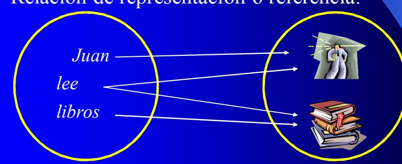
Sesión 3

Expresiones y Lenguajes Regulares

Sintaxis y Semántica

- En su uso normal, las expresiones lingüística hacen referencia a objetos individuales, así como a sus propiedades y relaciones
- Los símbolos lingüísticos son objetos *sintácticos*
- Relación de representación o referencia:



El Languaje refiere al Mundo

Sintaxis y Semántica

- Si el lenguaje es el objeto de estudio, necesitamos un lenguaje para hablar del lenguaje
- Los conjuntos de cadenas (i.e los lenguajes) se convierten en objetos *semánticos*
- La función de interpretación o referencia:

$\mathbf{0 + 1} \rightarrow \{0, 1\}$

$\mathbf{1^*} \rightarrow \{\Lambda, 1, 11,\ldots\}$

El Languaje refiere al Mundo

El lenguaje de las Expresiones Regulares

- Sintaxis
  - Constantes básicas (para un Σ dado)
    - Φ es una expresión regular (*ER*)
    - Λ es una *ER*
    - Si $\boldsymbol{a} \in \Sigma$ entonces $\boldsymbol{a}$ es una *ER*
    - Una variable, una italica mayúscula (e.g. L), es una *ER*
  - Reglas de composición:
    - Si E & F son *ER* entonces $E + F$ es una *ER* (unión)
    - Si E & F son *ER* entonces EF es una *ER* (concatenación)
    - Si E es una *ER* entonces E^* es una *ER* (cerradura)
    - Si E es una *ER* entonces (E) es una *ER* (introducción de paréntesis)
  - Sólo las expresiones construidas por un número FINITO de aplicaciones de estas reglas son *ER*

Una formulación alternativa

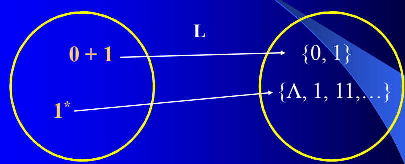
- Sintaxis
  - Constantes básicas (para un Σ dado)
    - Φ es una expresión regular (*ER*)
    - Λ es una *ER*
    - Si $\boldsymbol{a} \in \Sigma$ entonces $\boldsymbol{a}$ es una *ER*
    - Una variable, una italica mayúscula (e.g. L), is a *ER*
  - Reglas de composición (Los paréntesis son obligarorios):
    - Si E & F son *ER* entonces $(E + F)$ es una *ER* (unión)
    - Si E & F son *ER* entonces (EF) es una *ER* (concatenación)
    - Si E es una *ER* entonces (E^*) es una *ER* (cerradura)
  - Sólo las expresiones construidas por un número FINITO de aplicaciones de estas reglas son *ER*

El lenguaje de las Expresiones Regulares

- Semántica:
  - Sea *ER* el conjunto de todas las espresiones regulares sobre Σ, R el conjunto de lenguajes regulares y **L** una función de interpretación de *ER* a R
  - Interpretación de constantes básicas:
    - $\mathbf{L}(\Phi)$ es Φ (i.e. el lenguaje vacío)
    - $\mathbf{L}(\Lambda)$ es $\{\Lambda\}$ (i.e. el lenguaje con la cadena vacía)
    - Si $\boldsymbol{a} \in \Sigma$ entonces $\mathbf{L}(\boldsymbol{a})$ es $\{a\}$ {i.e. el lenguaje con a}
    - $\mathbf{L}(L)$ es un lenguaje arbitrario
  - Interpretación de expresiones compuestas:
    - $\mathbf{L}(E+F)$ es la unión de $\mathbf{L}(E)$ & $\mathbf{L}(F)$
    - $\mathbf{L}(EF)$ o $\mathbf{L}(E.F)$ es la concatención de $\mathbf{L}(E)$ & $\mathbf{L}(F)$
    - $\mathbf{L}(E^*)$ es $(\mathbf{L}(E))^*$ (i.e. la cerradura de $\mathrm{L}(E)$)
    - $\mathbf{L}((E))$ es $\mathbf{L}(E)$ (i.e. el mismo lenguaje)

