Ciencias de la Computación I

Gramáticas Regulares

Gramáticas

- Una gramática es un conjunto de reglas para formar correctamente las frases de un lenguaje.

- Por ejemplo, la gramática del castellano nos permite:

  - Identificar cuándo una frase es sintácticamente correcta

    "JUAN CORRE RAPIDO"
    "RAPIDO JUAN CAMINA"

  - Generar todas las posibles frases sintácticamente correctas

En esta materia estudiaremos:

Gramáticas Formales $\Rightarrow$ GENERADORAS de Lenguajes Formales

Gramáticas

Oración del lenguaje castellano /*ej. simplificado*/

<oración> se define como <sujeto> <predicado>

<sujeto> se define como <nombre>

<predicado> se define como <verbo> <complemento>
<predicado> se define como <verbo>

<nombre> se define como **JUAN** ó **PEDRO**

<verbo> se define como **CORRE** ó **CAMINA**

<complemento> se define como **RAPIDO** ó **LENTO**

Ejemplos

JUAN CORRE RAPIDO
PEDRO CAMINA
oraciones bien definidas según gramática anterior

RAPIDO JUAN CAMINA
LENTO CAMINA
oraciones mal definidas según gramática anterior

Gramáticas

Sentencia **asignación** de Pascal /*ej. simplificado*/

<asignación> se define como <variable> **:=** <expresión>

<expresión> se define como <número>
<expresión> se define como <variable>

<variable> se define como **A** o **B** o **C**
<número> se define como **0** ó **1** ó … ó **9**

Ejemplos

A := B
A := 3
asignaciones bien definidas según gramática anterior

SUMA:= A
3 := A
asignaciones mal definidas según gramática anterior

Gramáticas Formales

Una gramática formal es una cuadrupla G = <N, T, P, S>

N = conjunto finito de símbolos no terminales

T = conjunto finito de símbolos terminales

$\left.\right\}$ **$N \cap T = \emptyset$**

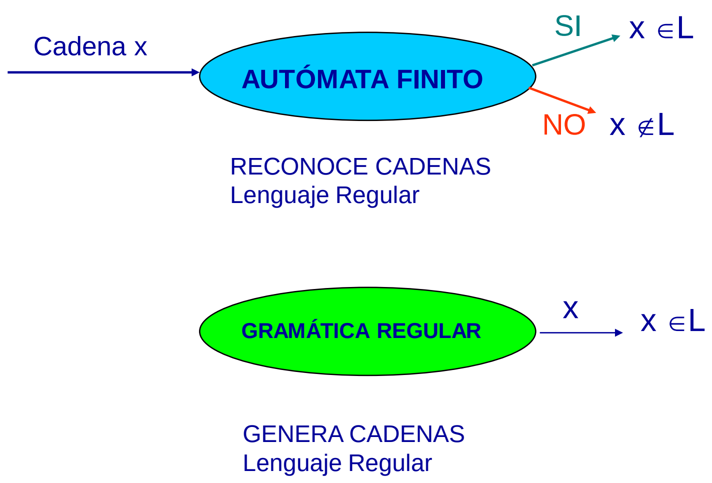
S = símbolo distinguido o axioma **$S \notin (N \cup T)$**

P = conjunto finito de reglas de producción **(permiten generar cadenas a partir de S)**

$$\left.\begin{array}{l} \alpha \rightarrow \beta \\ \alpha = \varphi A \rho \\ \beta = \varphi \omega \rho \end{array}\right\} \quad \begin{array}{l} A \in N \cup \{S\} \\ \varphi, \omega, \rho \in (N \cup T)^* \end{array}$$

De acuerdo al formato de las reglas se pueden definir 4 tipos de gramáticas y sus correspondientes lenguajes

Lenguajes Regulares



Gramáticas Regulares

Reglas de producción

1) $S \rightarrow b$
2) $S \rightarrow aA$
3) $A \rightarrow aA$
4) $A \rightarrow b$

- S es el símbolo distinguido a partir del cual se comienza a generar
- A es un símbolo No Terminal
- a, b son símbolos del Alfabeto
- $\rightarrow$ ¨se reemplaza con¨

Derivaciones

$S \Rightarrow b$ — $b \in L$

$S \Rightarrow aA \Rightarrow ab$ — $ab \in L$

$S \Rightarrow aA \Rightarrow aaA \Rightarrow aab$ — $aab \in L$

$S \Rightarrow aA \Rightarrow aaA \Rightarrow aaaA \Rightarrow aaab$ — $aaab \in L$

Se pueden generar infinitas cadenas …..

