpha^* \bullet \alpha$

12) $(\alpha^*)^* = \alpha^*$ (IMPORTANTE)

13) $\alpha^* = \lambda + \alpha + \alpha^2 + .. + \alpha^n + \alpha^{n+1}. \alpha^*$

14) $\alpha^* = \lambda + \alpha \bullet \alpha^*$ (13 con n=0) (IMPORTANTE)

15) $\alpha^* = (\lambda + \alpha)n\text{-}1 + \alpha n \bullet \alpha^*$ (de 14, sustituyendo)

16) Sea f una función, $f{:}E^n{}_\Sigma \rightarrow E_\Sigma$ se verifica:

$$f(\alpha, \beta, \ldots, \sigma) + (\alpha + \beta + \ldots + \sigma)^* = (\alpha + \beta + \ldots + \sigma)^*$$

17) Sea f una función, $f{:}E^n{}_\Sigma \rightarrow E_\Sigma$ se verifica:

$$(f(\alpha^*, \beta^*, \ldots, \sigma^*))^* = (\alpha + \beta + \ldots + \sigma)^*$$

19

Equivalencia de EERR (III)

18) $(\alpha^* + \beta^*)^* = (\alpha^* \bullet \beta^*)^* = (\alpha + \beta)^*$ (IMPORTANTE)

19) $(\alpha \bullet \beta)^* \bullet \alpha = \alpha \bullet (\beta \bullet \alpha)^*$

20) $(\alpha^* \bullet \beta)^* \bullet \alpha^* = (\alpha + \beta)^*$

21) $(\alpha^* \bullet \beta)^* = \lambda + (\alpha + \beta)^* \bullet \beta$ (de 14 con 20)

22) Reglas de Inferencia:

Dadas tres EERR (L, A y B), sea la ecuación

L = AL + B,

donde $\lambda \notin A$, entonces se verifica que

L = A*B

20

- Definición
- Expresiones y Lenguajes Regulares
- Equivalencias
- Teoremas

21

Teoremas de análisis y síntesis de Kleene

1. **Teorema de análisis de Kleene**

Todo lenguaje aceptado por un AF es un lenguaje regular.

Solución al problema de análisis:

Encontrar el lenguaje asociado a un determinado AF: "Dado un AF, A, encontrar la ER que describe L(A)".

1. **Teorema de síntesis de Kleene**

Todo lenguaje regular es el lenguaje aceptado por un AF.

Solución al problema de síntesis:

Encontrar un reconocedor para un lenguaje regular dado: *"Dada una ER que representa a un lenguaje regular, construir un AF que acepte ese lenguaje regular".*

22

1. Solución al problema de análisis. Ecuaciones características.

Problema Análisis: AF -> ER

Resolución:

Dado un AF, escribir las ecuaciones características de cada uno de sus estados, resolverlas y obtener la ER buscada.

Solución al problema de análisis. Ecuaciones características.

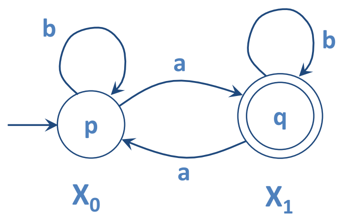
ECUACIONES CARACTERÍSTICAS:
Describen todas las cadenas que se pueden reconocer desde un estado dado

Se escribe una ecuación x_i por estado q_i

- Primer miembro: $\mathbf{x_i}$
- El segundo miembro tiene un término por cada rama que salga de q_i
  - Las ramas tienen la forma $\mathbf{a_{ij} \bullet x_j}$ donde a_{ij} es la etiqueta de la rama que une q_i con q_j, x_j es la variable correspondiente a q_j
  - **Se añade un término a_{ij} por cada rama que une qi con un estado final**
  - ~~**Se añade λ si q_i es final.**~~ **SOLO si es final sin ramas o SOLO ramas al sumidero**
  - Si de un estado q_i no sale ninguna rama, el segundo miembro será:
    - si es final: $x_i = \lambda$
    - si no es final: $x_i = \varnothing$

Solución al problema de análisis. Ecuaciones características.

Ecuaciones Características de un AF – Ejemplo 1:



El AF tiene 2 estados, por lo que tendrá 2 ecuaciones características:

Conjunto de palabras que permiten pasar desde el estado *p* a un estado final.

$X_0 = b\ X_0 + a\ X_1 + a$

$X_1 = b\ X_1 + a\ X_0 + b$

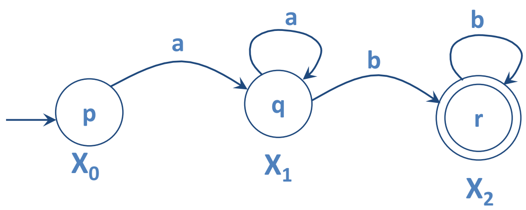
Porque *q* es un estado final

Porque *q* es un estado final

25

Solución al problema de análisis. Ecuaciones características.

