Ciencias de la Computación I

Gramáticas Regulares

Gramáticas

- Intuitivamente una gramática es un conjunto de reglas para formar correctamente las frases de un lenguaje.
- Por ejemplo, la gramática del castellano (o de cualquier idioma) 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 Formales

Una gramática formal se define como una cuadrupla G = <N, T, P,S >

N = conjunto finito de símbolos no terminales

T = conjunto finito de símbolos terminales $\quad N \cap T = \varnothing$

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

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

$\alpha \rightarrow \beta$

donde:

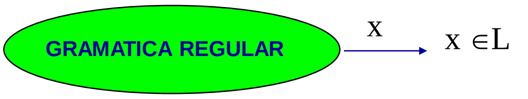
$\alpha = \varphi A \rho \qquad A \in N \cup \{S\}$

$\beta = \varphi w \rho \qquad \varphi, \rho, w \in (N \cup T)^*$

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

Lenguajes Regulares

RECONOCE CADENAS
Lenguaje Regular



GENERA CADENAS
Lenguaje Regular

Gramática Regular

Ejemplo de Reglas de producción

1) S $\longrightarrow$ aA
2) A $\longrightarrow$ aA
3) A $\longrightarrow$ b

- •S es el símbolo distinguido, comienza a generar
- •A es un símbolo no terminal
- • a,b son símbolos terminales

- • S se reemplaza por aA ó
- • A se reemplaza por aA ó
- • A se reemplaza por b

Derivaciones

$S \Rightarrow aA \Rightarrow ab$ 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 …..

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

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)

Ejemplos:

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

$A \to aA$,

$A \to b$ }

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:

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

Gramáticas Regulares

Ejemplo 2

Sea G_2 = <{A}, {a, b}, P_2 , S > donde P_2 = { $S \to Ab$,

$A \to Aa$,

$A \to a$ }

Derivaciones

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

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

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

Se pueden generar infinitas cadenas …..

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

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 $A \rightarrow \delta$ 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 $G = <\{A\}, \{a, b\}, P_1, S>$ donde $P_1 = \{ S \rightarrow aA, A \rightarrow aA, A \rightarrow b \}$

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

En un paso se aplicó la regla $A \rightarrow aA$ de P_1

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 $\Rightarrow$

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

Ejemplo

$\boxed{S \Rightarrow aA \Rightarrow aaA \Rightarrow aaaA \Rightarrow aaab} \xrightarrow{\text{En varios pasos}} S \overset{*}{\Rightarrow} aaab$

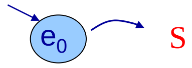
Lenguaje generado por una gramática regular G = <N, T, P, S>:

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

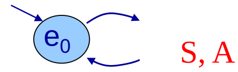
Es decir, una cadena $\in L(G)$ si:

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

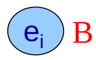
Pasaje de Autómata Finito a Gramática Regular

1) 

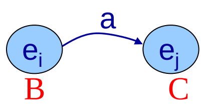
Nombrar S si tiene solo arcos salientes



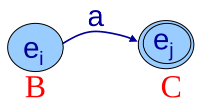
Nombrar S y un no terminal A si tiene arcos salientes, y entrantes

2) 

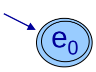
Para el resto de los estados asociar un no terminal

3) 

Agregar la regla $B \rightarrow aC$

4) 

Agregar la 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
$r = a + b$	$L(r) = \{a, b\}$
$r = ab$	$L(r) = \{ab\}$
$r = a^*b$	$L(r) = \{a^n b \,/\, n \geq 0\}$
$r = (a + b)^*b$	$L(r) = \{ x \,/\, x \in \{a, b\}^*$ y x termina en b$\}$
$r = (a + b)^*ab\,(a + b)^*$	$L(r) = \{ x \,/\, x \in \{a, b\}^*$ y x contiene ab$\}$

Expresiones Regulares

Ejemplos:

1) ¿Qué lenguaje describe ER r_1 = digito . dig* ?

donde digito = {1, 2, 3, 4, 5, 6, 7, 8, 9} y dig = {0, 1, 2, 3, 4, 5, 6, 7, 8, 9}

2) ¿Qué lenguaje describe ER r_2 = letra. (letra + dig)* ?

donde letra = {a, b, …, z} y dig = {0, 1, 2, 3, 4, 5, 6, 7, 8, 9}

3) ¿Qué lenguaje describe ER $r_3 = ab^* + a$?

4) ¿Qué lenguaje describe ER $r_4 = ab^*$?

5) ¿Qué puede decirse de r_3 y r_4?

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)**

$\boxed{a \,.\, b^* + a \equiv a \,.\, b^* \quad \text{son ER equivalentes}}$

Lenguajes Regulares

Lenguajes regulares

- Se reconocen con → Autómatas Finitos
- Se generan con → Gramáticas Regulares
- Se describen con → Expresiones Regulares