Sintaxis y Semántica

- La función de referencia o interpretación **L**:



- La interpretación de una expresión compuesta es función de:
  - La interpretación de sus partes
  - Su forma de compósición gramatical (i.e. sintáctica)

Ejemplos

ER	Lenguaje
Λ:	$\{\Lambda\}$
0:	$\{0\}$
001:	$\{001\}$
0 + 1:	$\{0, 1\}$
0 + 10:	$\{0, 10\}$
(1 + Λ)001:	$\{1, \Lambda\}\{001\} = \{1001, 001\}$
$\mathbf{(110)^*(0 + 1)}$:	$\{110\}^*\{0, 1\}$
$\mathbf{1^*10}$:	$\{1\}^*\{10\}$

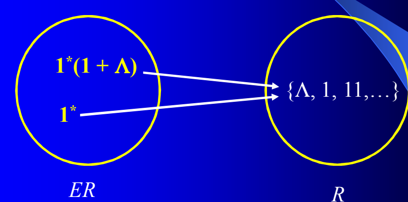
Ejemplos

- … $\mathbf{(10 + 111 + 11010)^*}$: $\{10, 111, 11010\}^*$
- $\mathbf{(0 + 10)^*((11)^* + 001 + \Lambda)}$: $\{0, 10\}^*\{\{11\}^* \cup \{001, \Lambda\}\}$
- $\mathbf{01^* + 1}$: $\{0\}\{1\}^* \cup \{1\} = \{1, 0, 01, 011, \ldots, 011\ldots1\}$
- $\mathbf{(01)^* + 1}$: $\{01\}^* \cup \{1\} = \{1, \Lambda, 01, 0101, \ldots, 0101\ldots01\}$
- $\mathbf{0(1^* + 1)}$: $\{0\}\{\{1\}^* \cup \{1\}\} = \{0, 01, 011, \ldots, 011\ldots1\}$

En particular: $\mathbf{0(1^* + 1) = 01^*}$

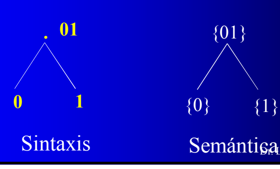
Igualdad entre Expresiones Regulares

- Dos expresiones regulares son iguales si se refieren al mismo lenguaje:



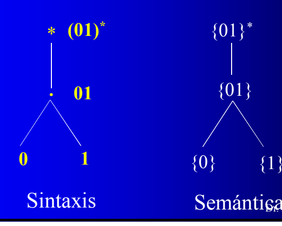
Ejemplo: El lenguaje de 0's y 1's alternados

