Gramáticas

- Recordemos que un lenguaje es un conjunto de cadenas o palabras sobre un determinado alfabeto
  $L \subseteq \Sigma^*$
- Los conjuntos los podemos definir por
  - Extensión: pero si el conjunto es infinito no nos sirve
  - Compresión
    - Frase explicativa: impreciso y complejo
    - Fórmula sobre las operaciones sobre cadenas: insuficiente para todos los lenguajes
    - Constructivo: Da un método para generar todos los elementos del conjunto y **solamente** los del conjunto
- Una gramática es un mecanismo constructivo que permite generar un lenguaje formal, basado en producciones

Definición Formal

Una gramática es una 4-úpla

$G = (V_N, V_T, P, S)$

Donde

V_N es el conjunto de NO terminales o variables

V_T, es el conjunto de terminales o alfabeto del lenguaje

$V_N \cap V_T = \emptyset$

$V = V_N \cup V_T$ (alfabeto total)

$S \in V_N$ (Axioma o símbolo inicial)

P es el conjunto de producciones

Producciones

P es un conjunto de pares ordenados que denotamos como

$\alpha \rightarrow \beta$ (leemos: α produce β)

Donde

$\alpha \in V^+ \wedge \beta \in V^*$

$\alpha = \varphi A \rho \wedge \varphi, \rho \in V^* \wedge A \in V_N$

Es decir, α debe contener al menos un no terminal

<u>Nota</u>: la aplicación sucesiva de producciones se conoce como derivación, algunos usan la notación $\alpha \rightarrow^* \gamma$ (leemos: α deriva en γ)

Ejemplos

Sea L = {“aa”, “ab”} Puede ser generado por:
G = ({S,T}, {a, b}, P, S)
P = {S → aT, T → a, T → b}

Dada la gramática
G = ({R,S,T,U}, {a, b, c, d, e}, P, S)
P = {S → aT | bRd,
T → c | Ub,
U → e,
R → a | dU}
¿Qué lenguaje genera?

Jerarquía de Chomsky

- Es de acuerdo a las restricciones que se agregan sobre las producciones
- Tipo **0** : irrestrictas
- Tipo **1**: sensibles al contexto
- Tipo **2**: independientes del contexto
- Tipo **3**: Regulares
- Un lenguaje generado por una gramática de un tipo se dice que es de ese tipo, por ejemplo lenguaje regular si es generado por una gramática tipo 3

Gramáticas Regulares

- Sus producciones deben cumplir
  - A izquierda hay un solo símbolo no terminal
  - A derecha hay un símbolo terminal, posiblemente acompañado de un no terminal, o ε
- Expresado de otro modo sus producciones toman la forma
  - $A \rightarrow a$
  - $A \rightarrow Ba$ ó $A \rightarrow aB$
  - $A \rightarrow \varepsilon$

  Donde $A,B \in V_N \wedge a \in V_T$

Lineales a izquierda o derecha

- Se dice que una gramática regular es **lineal a derecha** si todas las producciones con terminales y no terminales son de la forma $\mathbf{A} \rightarrow a\mathbf{B}$
- Se dice que es **lineal a izquierda** si dichas producciones son de la forma $\mathbf{A} \rightarrow \mathbf{B}a$
- Para que la gramática sea regular debe ser una o la otra, la mezcla hace perder la condición de regular
- Algunos autores llaman a esto “gramáticas **estrictamente** regulares a izquierda (o derecha)”

Definiciones Alternativas

- Hay quienes no admiten ε a derecha
  - Para incluir la cadena vacía permiten la producción $S \rightarrow \varepsilon$ pero en esos casos S no puede figurar a derecha en ninguna producción
- Otros[1] plantean que basta cumplir con
  $A \rightarrow wB$ (o $A \rightarrow Bw$)
  $A \rightarrow w$
  donde $A \wedge B \in V_N \wedge w \in V_T{}^*$
- Los autores que a la anterior notación las llaman "estrictas" a estas simplemente las llaman regulares. Si a las anteriores las llaman regulares, entonces a estas las llaman "**extendidas**"

1: John Hopcroft – jeffrey Ullman (1979) "Introduction to Automata Theory, Languages and Computation" Addison-Wesley

Producciones recursivas

- Es cuando un no terminal aparece a izquierda y derecha de la producción, por ejemplo

  $\mathbf{T} \rightarrow a\mathbf{T}$

- Esto permite generar lenguajes infinitos
- Ejemplos

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

$S \rightarrow \varepsilon \mid aS$

$L = \{a^{2n} / n \geq 0 \}$

$S \rightarrow \varepsilon \mid aT$

$T \rightarrow a \mid aS$

Ejemplo Gramática Regular

- $L = \{a^n b c^t \ / \ n \geq 0 \wedge t \geq 0\}$
  - $S \rightarrow aS \ | \ bC$
  - $C \rightarrow cC \ | \ \varepsilon$

- Mismo lenguaje pero con lineal a izquierda
  - $S \rightarrow Sc \ | \ Ab$
  - $A \rightarrow Aa \ | \ \varepsilon$

