

# 5. Lenguajes Regulares

Grado Ingeniería Informática  
Teoría de Autómatas y Lenguajes Formales



AUTÓMATAS FINITOS Y G3



[ 2 ]

## Gramática asociada a un AF

Sea el AF,  $A = (\Sigma, Q, q_0, f, F)$ , 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  $G = \{\Sigma_T, \Sigma_N, S, P\}$  a partir del AF  $= \{Q, \Sigma, q_0, f, F\}$ .

[ 3 ]



## Gramática asociada a un AF

Se construye la **gramática G3LD** ( $G = \{\Sigma_T, \Sigma_N, S, P\}$ ) de la siguiente forma, **a partir del Autómata** ( $AF = \{\Sigma, Q, q_0, f, F\}$ ):

- $\Sigma_T = \Sigma$  ;  $\Sigma_N = Q$  ;  $S = q_0$
- $P = \{ \dots \}$ 
  1. transición  $f(p, a) = q \rightarrow$  si  $q \notin F \rightarrow p ::= a q$
  2. Si  $q \in F$  y  $f(p, a) = q \rightarrow p ::= a$  y  $p ::= a q$
  3. Si  $q_0 \in F \rightarrow q_0 ::= \lambda$  (es axioma para dar  $\lambda$ )
  4. si  $f(p, \lambda) = q \rightarrow$  si  $q \notin F \rightarrow p ::= q$  (redenominación);
  5.  $q \in F$  y  $f(p, \lambda) = q \rightarrow p ::= q$  y  $p ::= \lambda$  (redenominación y no generativa)

[ 4 ]



## AF asociado a una G3 (cuando es LD)

### De AF $\rightarrow$ 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
$\rightarrow A$	A	C
B	A	C
$C^*$	C	B

[ 5 ]



## AF asociado a una G3LD

Sea la G3LD,  $G=(\Sigma_T, \Sigma_N, S, P)$ , 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  $A ::= a B \rightarrow f(A, a) = B$
  - Si  $A ::= a \rightarrow f(A, a) = F$
  - Si  $S ::= \lambda \rightarrow f(S, \lambda) = F$  (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:  $G3LD \rightarrow AFND \rightarrow AFD$

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

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

[ 7 ]



## AF asociado a una G3

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

$$G3LI \rightarrow G3LD \rightarrow AF$$

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

$$AF \rightarrow G3LD \rightarrow G3LI$$

[ 8 ]



## EXPRESIONES REGULARES

[ 9 ]



[ 10 ]



## 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*

[ 11 ]



## Definición de ER(I)

### Ejemplo

Dado el alfabeto  $\Sigma = \{0,1\}$ ,

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  $\Sigma^*$  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.

[ 12 ]



## Definición de ER(II)

Dados los elementos/símbolos:

$\Sigma$ ,  $\emptyset$  (lenguaje vacío),  $\lambda$  (palabra vacía)

y las operaciones:

$+$  (unión),  $\bullet$  (concatenación),  $*$  (cierre o clausura)

se cumple que:

- $\emptyset$  es una ER
- $\lambda$  es una ER
- cualquier  $a \in \Sigma$  es una ER
- si  $\alpha$  y  $\beta$  son EERR entonces  $\alpha + \beta$  y  $\alpha \bullet \beta$  son EERR
- si  $\alpha$  es una ER entonces  $\alpha^*$  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  $\Sigma$ ,  $\emptyset$ ,  $\lambda$

La prioridad de las operaciones es la siguiente:

$$* > \bullet > +$$

[ 14 ]



Definición

Expresiones y Lenguajes Regulares

Equivalencias

Teoremas

[ 15 ]

Universidad Carlos III de Madrid

CC BY-NC-SA

## EERR y LR

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

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

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

[ 16 ]

Universidad Carlos III de Madrid

CC BY-NC-SA





Definición

Expresiones y Lenguajes Regulares

Equivalencias

Teoremas

[ 17 ]