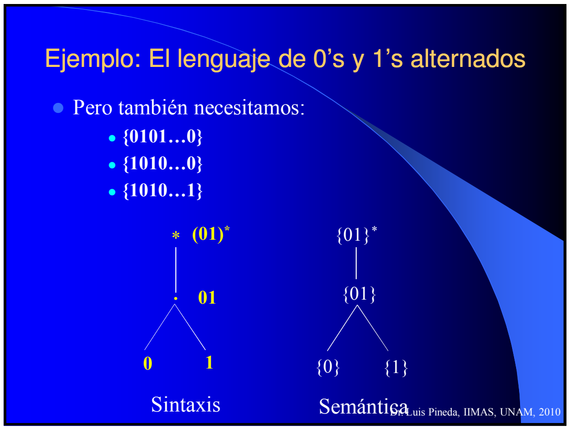
- Primero: El lenguaje $\{01\}$: **01**

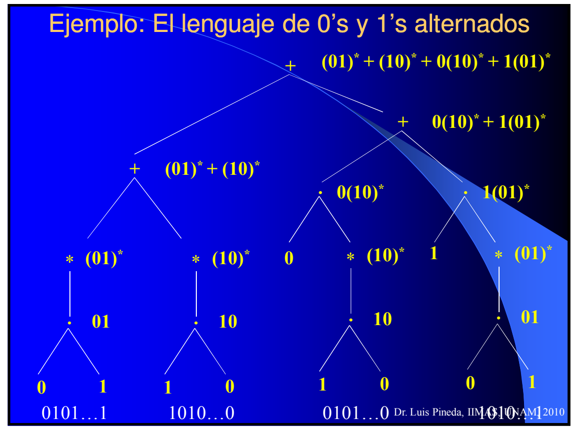


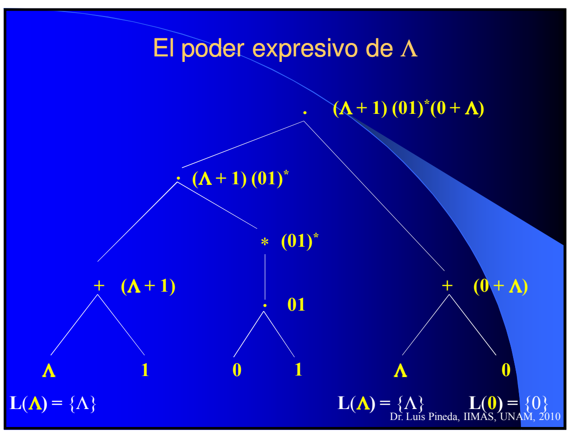
Ejemplo: El lenguaje de 0's y 1's alternados

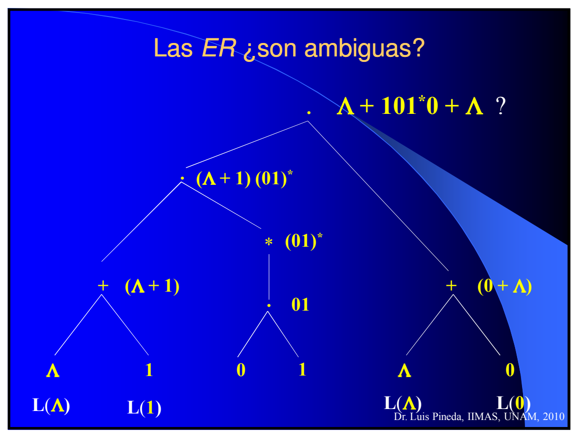
- Segundo: El lenguaje $\{01\}^*$: $\mathbf{(01)^*}$

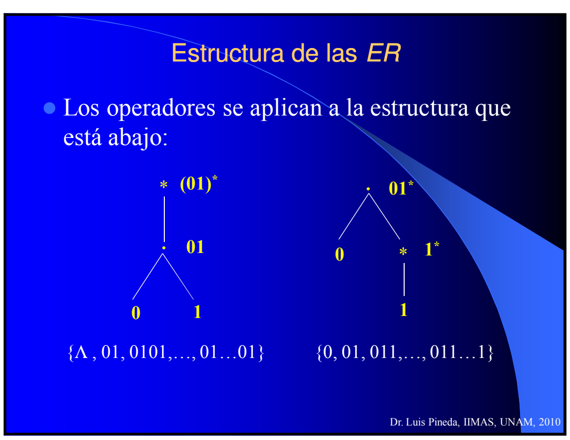


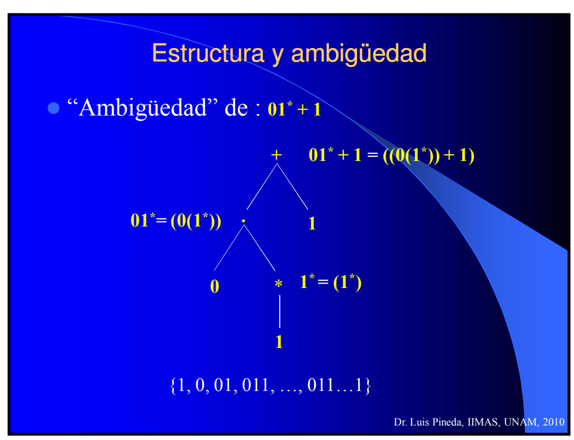

Ejemplo: El lenguaje de 0's y 1's alternados
Pero también necesitamos:
{0101…0}
{1010…0}
{1010…1}
* (01)*
01
0 1
{01}*
{01}
{0} {1}
Sintaxis
Semántica
Dr. Luis Pineda, IIMAS, UNAM, 2010