Ejemplo 2:



Ecuaciones Características:

$X_0 = a\ X_1$

$X_1 = b\ X_2 + a\ X_1 + b$

$X_2 = b\ X_2 + b$

26

Algoritmo de resolución problema de Análisis.

1. Escribir las ecuaciones características del AF
2. Resolverlas
3. Si el estado inicial es q_0, X_0 nos da el conjunto de cadenas que conducen desde q_0 a q_f y por tanto el lenguaje aceptado por el AF

Solución de las ecuaciones características.

La Ecuación Característica de la forma: X = AX + B, donde:

X: conjunto de cadenas que permiten pasar de q_i a $q_f \in F$

A: conjunto de cadenas que permiten, partiendo de un estado q, llegar a q.

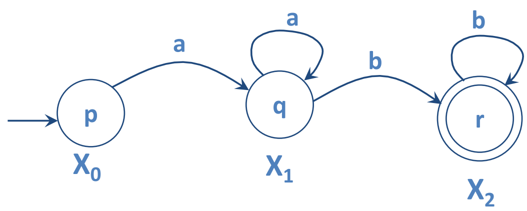
B: conjunto de cadenas que permiten llegar al estado final, sin volver a pasar por el q_i de partida.

$\Downarrow$ (solución de Arden o reducción al absurdo)

La solución es: **X = A* • B**

Solución al problema de análisis. Ecuaciones características.

Ejemplo 1:



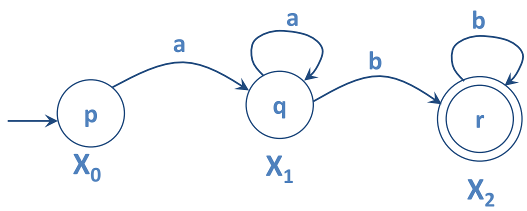
Ecuaciones Características:

$X_0 = a\, X_1$
$X_1 = b\, X_2 + a\, X_1 + b$
$X_2 = b\, X_2 + b$

29

Solución al problema de análisis. Ecuaciones características.

Ejemplo 1:



Recuerda:
L=AL+B

L=A*B

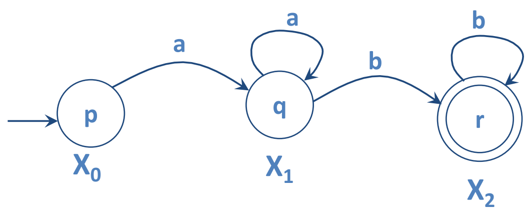
Ecuaciones Características:

$X_0 = a\, X_1$
$X_1 = b\, X_2 + a\, X_1 + b$
$X_2 = b\, X_2 + b + \lambda$

30

Solución al problema de análisis. Ecuaciones características.

Ejemplo 1:



Recuerda:

L=AL+B

L=A*B

Ecuaciones Características:

$X_0 = a\, X_1$

$X_1 = b\, X_2 + a\, X_1 + b$

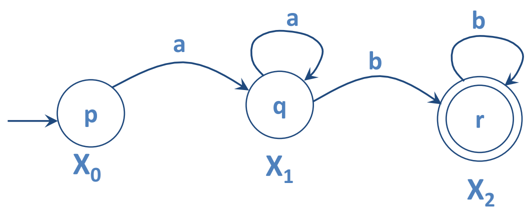
$X_2 = b\, X_2 + b$

$X_2 = b^*\, b$

31

Solución al problema de análisis. Ecuaciones características.

Ejemplo 1:



Recuerda:

L=AL+B

L=A*B

Ecuaciones Características:

$X_0 = a\, X_1$

$X_1 = b\, X_2 + a\, X_1 + b$

$X_2 = b\, X_2 + b$

$X_2 = b^*\, b$

$X_1 = bb^*\, b + aX_1 + b$

$X_1 = aX_1 + bb^*\, b + b$

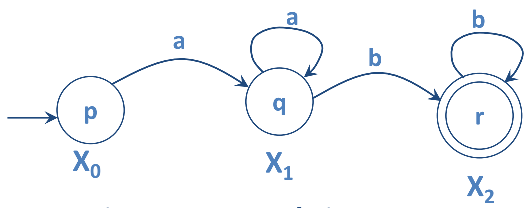
$X_1 = a^*(b\, b^*\, b + b) =$

$a^*\,(b(b^*b + \lambda\,)) = a^*bb^*$

32

Solución al problema de análisis. Ecuaciones características.