Gramáticas Quasi Regulares

- Son equivalentes a las regulares
- Son útiles porque reducen la cantidad de producciones
- Consiste en reemplazar un conjunto de terminales en varias producciones con un no terminal y agregar una producción donde el nuevo no terminal produce el conjunto que reemplaza
- Ejemplo: las producciones
  $S \rightarrow N \mid NS$
  $N \rightarrow a \mid b \mid c \mid d$
  Reemplazan a
  $S \rightarrow a \mid b \mid c \mid d \mid aS \mid bS \mid cS \mid dS$

Gramáticas independientes del contexto

- El lado izquierdo debe seguir siendo un único símbolo no terminal.
- No hay restricciones sobre el lado derecho
- Ejemplos

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

$S \rightarrow aSb \mid \varepsilon$

$L = \{a^{2n} b^{n+1} a^r \ / \ n \geq 1 \wedge r \geq 0\}$

$S \rightarrow aaTbQ$
$T \rightarrow aaTb \mid b$
$Q \rightarrow aQ \mid \varepsilon$

Proceso de derivación

- Representaciones
  - Horizontal utilizando el símbolo ⇒
  - Vertical con una línea por cada producción aplicada
  - Árbol con el axioma como raíz y en cada producción aplicada el no terminal reemplazado es padre de los símbolos que lo reemplazan
- Verticales
  - A izquierda en cada paso se reemplaza el no terminal más a la izquierda
  - A derecha: igual pero a derecha

Otras Gramáticas

- Irrestrictas
  - Basta con que sean gramáticas
- Sensibles al contexto
  - Elimina la restricción de un único símbolo a izquierda, pero el largo de la cadena producida deber ser mayor o igual al de la cadena a iquierda de la producción
  - Entonces si $\alpha \rightarrow \beta \Rightarrow |\alpha| \leq |\beta|$

Ejemplo gramática sensible al contexto

Podemos generar $L = \{a^n b^n c^n \ / \ n > 0\}$

$G = (\{S,A,B,C\} , \{a,b,c\}, P, S)$

Producciones de P

Nro Regla	Producción	Comentario
1	$S \rightarrow A$	Inicio
2	$A \rightarrow aABC$	Genero tantas a como quiera con igual cantidad de B y C
3	$A \rightarrow abC$	Última a con correspondiente b y C
4	$CB \rightarrow BC$	Ordeno haciendo que B viaje a izquierda
5	$bB \rightarrow bb$	Convierto B en b solo en el contexto adecuado
6	$bC \rightarrow bc$	Convierto C en c solo en el contexto adecuado
7	$cC \rightarrow cc$	Convierto C en c solo en el contexto adecuado

Ejemplo para producir aaabbbccc

Rojo: carácter sobre el que aplico la producción

Sobrelineado: Parte derecha de la producción aplicada

Aplicando

Obtengo

Cadena	Por aplicar Regla
S	Axioma
$\overline{\mathbf{A}}$	1 (S → A)
$\overline{a\mathbf{A}BC}$	2 (A → aABC)
$a\overline{a\mathbf{A}BC}BC$	2 (A → aABC)
$aa\overline{ab\mathbf{C}}\mathbf{B}CBC$	3 (A → abC)
$aaab\overline{BC}\mathbf{CB}C$	4 (CB → BC)
$aaabB\mathbf{C}\overline{\mathbf{B}C}C$	4 (CB → BC)
$aaa\mathbf{bB}\overline{BC}CC$	4 (CB → BC)
$aaa\overline{b\mathbf{bB}}CCC$	5 (bB → bb)
$aaab\overline{b\mathbf{bC}}CC$	5 (bB → bb)
$aaabb\overline{b\mathbf{cC}}C$	6 (bC → bc)
$aaabbb\overline{c\mathbf{cC}}$	7 (cC → cc)
$aaabbbc\overline{cc}$	7 (cC → cc)

Ejemplo gramática irrestricta

Podemos generar $L = \{a^i \ / \ i \text{ es } 2^n \text{ con } n > 0\}$
$G = (\{S,A,B,C,D,E\}, \{a\}, P, S)$
Producciones de P

Nro Regla	Producción	Comentario
1	$S \rightarrow ACaB$	A y B son límites para los no terminales “viajeros” C, D y E
2	$Ca \rightarrow aaC$	C duplica a viajando a derecha
3	$CB \rightarrow DB$	Al encontrar el límite muta en D si quiere continuar
4	$CB \rightarrow E$	O muta en E si quiere finalizar (consumiendo el límite)
5	$aD \rightarrow Da$	D simplemente viaja a izquierda sin alterar nada
6	$AD \rightarrow AC$	Al encontrar el límite muta a C para volver a duplicar a
7	$aE \rightarrow Ea$	E es similar a D, viaja sin alterar
8	$AE \rightarrow \varepsilon$	Pero al encontrar el límite lo consume para finalizar

Ejemplo para producir aaaa

Cadena	Por aplicar Regla
S	Axioma
ACaB	1 (S → ACaB)
AaaCB	2 (Ca → aaC)
AaaDB	3 (CB → DB)
AaDaB	5 (aD → Da)
ADaaB	5 (aD → Da)
ACaaB	6 (AD → AC)
AaaCaB	2 (Ca → aaC)

Cadena	Por aplicar Regla
AaaaaCB	2 (Ca → aaC)
AaaaaE	4 (CB → E)
AaaaEa	7 (aE → Ea)
AaaEaa	7 (aE → Ea)
AaEaaa	7 (aE → Ea)
AEaaaa	7 (aE → Ea)
aaaa	8 (AE → ε)

Licencia