Gramáticas

Son dispositivos generadores de cadenas. Se componen de una serie de algoritmos conclusivos, pero no son algoritmos.

Definiciones

- **Producir**: Es una relación binaria con cierre transitivo (producir en al menos 1 paso) y cierre transitivo y reflexivo (producir en n pasos)
- **Derivación**: Secuencia finita de producciones desde S hasta $w \in V^*$
- **Forma sentencial**: Cada una de las cadenas que se derivan a partir de S. Si $S \Rightarrow^* \alpha \in V^*$, entonces α es una forma sentencial de G
- **Cadena generada por una gramática**: w es una cadena generada por G sii $w \in T^*$ y w es una forma sentencial
- **Lenguaje generado por una gramática**: Todas las cadenas generadas por G constituyen el lenguaje $L(G)$

Tipos de reglas

Nombre	Símbolo	Regla	Restricciones
Estructura de frase	$Tipo0$	$\alpha \rightarrow \beta$	$(\alpha \in V^+) \wedge (\beta \in V^*)$
Sensible al contexto	$Tipo1$	$\alpha A\beta \rightarrow \alpha\gamma\beta$	$(\alpha, \beta \in V^*) \wedge (\gamma \in V^+) \wedge (A \in N)$
De contexto libre	$Tipo2$	$A \rightarrow \alpha$	$(A \in N) \wedge (\alpha \in V^+)$
Regular izquierda	R_{izq}	$A \rightarrow aB$	$(A, B \in N) \wedge (a \in T)$
Regular derecha	R_{der}	$A \rightarrow Ba$	$(A, B \in N) \wedge (a \in T)$
Regular terminal	R_{ter}	$A \rightarrow a$	$(A \in N) \wedge (a \in T)$
Tipo 3	R		$(p \in R_{izq}) \vee (p \in R_{der})$
Lineal	L	$A \rightarrow \alpha B\beta$	$((A, B \in N) \wedge (\alpha, \beta \in T^*)) \vee (p \in R_{izq}) \vee (p \in R_{der})$
Lineal terminal	L_{ter}	$A \rightarrow \alpha$	$(A \in N) \wedge (\alpha \in T^+)$
Lineal izquierda	L_{izq}	$A \rightarrow \alpha B$	$(A, B \in N) \wedge (\alpha \in T^*)$
Lineal derecha	L_{der}	$A \rightarrow B\alpha$	$(A, B \in N) \wedge (\alpha \in T^*)$
Unitaria	U	$A \rightarrow B$	$A, B \in N$
Épsilon	R_ε	$A \rightarrow \varepsilon$	$A \in N$

- Las gramáticas de $tipo0$ es la única que pueden sustituir más de un símbolo en otra cadena (antecedente > consecuente). Además, pueden convertir símbolos terminales en no terminales o en otros terminales.
- A partir de las reglas de contexto libre, toda regla de producción se compondrá de un antecedente **único** que precede a una cadena de V^+. Por otra parte, las reglas lineales son muy similares a las reglas regulares pero difieren en los conjuntos terminales que utilizan. Mientras que las reglas terminales utilizan símbolos de T, las reglas lineales utilizan símbolos de T^+ o T^*
- Las reglas unitarias y lineales son de tipo 2
- Las reglas Épsilon son de tipo 0

Tipos de gramáticas

Nombre	Símbolo	Condiciones
Estructura de frase	$Tipo0$	$\forall p \in P \rightarrow p \in Tipo0$
Sensible al contexto	$Tipo1$	$\forall p \in P \rightarrow p \in Tipo1$
De contexto libre	$Tipo2$	$\forall p \in P \rightarrow p \in Tipo2$
Regular	$Tipo3$	$P \subset GRI \vee P \subset GRD$
Regular izquierda	GRI	$\forall p \in P \rightarrow (p \in R_{ter}) \vee (p \in R_{izq})$
Regular derecha	GRD	$\forall p \in P \rightarrow (p \in R_{ter}) \vee (p \in R_{der})$
Lineal	GL	$\forall p \in P \rightarrow p \in Lineal$
Lineal izquierda	GLI	$\forall p \in P \rightarrow (p \in L_{izq}) \vee (p \in L_{ter})$
Lineal derecha	GLD	$\forall p \in P \rightarrow (p \in L_{der}) \vee (p \in L_{ter})$
Épsilon regular	$G\varepsilon R$	$P \subset G\varepsilon RI \vee P \subset G\varepsilon RD$
Épsilon regular izquierda	$G\varepsilon RI$	$\forall p \in P \rightarrow (p \in R_{ter}) \vee (p \in R_{izq}) \vee (p \in R_{\varepsilon})$
Épsilon regular derecha	$G\varepsilon RD$	$\forall p \in P \rightarrow (p \in R_{ter}) \vee (p \in R_{der}) \vee (p \in R_{\varepsilon})$
Épsilon - contexto libre	$G\varepsilon CL$	$\forall p \in P \rightarrow (p \in Tipo2) \vee (p \in R_{\epsilon})$

Jerarquía de Chomsky. Clasificación de lenguajes

$Finitos \subset Tipo3 \subset Lineales \subset Tipo2 \subset Tipo1 \subset Tipo0 \subset L_{REP} \subset 2^{\Sigma^*}$

¿Existe un algoritmo conclusivo para ...?