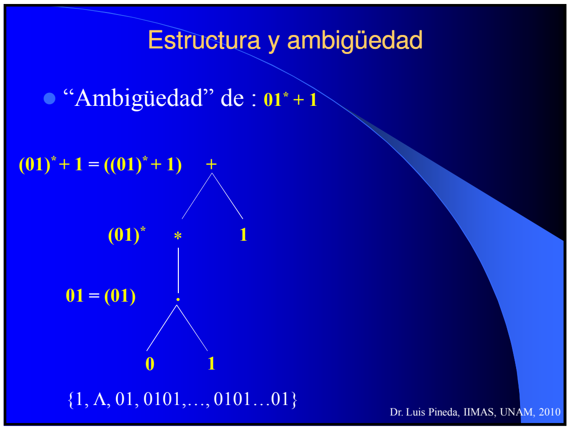

Ejemplo: El lenguaje de 0's y 1's alternados
+ (01)* + (10)* + 0(10)* + 1(01)*
+ 0(10)* + 1(01)*
+ (01)* + (10)*
0(10)*
1(01)*
* (01)*
* (10)*
0
* (10)*
1
* (01)*
01
10
10
01
0 1
1 0
1 0
0 1
0101…1
1010…0
0101…0
1010…1
Dr. Luis Pineda, IIMAS, UNAM, 2010

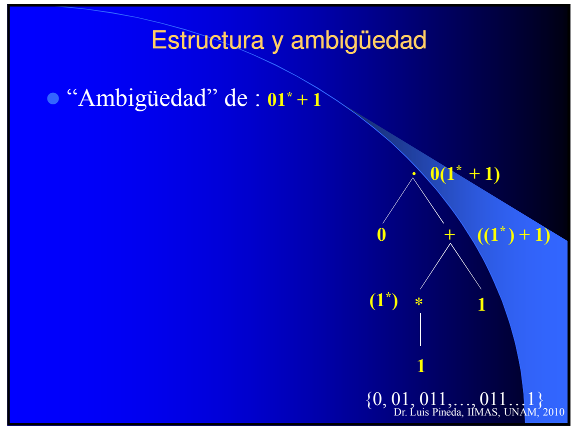

El poder expresivo de Λ
(Λ + 1) (01)*(0 + Λ)
(Λ + 1) (01)*
* (01)*
+ (Λ + 1)
01
+ (0 + Λ)
Λ 1
0 1
Λ 0
L(Λ) = {Λ}
L(Λ) = {Λ}
L(0) = {0}
Dr. Luis Pineda, IIMAS, UNAM, 2010

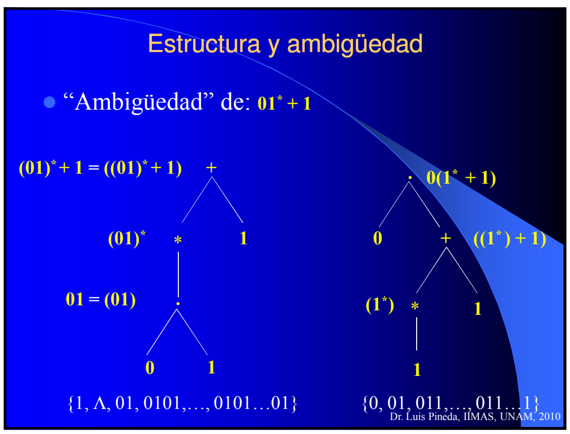

Las ER ¿son ambiguas?
Λ + 101*0 + Λ ?
(Λ + 1) (01)*
* (01)*
+ (Λ + 1)
01
+ (0 + Λ)
Λ 1
0 1
Λ 0
L(Λ)
L(1)
L(Λ)
L(0)
Dr. Luis Pineda, IIMAS, UNAM, 2010