Universidad Carlos III de Madrid

CC BY-NC-SA

## 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)  $(\alpha \cdot \beta) \cdot \sigma = \alpha \cdot (\beta \cdot \sigma)$  ( $\cdot$  es asociativa)
- 4)  $\alpha \cdot (\beta + \sigma) = (\alpha \cdot \beta) + (\alpha \cdot \sigma)$  (+ es distributiva)
- 5)  $(\beta + \sigma) \cdot \alpha = (\beta \cdot \alpha) + (\sigma \cdot \alpha)$  respecto de  $\cdot$
- 6)  $\alpha \cdot \lambda = \lambda \cdot \alpha = \alpha$  ( $\cdot$  tiene elemento neutro)
- 7)  $\alpha + \emptyset = \emptyset + \alpha = \alpha$  (+ tiene elemento neutro)
- 8)  $\lambda^* = \lambda$
- 9)  $\alpha \cdot \emptyset = \emptyset \cdot \alpha = \emptyset$

[ 18 ]

Universidad Carlos III de Madrid

CC BY-NC-SA

## Equivalencia de EERR (II)

9)  $\emptyset^* = \lambda$

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

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

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

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

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

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

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

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

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

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



[ 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)

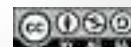
22) Reglas de Inferencia:

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 ]



## 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 $\rightarrow$ ER

#### Resolución:

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

[ 23 ]



## Solución al problema de análisis. Ecuaciones características.

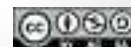
### ECUACIONES CARACTERÍSTICAS:

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

Se escribe una ecuación  $x_i$  por estado  $q_i$

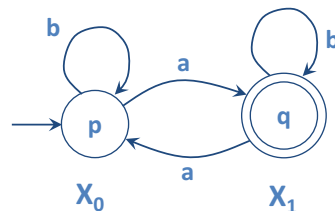
- Primer miembro:  $x_i$
- El segundo miembro tiene un término por cada rama que salga de  $q_i$ 
  - Las ramas tienen la forma  $a_{ij} \bullet x_j$  donde  $a_{ij}$  es la etiqueta de la rama que une  $q_i$  con  $q_j$ ,  $x_j$  es la variable correspondiente a  $q_j$
  - Se añade un término  $a_{ij}$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

[ 24 ]



## 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

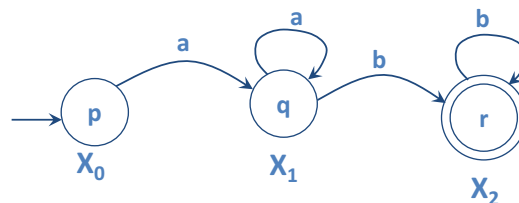
Porque  $q$  es un estado final

( 25 )



## Solución al problema de análisis. Ecuaciones características.

Ejemplo 2:



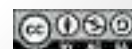
**Ecuaciones Características:**

$$X_0 = a X_1$$

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

$$X_2 = b X_2 + b$$

( 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

[ 27 ]



## 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  $q_f \in F$

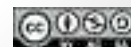
$A$ : conjunto de cadenas que permiten, partiendo de un estado  $q$ , llegar a  $q$ .

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

⇓ (solución de Arden o reducción al absurdo)

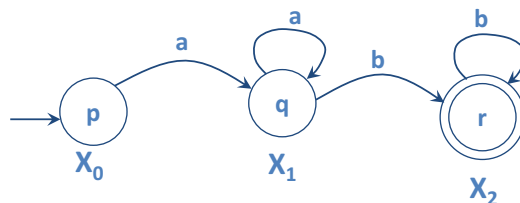
La solución es:  $X = A^* \cdot B$

[ 28 ]



## Solución al problema de análisis. Ecuaciones características.

Ejemplo 1:



**Ecuaciones Características:**

$$X_0 = a X_1$$

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

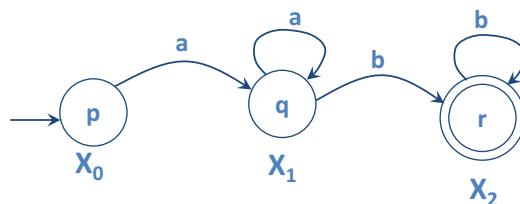
$$X_2 = b X_2 + b + \lambda$$

[ 29 ]



## Solución al problema de análisis. Ecuaciones características.

Ejemplo 1:



Recuerda:

$$L = AL + B$$

$$L = A^*B$$

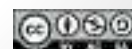
**Ecuaciones Características:**

$$X_0 = a X_1$$

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

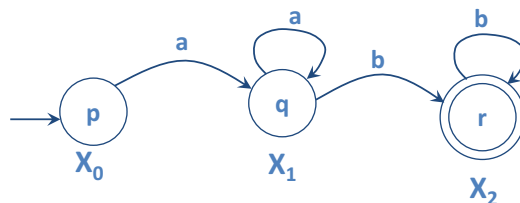
$$X_2 = b X_2 + b + \lambda$$

[ 30 ]



## Solución al problema de análisis. Ecuaciones características.

Ejemplo 1:



Recuerda:

$$L = AL + B$$

$$L = A^*B$$

**Ecuaciones Características:**

$$X_0 = a X_1$$

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

$$X_2 = b X_2 + b + \lambda$$

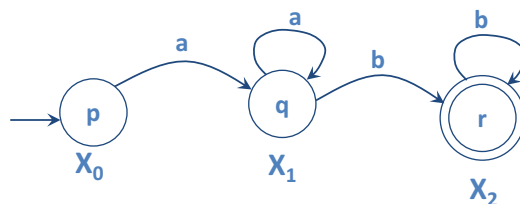
$$X_2 = b^* (b + \lambda)$$

[ 31 ]



## Solución al problema de análisis. Ecuaciones características.

Ejemplo 1:



Recuerda:

$$L = AL + B$$

$$L = A^*B$$

**Ecuaciones Características:**

$$X_0 = a X_1$$

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

$$X_2 = b X_2 + b + \lambda$$

$$X_2 = b^* (b + \lambda)$$

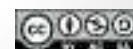
$$X_1 = b b^* (b + \lambda) + a X_1 + b$$

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

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

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

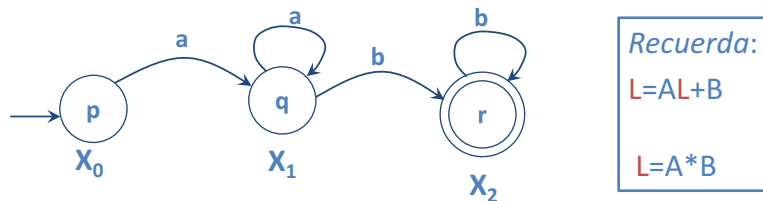
[ 32 ]





## Solución al problema de análisis. Ecuaciones características.

Ejemplo 1:



Recuerda:

$$L = AL + B$$

$$L = A^*B$$

**Ecuaciones Características:**

$$X_0 = a X_1$$

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

$$X_2 = b X_2 + b + \lambda$$

$$X_2 = b^*(b + \lambda) = b^* + b^* = b^*$$

$$X_1 = b b^* + a X_1 + b$$

$$X_1 = a X_1 + b b^* + b$$

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

$$X_0 = a a^* b b^*$$

[ 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  $\alpha$  una Expresión Regular

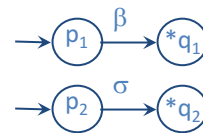
- si  $\alpha = \emptyset$ , el autómata será:  $\rightarrow p \quad *q$
- si  $\alpha = \lambda$ , el autómata será:  $\rightarrow p \xrightarrow{\lambda} *q$
- si  $\alpha = a, a \in \Sigma$ , el autómata será:  $\rightarrow p \xrightarrow{a} *q$

[ 34 ]