	Estructura de frase	Sensible al contexto	Contexto libre	Lineales	Regulares
$G_1 \equiv G_2$	No	No	No	No	**Sí**
$L(G_1) \subseteq L(G_2)$	No	No	No	No	**Sí**
$X \in L(G_1)$	No	**Sí**	**Sí**	**Sí**	**Sí**
$L(G_1) = \emptyset$	No	No	**Sí**	**Sí**	**Sí**
$\|\|L(G_1)\|\| \in N$	No	No	**Sí**	**Sí**	**Sí**
$L(G_1) \in L.3$	No	No	No	No	Trivial

Cierre de los tipos de lenguaje

	Estructura de frase	Sensible al contexto	Contexto libre	Lineales	Regulares
$L \cup L'$	Sí	Sí	Sí	**Sí**	Sí
$L \cap L'$	Sí	Sí	**No**	No	Sí
$L \cap L.3$	Sí	Sí	Sí	**Sí**	Sí
$\overline{L}$	**No**	Sí	**No**	No	Sí
LL'	Sí	Sí	Sí	No	Sí
L^N	Sí	Sí	Sí	No	Sí
L^*	Sí	Sí	Sí	No	Sí
L^+	Sí	Sí	Sí	No	Sí
L^R	Sí	Sí	Sí	**Sí**	Sí

Proposiciones

General

- La cadena w necesita al menos $|w|$ pasos para que un AFD lo compute
- La cadena w puede rechazarse en un autómata ND en menos de $|w|$ pasos si todas sus computaciones sobre w son de bloqueo
- $\varepsilon \in T^*$ por lo que la cadena vacía siempre es un símbolo terminal (pertenece a las cadenas terminales)

Reglas de producción

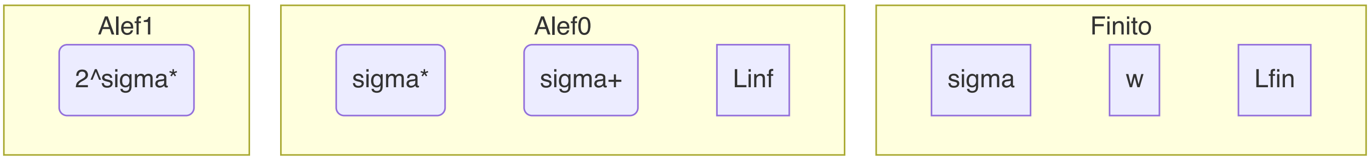
- Una regla de producción puede generar el mismo símbolo que su antecesor (Regla unitaria donde $A \rightarrow A$)
- Solo hacen falta 2 reglas para tener un lenguaje infinito.

Gramáticas

- Toda gramática es de tipo 0 (al menos)
- Existen infinitas gramáticas equivalentes
- Dada $G \in GRI$, se puede obtener $G \equiv G' \in GR\varepsilon I$ tal que $L(G') = L(G) \cup \{\varepsilon\}$
- Dada $G \in GCL$, se puede obtener $G \equiv G' \in GR\varepsilon CL$ tal que $L(G') = L(G) \cup \{\varepsilon\}$
- Una gramática puede ser lineal sin ser lineal derecha ni lineal izquierda
- Si G es Lineal izq y Lineal der, entonces $||L(G)|| \leq |P|$
- Si G es Lineal izq o Lineal der, entonces $L(G)$ es regular
- Si G es GRI y GRD, entonces solo contiene reglas terminales y por tanto $||L(G)|| \leq |T|$

Lenguajes

- Los lenguajes se analizan independientemente del símbolo Épsilon $L(G) = L - \{\varepsilon\}$
- Σ^* es un lenguaje regular
- No todo subconjunto de un lenguaje regular es también lenguaje regular
- Todo lenguaje es numerable puesto que es subconjunto de Σ^*
- Todo lenguaje no representable es la unión de infinitos lenguajes representables


Alef1
2^sigma*
Alef0
sigma*
sigma+
Linf
Finito
sigma
w
Lfin