

Gramáticas Regulares

Introducción a la Logica y la Computación (3era Parte)

Docentes: Badano, Bustos, Costamagna, Tellechea, Zigaran

Año 2024

Gramáticas de Chomsky

- ▶ Hasta aquí, hemos probado que los lenguajes regulares, las expresiones regulares y los autómatas finitos son equivalentes:

$$LR^{\Sigma} \equiv ER^{\Sigma} \equiv AF^{\Sigma}$$

Gramáticas de Chomsky

- ▶ Hasta aquí, hemos probado que los lenguajes regulares, las expresiones regulares y los autómatas finitos son equivalentes:

$$LR^{\Sigma} \equiv ER^{\Sigma} \equiv AF^{\Sigma}$$

- ▶ Vamos a presentar el último modelo computacional que caracteriza a los lenguajes regulares, las llamadas **gramáticas regulares**.

Gramáticas de Chomsky

- ▶ Hasta aquí, hemos probado que los lenguajes regulares, las expresiones regulares y los autómatas finitos son equivalentes:

$$LR^{\Sigma} \equiv ER^{\Sigma} \equiv AF^{\Sigma}$$

- ▶ Vamos a presentar el último modelo computacional que caracteriza a los lenguajes regulares, las llamadas **gramáticas regulares**.
- ▶ Las gramáticas fueron el modelo matemático propuesto originalmente por Chomsky para caracterizar cada una de las clases de lenguajes de su jerarquía:

Gramáticas de Chomsky

- ▶ Hasta aquí, hemos probado que los lenguajes regulares, las expresiones regulares y los autómatas finitos son equivalentes:

$$LR^{\Sigma} \equiv ER^{\Sigma} \equiv AF^{\Sigma}$$

- ▶ Vamos a presentar el último modelo computacional que caracteriza a los lenguajes regulares, las llamadas **gramáticas regulares**.
- ▶ Las gramáticas fueron el modelo matemático propuesto originalmente por Chomsky para caracterizar cada una de las clases de lenguajes de su jerarquía:

Lenguaje Regular	\leftrightarrow	Gramática Regular
Lenguaje Indep. Contexto	\leftrightarrow	Gramática Indep. Contexto
Lenguaje Sens. Contexto	\leftrightarrow	Gramática Sens. de Contexto
Lenguaje Rec. Enumerables	\leftrightarrow	Gramática Sin Restricciones

Gramática de Chomsky

- ▶ Definimos primero la versión más general del modelo, denominado gramática sin restricciones.

Gramática de Chomsky

- ▶ Definimos primero la versión más general del modelo, denominado gramática sin restricciones.
- ▶ Y luego, le impondremos una fuerte restricción dando lugar a la llamada gramática regular.

Gramática de Chomsky

- ▶ Definimos primero la versión más general del modelo, denominado gramática sin restricciones.
- ▶ Y luego, le impondremos una fuerte restricción dando lugar a la llamada gramática regular.

Definición (Gramática Sin Restricciones)

Una gramática es una 4-upla $G = (V, \Sigma, S, P)$, con $V \cap \Sigma = \emptyset$, donde:

- ▶ *V es un conjunto finito de **símbolos no terminales**, llamado conjunto de variables.*
- ▶ *Σ es un conjunto finito de **símbolos terminales**.*
- ▶ *$S \in V$, llamada **variable inicial**.*
- ▶ *P es un conjunto finito de **producciones** de la forma $\alpha \rightarrow \beta$ tal que $\alpha, \beta \in (V \cup \Sigma)^*$ y $\alpha \neq \epsilon$.*

Cómo Funciona una Gramática?

- ▶ Una gramática permite generar o derivar cadenas de símbolos terminales mediante una sucesión de reemplazos sintácticos definidos en su conjunto de producciones.

Cómo Funciona una Gramática?

- ▶ Una gramática permite generar o derivar cadenas de símbolos terminales mediante una sucesión de reemplazos sintácticos definidos en su conjunto de producciones.
- ▶ La producción $\alpha \rightarrow \beta$ significa que podemos **sustituir** la ocurrencia de la cadena α por la cadena β .

Cómo Funciona una Gramática?

- ▶ Una gramática permite generar o derivar cadenas de símbolos terminales mediante una sucesión de reemplazos sintácticos definidos en su conjunto de producciones.
- ▶ La producción $\alpha \rightarrow \beta$ significa que podemos **sustituir** la ocurrencia de la cadena α por la cadena β .
- ▶ Por ejemplo, sea $G_1 = (\{S\}, \{a, b\}, S, \{S \rightarrow aS, S \rightarrow \epsilon\})$ una gramática, podemos derivar la cadena “aaa” en G_1 de la siguiente manera:

Cómo Funciona una Gramática?