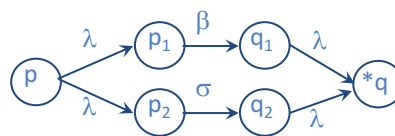
## Problema de Síntesis: Algoritmo Recursivo (II)

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  $\beta$  y  $\sigma$



el resultado es:

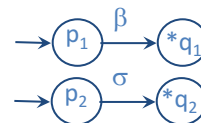


[ 35 ]

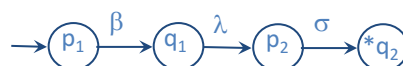
## Problema de Síntesis: Algoritmo Recursivo (III)

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

- si  $\alpha = \beta \cdot \sigma$ , con los autómatas de  $\beta$  y  $\sigma$



el resultado es:



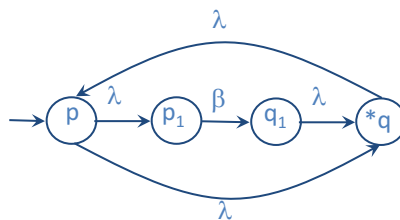
[ 36 ]

## 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  $\beta \rightarrow \begin{array}{c} \text{---} \text{p}_1 \xrightarrow{\beta} \text{---} \text{q}_1 \end{array}$

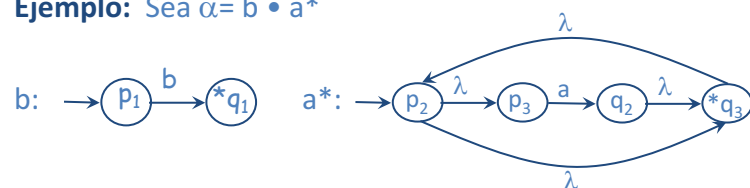
el resultado es:



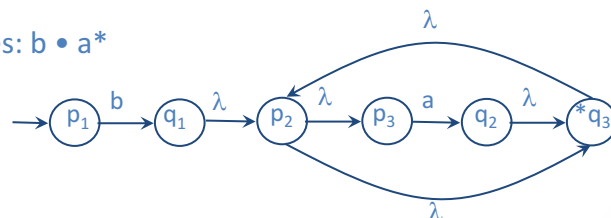
[ 37 ]

## Problema de Síntesis: Algoritmo Recursivo (IV)

**Ejemplo:** Sea  $\alpha = b \cdot a^*$



Entonces:  $b \cdot a^*$

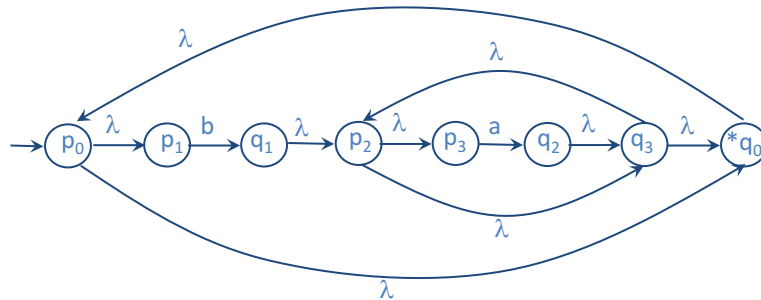


[ 38 ]

## Problema de Síntesis: Algoritmo Recursivo (IV)

**Ejemplo:**

Sea  $\alpha = (b \bullet a^*)^*$



[ 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:  $D_a(R) = \{ x \mid a \bullet x \in R \}$ .