$L = \{a^n\, b \,/\, n \geq 0\}$

Gramáticas Regulares (Tipo 3)

- **Generan los lenguajes regulares (reconocidos por Autómatas Finitos)**
- **Se definen como una cuadrupla G = <N, T, P, S>**

N = conjunto finito de símbolos no terminales

T = conjunto finito de símbolos terminales

S = símbolo distinguido o axioma $S \notin (N \cup T)$

P = conjunto finito de reglas de producción

Formato reglas de producción de Gramáticas Regulares

Lineal a derecha	**Lineal a izquierda**
$A \rightarrow aB$	$A \rightarrow Ba$
$A \rightarrow a$	$A \rightarrow a$
$S \rightarrow \varepsilon$ (para generar la cadena vacía)	$S \rightarrow \varepsilon$ (para generar la cadena vacía)
$A \in N \cup \{S\}$ $a \in T$ $B \in N$	$A \in N \cup \{S\}$ $a \in T$ $B \in N$

Gramática Regular (Tipo 3)

Ejemplo 1:

Sea G_1 = <{A}, {a, b}, P_1 , S >   donde $P_1 = \{ S \rightarrow b$

$S \rightarrow aA,$

$A \rightarrow aA,$

$A \rightarrow b \}$

G_1 es una gramática Lineal a Izquierda o Lineal a Derecha?

G_1 es una gramática regular lineal a derecha que genera el lenguaje:

$L = \{a^n\, b \;/\; n \geq 0 \}$

Gramáticas Regulares

Ejemplo 2   Sea G_2 = <{A}, {a, b}, P_2 , S >   donde P_2: 1) $S \rightarrow b,$

2) $S \rightarrow Ab,$

3) $A \rightarrow Aa,$

4) $A \rightarrow a$

Derivaciones

$S \Rightarrow b$ $b \in L$

$S \Rightarrow Ab \Rightarrow ab$ $ab \in L$

$S \Rightarrow Ab \Rightarrow Aab \Rightarrow aab$ $aab \in L$

$S \Rightarrow Ab \Rightarrow Aab \Rightarrow Aaab \Rightarrow aaab$ $aaab \in L$

$L = \{a^n\, b \;/\; n \geq 0\}$

Se pueden generar infinitas cadenas   …..

G_2 es una gramática regular lineal a izquierda que genera:

Gramáticas Regulares (Tipo 3)

Derivación inmediata (lineal a derecha):

$\omega \Rightarrow \beta$ La cadena β se obtiene de la cadena ω en un paso usando las reglas de P. Si $\omega = \alpha A$ y $\beta = \alpha\delta$ entonces:

$\alpha A \Rightarrow \alpha\delta$ sii existe en P la regla $A \to \delta$ y $\alpha \in T^*$ $A \in N \cup \{S\}$

siendo $\delta = aB$ o $\delta = a$ $a \in T$ $B \in N$

Cuando $A = S$ puede ser $\delta = \varepsilon$

Ejemplo Si G = <{A}, {a, b}, P, S> donde $P = \{S \to b, S \to aA, A \to aA, A \to b\}$

$S \Rightarrow aA \Rightarrow \underbrace{a}_{\alpha}\underbrace{aA}_{A} \Rightarrow \underbrace{aa}_{\alpha}\underbrace{aA}_{\delta} \Rightarrow aaab$

En un paso se aplicó la regla $A \to aA$

Gramáticas Regulares (Tipo 3)

Derivación: La cadena β se obtiene de la cadena ω en cero o más pasos usando las reglas de P. Se define la clausura reflexiva y transitiva de $\Rightarrow$

$\alpha_1 \Rightarrow \alpha_2 \Rightarrow \ldots \Rightarrow \alpha_n$ decimos que $\alpha_1 \overset{*}{\Rightarrow} \alpha_n$ para $\alpha_i \in (N \cup T)^*$

Ejemplo

$S \Rightarrow aA \Rightarrow aaA \Rightarrow aaaA \Rightarrow aaab$ → $S \overset{*}{\Rightarrow} aaab$

En varios pasos

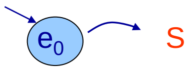
Lenguaje generado por una gramática regular G = <N, T, P, S>:

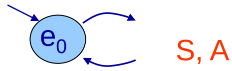
$$L(G) = \{ x \,/\, x \in T^* \text{ y } S \overset{*}{\Rightarrow} x \}$$

Es decir, una cadena $\in$ L(G) si:

1) La cadena está formada por símbolos terminales únicamente
2) La cadena puede ser derivada a partir de S

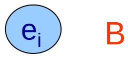
Pasaje de Autómata Finito a Gramática Regular

1) 

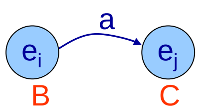


Nombrar S si tiene sólo arcos salientes

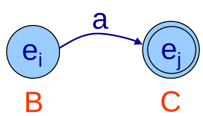
Nombrar S y un no terminal A si tiene arcos salientes y entrantes