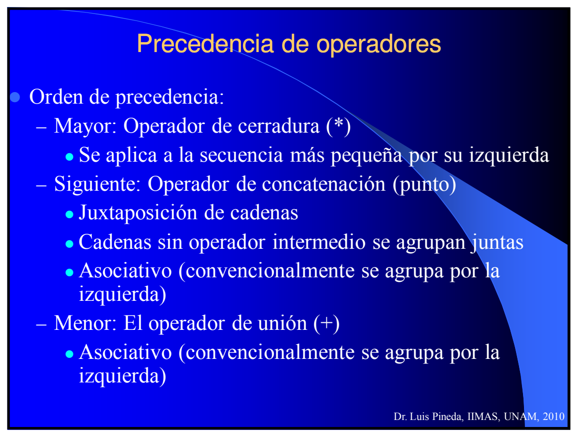

Estructura de las ER
Los operadores se aplican a la estructura que está abajo:
* (01)*
01
0 1
01*
0
* 1*
1
{Λ , 01, 0101,…, 01…01}
{0, 01, 011,…, 011…1}
Dr. Luis Pineda, IIMAS, UNAM, 2010

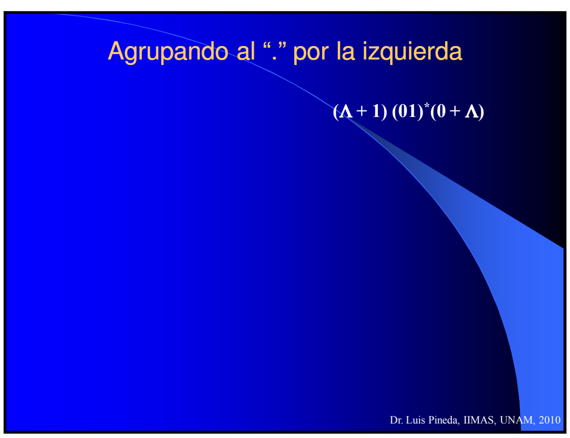

Estructura y ambigüedad
"Ambigüedad" de : 01* + 1
+ 01* + 1 = ((0(1*)) + 1)
01* = (0(1*))
1
0
* 1* = (1*)
1
{1, 0, 01, 011, …, 011…1}
Dr. Luis Pineda, IIMAS, UNAM, 2010

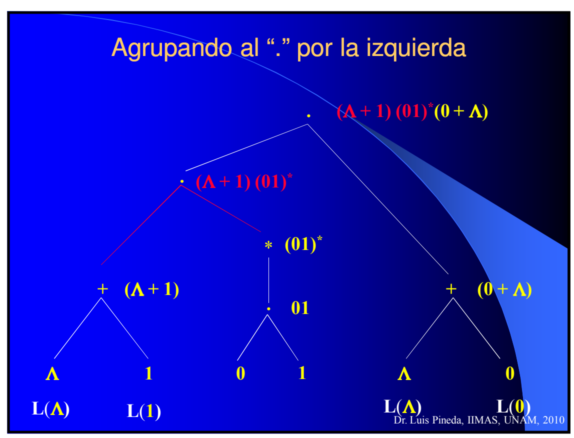

Estructura y ambigüedad
“Ambigüedad” de : $01^* + 1$
$(01)^* + 1 = ((01)^* + 1)$
+
$(01)^*$
*
1
01 = (01)
.
0
1
{1, Λ, 01, 0101,…, 0101…01}
Dr. Luis Pineda, IIMAS, UNAM, 2010


Estructura y ambigüedad
“Ambigüedad” de : $01^* + 1$
.
$0(1^* + 1)$
0
+
$((1^*) + 1)$
(1^*)
*
1
1
{0, 01, 011,…, 011…1}
Dr. Luis Pineda, IIMAS, UNAM, 2010