- Derivada de ER R respecto de  $a \in \Sigma$  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  $\rightarrow$  AF (Derivar la ER  $\rightarrow$  G3LD  $\rightarrow$  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

- $Da(\emptyset) = \emptyset$
- $Da(\lambda) = \emptyset$
- $Da(a) = \lambda, \quad a \in \Sigma$
- $Da(b) = \emptyset, \quad \forall b \neq a, b \in \Sigma$
- $Da(R+S) = Da(R) + Da(S)$
- $Da(R \cdot S) = Da(R) \cdot S + \delta(R) \cdot Da(S) \quad \forall R$ 
  - $\lambda \in R \Rightarrow \delta(R) = \lambda$
  - $\lambda \notin R \Rightarrow \delta(R) = \emptyset$
- $Da(R^*) = Da(R) \cdot R^*$

[ 41 ]



Universidad  
Carlos III de Madrid  
Instituto de Ingeniería de Sistemas y Robotics



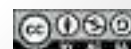
## Solución al problema de Síntesis. Derivada de una ER

- **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
  - Sea  $Da(R) = S$ , con  $S \neq \Phi$ 
    - $S \neq \lambda \Rightarrow R ::= aS \in P$
    - $S = \lambda \Rightarrow R ::= a \in P$
  - Sea  $\delta(Da(R)) = S$ 
    - $\delta(Da(R)) = \lambda \Rightarrow R ::= a \in P$
    - $\delta(Da(R)) = \Phi \Rightarrow$  no se incluye ninguna regla en P
  - El axioma es R (ER de partida)
    - $\Sigma_T$  = símbolos que formaban la ER de partida
    - $\Sigma_N$  = letras que distinguen cada una de las derivadas distintas

[ 42 ]



Universidad  
Carlos III de Madrid  
Instituto de Ingeniería de Sistemas y Robotics



## Ejemplos. Derivada Expresiones Regulares

Obtener las G3 LD equivalentes a las ER dadas:

$R = a a^* b b^*$ ,  $\Sigma = \{a, b\}$

$R = a a^* b b^*$  es igual que  
 $R = a \cdot a^* \cdot b \cdot b^*$

- $Da(R) = Da(a) a^* b b^* = a^* b b^*$
- $Db(R) = \emptyset$
- $Daa(R) = Da(a^* b b^*) = Da(a^*) b b^* + \lambda Da(b b^*) = a^* b b^* = Da(R)$
- $Dab(R) = Db(a^* b b^*) = Db(a^*) b b^* + \lambda Db(b b^*) = b^*$
- $Daba(R) = Da(b^*) = \emptyset$
- $Dabb(R) = Db(b^*) = Db(b) b^* = b^* = Dab(R)$

- |                        |                              |
|------------------------|------------------------------|
| - $Da(R) = a^* b b^*$  | $\delta(Da(R)) = \emptyset$  |
| - $Daa(R) = a^* b b^*$ | $\delta(Daa(R)) = \emptyset$ |
| - $Dab(R) = b^*$       | $\delta(Dab(R)) = \lambda$   |
| - $Dabb(R) = b^*$      | $\delta(Dabb(R)) = \lambda$  |



[ 43 ]

## Ejemplos. Derivada Expresiones Regulares

- Obtención de la G3 LD que genera el Lenguaje representado por la ER:

- |                    |                              |            |
|--------------------|------------------------------|------------|
| • $R0 = aa^* bb^*$ | $R1 = a^* bb^*$              | $R2 = b^*$ |
| • $Da(R0) = R1$    | $\delta(Da(R0)) = \emptyset$ |            |
| • $Da(R1) = R1$    | $\delta(Da(R1)) = \emptyset$ |            |
| • $Db(R1) = R2$    | $\delta(Db(R1)) = \lambda$   |            |
| • $Db(R2) = R2$    | $\delta(Db(R2)) = \lambda$   |            |

- |  |   |
|--|---|
| • $Da(R) = S \Rightarrow R \rightarrow aS$ | $\delta(Da(R)) = \lambda \Rightarrow R \rightarrow a$ |
| • $R0 \rightarrow aR1$                     | -----   |
| • $R1 \rightarrow aR1$                     | -----   |
| • $R1 \rightarrow bR2$                     | $R1 \rightarrow b$                                    |
| • $R2 \rightarrow bR2$                     | $R2 \rightarrow b$                                    |



[ 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 ]