2) 

al resto de los estados asociar un no terminal

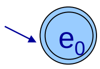
3) 

agregar la regla $B \rightarrow aC$

4) 

agregar las reglas $B \rightarrow aC$

$B \rightarrow a$

5) 

agregar la regla $S \rightarrow \varepsilon$

Ciencias de la Computación I

Expresiones Regulares

Expresiones Regulares

Se denominan Expresiones Regulares (ER) sobre un alfabeto A, a las expresiones que se pueden construir a partir de las siguientes reglas:

- $\varnothing$ es ER que describe el lenguaje vacío

- ε es ER que describe el lenguaje $\{\varepsilon\}$ (el lenguaje que contiene sólo la cadena vacía)

-Para cada símbolo $a \in A$, a es ER que describe el lenguaje $\{a\}$

- Si r y s son ER que describen los lenguajes $L(r)$ y $L(s)$ respectivamente:

  $r + s$ es ER que describe el lenguaje $L(r) \cup L(s)$

  $r . s$ es ER que describe el lenguaje $L(r) . L(s)$

  r^* es ER que describe el lenguaje $L(r)^*$

• Precedencia de operadores (de mayor a menor): $*$, $.$, $+$

• Se pueden usar paréntesis

Expresiones Regulares

Las ER describen a los lenguajes regulares (aquellos reconocidos por autómatas finitos y generados por gramáticas regulares).

Ejemplos:

Dado el alfabeto A = {a, b}

ER	Lenguaje que describe
$r = a + b$	$L(r) = \{a, b\}$
$r = ab$	$L(r) = \{ab\}$
$r = a^*b$	$L(r) = \{a^n b \mid n \geq 0\}$
$r = (a + b)^*b$	$L(r) = \{ x / x \in \{a, b\}^*$ y x termina en b$\}$
$r = (a + b)^*ab\,(a + b)^*$	$L(r) = \{ x / x \in \{a, b\}^*$ y x contiene ab$\}$
$r = a(aa)^*$	$L(r) = \{a^{2k+1} / k \geq 0\}$

Expresiones Regulares

Ejemplos:

¿Qué lenguaje describen las Expresiones Regulares ?

1) digito = 1+2+ 3+ 4+ 5+6+7+ 8+ 9
2) dig = 0+1+2+3+4+5+6+7+8+9
3) letra = a+b+ …+z
4) r_4 = digito . digito
5) r_5 = digito . dig^*
6) r_6 = letra. $(letra + dig)^*$
7) $r_7 = ab^* + a$
8) $r_8 = ab^*$

¿Qué puede decirse de r_7 y r_8?

Expresiones Regulares

Expresiones regulares equivalentes:

Dos ER r_1 y r_2 son equivalentes $r_1 \equiv r_2$ si $L(r_1) = L(r_2)$

(es decir, r_1 y r_2 describen el mismo conjunto de cadenas)

Leyes algebraicas para expresiones regulares

Sean r, s y t expresiones regulares:

1) $r + \varnothing \equiv \varnothing + r \equiv r$
2) $r \,.\, \varepsilon \equiv \varepsilon \,.\, r \equiv r$
3) $r \,.\, \varnothing \equiv \varnothing \,.\, r \equiv \varnothing$
4) $r + s \equiv s + r$
5) $(r + s) + t \equiv r + (s + t)$
6) $(r \,.\, s) \,.\, t \equiv r \,.\, (s \,.\, t)$
7) $r \,.\, (s + t) \equiv r \,.\, s + r \,.\, t$
8) $(s + t) \,.\, r \equiv s \,.\, r + t \,.\, r$
9) $r + r \equiv r$
10) $\varnothing^* \equiv \varepsilon$
11) $r \,.\, r^* \equiv r^* \,.\, r$
12) $r \,.\, r^* + \varepsilon \equiv r^*$
13) $(r^* \,.\, s^*)^* \equiv (r + s)^*$

Aplicación de Leyes Algebraicas para ER

Ejemplo

$r = a \,.\, b^* + a$ **Aplica ley**

$a \,.\, b^* + a \equiv a.\,(b^* + \varepsilon)$	**(7)**
$\equiv a \,.\, (\, \varepsilon + b.b^* + \varepsilon\,)$	**(12)**
$\equiv a \,.\, (\, \varepsilon + \varepsilon + b.b^*\,)$	**(4)**
$\equiv a \,.\, (\, \varepsilon + b.b^*\,)$	**(9)**
$\equiv a \,.\, b^*$	**(12)**

$a \,.\, b^* + a \equiv a \,.\, b^*$ son ER equivalentes

Lenguajes Regulares