Estructura y ambigüedad
“Ambigüedad” de: $01^* + 1$
$(01)^* + 1 = ((01)^* + 1)$
+
$(01)^*$
*
1
01 = (01)
.
0
1
{1, Λ, 01, 0101,…, 0101…01}
.
$0(1^* + 1)$
0
+
$((1^*) + 1)$
(1^*)
*
1
1
{0, 01, 011,…, 011…1}
Dr. Luis Pineda, IIMAS, UNAM, 2010


Precedencia de operadores
Orden de precedencia:
– Mayor: Operador de cerradura (*)
Se aplica a la secuencia más pequeña por su izquierda
– Siguiente: Operador de concatenación (punto)
Juxtaposición de cadenas
Cadenas sin operador intermedio se agrupan juntas
Asociativo (convencionalmente se agrupa por la izquierda)
– Menor: El operador de unión (+)
Asociativo (convencionalmente se agrupa por la izquierda)
Dr. Luis Pineda, IIMAS, UNAM, 2010


Agrupando al “.” por la izquierda
$(Λ + 1)\ (01)^*(0 + Λ)$
Dr. Luis Pineda, IIMAS, UNAM, 2010

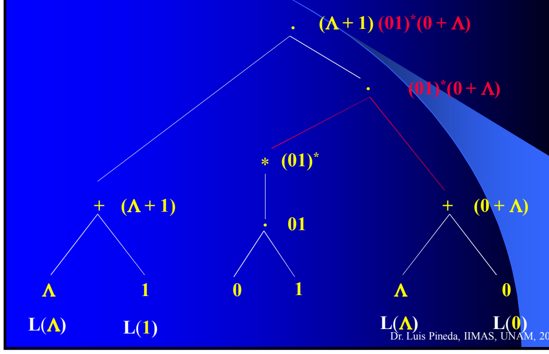

Agrupando al “.” por la izquierda
.
$(Λ + 1)\ (01)^*(0 + Λ)$
.
$(Λ + 1)\ (01)^*$
*
$(01)^*$
+
(Λ + 1)
.
01
+
(0 + Λ)
Λ
1
0
1
Λ
0
L(Λ)
L(1)
L(Λ)
L(0)
Dr. Luis Pineda, IIMAS, UNAM, 2010

Agrupando al "." por la derecha

$(\Lambda + 1)\ (01)^*(0 + \Lambda)$

Dr. Luis Pineda, IIMAS, UNAM, 2010

Agrupando al "." por la derecha



Dr. Luis Pineda, IIMAS, UNAM, 2010

Precedencia, paréntesis y ambigüedad

- Las *ER* son aparentemente ambiguas, pero
  - Los paréntesis y el orden de precedencia eliminan la ambigüedad
- También:
  - *ER* tienen estructura
  - Los árboles muestran la extructura explícitamente!
  - Las expresiones ambiguas tienen varias estructuras posibles
  - Sólo hay una estructura para cada *ER*
- Tomando en cuenta los paréntesis y la precedencia de operadores, no hay ambigüedad posible
- ¡*ER* no son ambiguas!

Dr. Luis Pineda, IIMAS, UNAM, 2010

Igualdad de expresiones regulares

- Util para la simplificación de expresiones
- Como veremos, útil para simplificar Autómatas (con el menor número de estados posibles)
  - $1^*(1 + \Lambda) = 1^*$
  - $1^*1^* = 1^*$
  - $0^* + 1^* = 1^* + 0^*$
  - $(0^*1^*)^* = (0 + 1)^*$
  - $(0 + 1)^*01(0 + 1)^* + 1^*0^* = (0 + 1)^*$
- Existe un método general (un algoritmo) para decidir si dos expresiones denotan al mismo lenguaje