Ejemplo 1:



Recuerda:
L=AL+B

L=A*B

Ecuaciones Características:

$X_0 = a\, X_1$
$X_1 = b\, X_2 + a\, X_1 + b$
$X_2 = b\, X_2 + b$

$X_1 = a^*bb^*$
$X_0 = a\,(X_1) = aa^*bb^*$

33

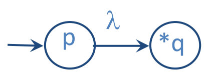
2. Problema de Síntesis: Algoritmo Recursivo (I)

Dada una ER que representa a un lenguaje regular, construir un AF que acepte ese lenguaje regular.

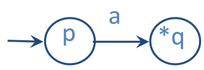
Sea α una Expresión Regular

- si $\alpha = \varnothing$, el autómata será:

- si $\alpha = \lambda$, el autómata será:



- si $\alpha = a$, $a \in \Sigma$, el autómata será:

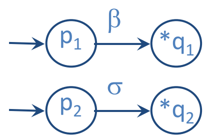


34

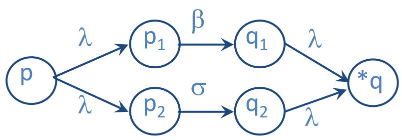
Problema de Síntesis: Algoritmo Recursivo (II)

Dada una ER que representa a un lenguaje regular, construir un AF que acepte ese lenguaje regular *(cont.)*

- si $\alpha=\beta+\sigma$, con los autómatas de β y σ



el resultado es:

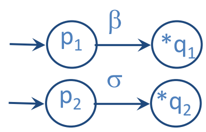


35

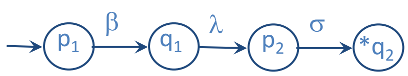
Problema de Síntesis: Algoritmo Recursivo (III)

Dada una ER que representa a un lenguaje regular, construir un AF que acepte ese lenguaje regular. (*cont.)*

- si $\alpha=\beta \bullet \sigma$, con los autómatas de β y σ



el resultado es:

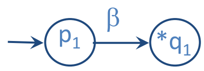


36

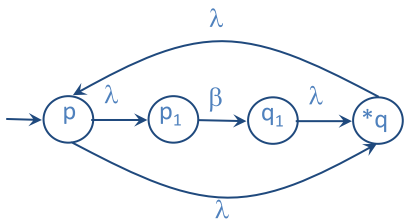
Problema de Síntesis: Algoritmo Recursivo (IV)

Dada una ER que representa a un lenguaje regular, construir un AF que acepte ese lenguaje regular. *(cont.)*

- si $\alpha=\beta^*$, con el autómata de β



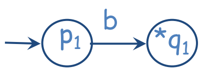
el resultado es:



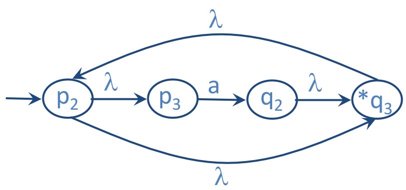
Problema de Síntesis: Algoritmo Recursivo (IV)

Ejemplo: Sea α= b • a*

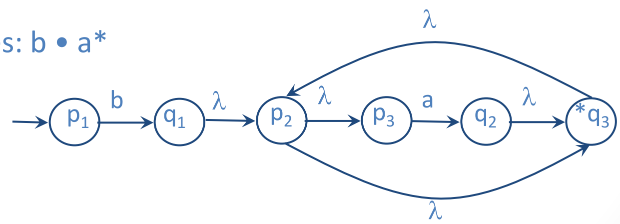
b:



a*:



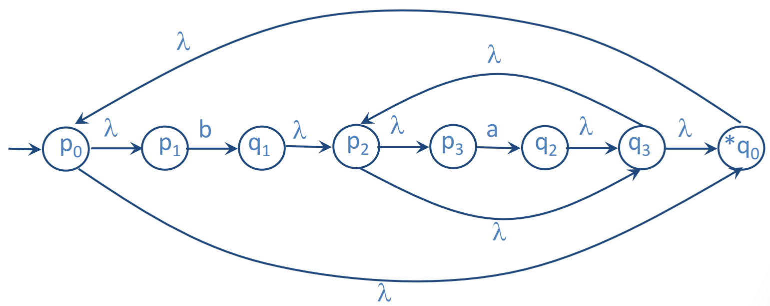
Entonces: b • a*



Problema de Síntesis: Algoritmo Recursivo (IV)

Ejemplo:

Sea α=(b • a*)*



39

2. Problema de Síntesis: Derivada de una ER.

Dada una ER, construir un AF que reconozca el lenguaje que la ER describe.

Solución: derivar la ER y obtener una G3LD y de ella un AF

Derivada de una ER: $Da(R) = \{ x \mid a \bullet x \in R \}$.

- Derivada de ER R respecto de a ∈ Σ es el conjunto de colas de todas las palabras representadas por R cuya cabeza es a.

Veamos una definición recursiva

40

Problema de Síntesis: Derivada de una ER

ER -> AF (Derivar la ER -> G3LD -> AF. $Da(R) = \{ x \mid a\bullet x \in R \}$

Derivada de una ER. ***Definición recursiva***

$\forall$ a, b $\in \Sigma$ y R,S expresiones regulares

- Da ($\varnothing$) = $\varnothing$
- Da (λ) = $\varnothing$
- Da (a) = λ, a$\in\Sigma$
- Da (b) = $\varnothing$, $\forall$ b $\neq$ a, b$\in\Sigma$
- Da (R+S) = Da (R) + Da (S)
- Da (R • S) = Da(R) • S + δ(R) • Da(S) $\forall$R
  - $\lambda\in R \Rightarrow \delta(R) = \lambda$
  - $\lambda\notin R \Rightarrow \delta(R) = \varnothing$
- Da (R*) = Da(R) • R*

41

Solución al problema de Síntesis. Derivada de una ER

- **Definición: Dab(R) = Db(Da(R))**
- A partir de la Derivada de una ER. Se obtendrá la gramática regular lineal derecha:
  - El número de derivadas distintas de una ER es finito.
    Una vez que se han obtenido todas, se puede obtener la G3
  - **Si δ (R0) = λ añadimos R0 → λ**
  - Sea Da(R) = S, con S $\neq \Phi$
    - S $\neq \lambda \Rightarrow$ R ::= aS $\in$ P
    - S = $\lambda \Rightarrow$ R ::= a $\in$ P
  - Sea δ(Da(R)) = S
    - δ(Da(R)) = $\lambda \Rightarrow$ R::= a $\in$ P
    - δ(Da(R)) = $\Phi \Rightarrow$ no se incluye ninguna regla en P
  - El axioma es R (ER de partida)
    - Σ_T = símbolos que formaban la ER de partida
    - Σ_N = letras que distinguen cada una de las derivadas distintas

42

Ejemplos. Derivada Expresiones Regulares

Obtener las G3 LD equivalentes a las ER dadas:

R = a a* b b*, Σ={a,b}

R = a a* b b* es igual que
R = a • a* • b • b*

- Da(R) = Da(a) a* b b* = a* b b*
- Db(R) = ∅
- Daa(R) = Da(a* b b*) = Da(a*) b b* + λ Da(b b*) = a*bb* = Da(R)
- Dab(R) = Db(a* b b*) = Db(a*) b b* + λ Db(b b*) = b*
- Daba(R) = Da(b*) = ∅
- Dabb(R) = Db(b*) = Db(b) b* = b* = Dab(R)

- Da(R)= a*bb* δ(Da(R))= ∅
- Daa(R)= a*bb* δ(Daa(R))= ∅
- Dab(R)= b* δ(Dab(R))= λ
- Dabb(R)= b* δ(Dabb(R))= λ

43

Ejemplos. Derivada Expresiones Regulares

- Obtención de la G3 LD que genera el Lenguaje representado por la ER:
- R0=aa*bb* R1=a*bb* R2=b*
  - Da(R0)=R1 δ(Da(R0))= ∅
  - Da(R1)=R1 δ(Da(R1))= ∅
  - Db(R1)=R2 δ(Db(R1))= λ
  - Db(R2)=R2 δ(Db(R2))= λ

- Da(R)=S ⇒ R→aS δ(Da(R))= λ ⇒ R→a
  - R0 → aR1 ----------
  - R1 → aR1 ----------
  - R1 → bR2 R1 → b
  - R2 → bR2 R2 → b

44

Bibliografía

- Libro Básico 1 Bibliografía (AAM). Enrique Alfonseca Cubero, Manuel Alfonseca Cubero, Roberto Moriyón Salomón. Teoría de autómatas y lenguajes formales. McGraw-Hill (2007).
  Apartado 7.2

- Libro Básico 2 Bibliografía (HMU). John E. Hopcroft, Rajeev Motwani, Jeffrey D.Ullman. Introducción a la teoría de autómatas, lenguajes y computación (3ª edición). Ed, Pearson Addison Wesley.
  Tema 3

- Libro Básico 4 Bibliografía (AAM). Manuel Alfonseca, Justo Sancho, Miguel Martínez Orga. Teoría de lenguajes, gramáticas y autómatas. Publicaciones R.A.E.C. 1997
  Tema 7

45