Dr. Luis Pineda, IIMAS, UNAM, 2010

Simplificando *ER*

- $(r + s + rs + sr)^*$

Dr. Luis Pineda, IIMAS, UNAM, 2010

Simplificando *ER*

- $(r + s + rs + sr)^*$
  - *rs* puede formarse tomando *r* & *s*;
  - *sr* puede formarse tomando *s* & *r*,
  - entonces:

$$(r + s + rs + sr)^* = (r + s)^*$$

Dr. Luis Pineda, IIMAS, UNAM, 2010

Simplificando *ER*

- $r(r^*r + r^*) + r^*$

Dr. Luis Pineda, IIMAS, UNAM, 2010

Simplificando *ER*

- $r(r^*r + r^*) + r^*$
  - $r^*r = r^+$
  - $r(r^*r + r^*) + r^* = r(r^+ + r^*) + r^*$
    $= r(r^*) + r^*$
    $= rr^* + r^*$
    $= r^+ + r^*$
    $= r^*$

Dr. Luis Pineda, IIMAS, UNAM, 2010

Interpretando *RE*

- Considere las expresiones regulares:
  - $r = 0^* + 1^*$
  - $s = 01^* + 10^* + 1^*0 + (0^*1)^*$
- Una cadena en r pero no en s
  - 00
- Una cadena en s pero no en r
  - 01
- Una cadena en ambas r & s
  - Hay varias obvias: Λ, 0, 1
- Una cadena en $\{0, 1\}^*$ que no está en r ni en s
  - Cualquier cadena de forma: 1^i0^i para $i \geq 2$

Dr. Luis Pineda, IIMAS, UNAM, 2010

Encontrar una *RE*

- Dar una expresión regular para el lenguaje:
  - $\{ s \in \{a, b\}^* : |s|$ no es divisible entre $2\}$
- Si $|s|$ es divisible entre 2
  - Su longitud es par (i.e. en otro caso es non)
- *ER* para cadenas de longitud par:
  - $(aa + ab + ba + bb)^*$

Dr. Luis Pineda, IIMAS, UNAM, 2010

Encontrar una *RE*

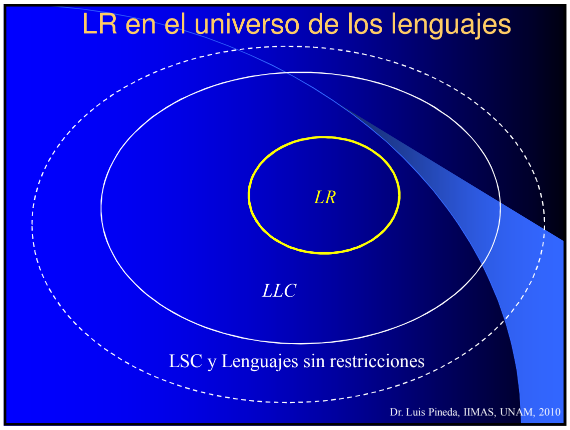
- Aumentando un símbolo, ya sea a o b (i.e. cadenas de longitud non):
  - $(aa + ab + ba + bb)^*(a + b)$
- Alternativamente:
  - $(a + b)(aa + ab + ba + bb)^*$
- Introduciendo abstracción:
  - $(a + b)((a + b)(a + \mathrm{b}))^*$

Dr. Luis Pineda, IIMAS, UNAM, 2010

Conjuntos regulares

- Lenguajes generados a partir de:
  - Φ, Λ y los símbolos en Σ
- Por medio de:
  - Unión
  - Concatenación
  - Cerradura de Kleen (Kleen-star)
- A través de un número *finito* de operaciones
  - Una expresión regular es una cadena finita!
  - Formalmente, no permitimos elípsis (…)
  - Una árbol es también una estructura finita!
- El conjunto denotado es un subconjunto del conjunto potencia de Σ^* (2^{Σ^*}:), el cual no puede contarse!
- Hay muchos, muchos, lenguajes que no son regulares!

Dr. Luis Pineda, IIMAS, UNAM, 2010


LR en el universo de los lenguajes
LR
LLC
LSC y Lenguajes sin restricciones
Dr. Luis Pineda, IIMAS, UNAM, 2010