- ▶ Una gramática permite generar o derivar cadenas de símbolos terminales mediante una sucesión de reemplazos sintácticos definidos en su conjunto de producciones.
- ▶ La producción $\alpha \rightarrow \beta$ significa que podemos **sustituir** la ocurrencia de la cadena α por la cadena β .
- ▶ Por ejemplo, sea $G_1 = (\{S\}, \{a, b\}, S, \{S \rightarrow aS, S \rightarrow \epsilon\})$ una gramática, podemos derivar la cadena “aaa” en G_1 de la siguiente manera:

$$S \xRightarrow{G_1} aS \xRightarrow{G_1} aaS \xRightarrow{G_1} aaaS \xRightarrow{G_1} aaa$$

Cómo Funciona una Gramática?

- ▶ Una gramática permite generar o derivar cadenas de símbolos terminales mediante una sucesión de reemplazos sintácticos definidos en su conjunto de producciones.
- ▶ La producción $\alpha \rightarrow \beta$ significa que podemos **sustituir** la ocurrencia de la cadena α por la cadena β .
- ▶ Por ejemplo, sea $G_1 = (\{S\}, \{a, b\}, S, \{S \rightarrow aS, S \rightarrow \epsilon\})$ una gramática, podemos derivar la cadena “aaa” en G_1 de la siguiente manera:

$$S \xRightarrow{G_1} aS \xRightarrow{G_1} aaS \xRightarrow{G_1} aaaS \xRightarrow{G_1} aaa$$

- ▶ Todas la cadenas que se pueden derivar en una gramática definen el lenguaje de la gramática.

Cómo Funciona una Gramática?

- ▶ Una gramática permite generar o derivar cadenas de símbolos terminales mediante una sucesión de reemplazos sintácticos definidos en su conjunto de producciones.
- ▶ La producción $\alpha \rightarrow \beta$ significa que podemos **sustituir** la ocurrencia de la cadena α por la cadena β .
- ▶ Por ejemplo, sea $G_1 = (\{S\}, \{a, b\}, S, \{S \rightarrow aS, S \rightarrow \epsilon\})$ una gramática, podemos derivar la cadena “aaa” en G_1 de la siguiente manera:

$$S \xRightarrow{G_1} aS \xRightarrow{G_1} aaS \xRightarrow{G_1} aaaS \xRightarrow{G_1} aaa$$

- ▶ Todas la cadenas que se pueden derivar en una gramática definen el lenguaje de la gramática.
- ▶ Por legibilidad, describimos las gramáticas listando sus producciones separadas por un “|”:

$$G_1 : S \rightarrow aS | \epsilon$$

Derivación de una Cadena

Definimos formalmente la noción de 1-paso de derivación:

$$\alpha \xRightarrow{G} \beta \text{ sii } \exists \alpha_1, \alpha_2, \alpha'_2, \alpha_3 \in (V \cup \Sigma)^* \text{ tq } \alpha = \alpha_1 \alpha_2 \alpha_3$$

$$\beta = \alpha_1 \alpha'_2 \alpha_3$$

$$\alpha_2 \rightarrow \alpha'_2 \in P$$

Derivación de una Cadena

Definimos formalmente la noción de 1-paso de derivación:

$$\alpha \xRightarrow{G} \beta \text{ sii } \exists \alpha_1, \alpha_2, \alpha'_2, \alpha_3 \in (V \cup \Sigma)^* \text{ tq } \alpha = \alpha_1 \alpha_2 \alpha_3$$

$$\beta = \alpha_1 \alpha'_2 \alpha_3$$

$$\alpha_2 \rightarrow \alpha'_2 \in P$$

Y luego, n-pasos de derivación:

$$\alpha \xRightarrow[n]{G} \beta \text{ sii } \exists \alpha_1, \dots, \alpha_{n+1} \in (V \cup \Sigma)^* \text{ tq}$$

$$\alpha = \alpha_1 \xRightarrow{G} \dots \xRightarrow{G} \alpha_{n+1} = \beta$$

Derivación de una Cadena

Definimos formalmente la noción de 1-paso de derivación:

$$\alpha \xRightarrow{G} \beta \text{ sii } \exists \alpha_1, \alpha_2, \alpha'_2, \alpha_3 \in (V \cup \Sigma)^* \text{ tq } \alpha = \alpha_1 \alpha_2 \alpha_3$$

$$\beta = \alpha_1 \alpha'_2 \alpha_3$$

$$\alpha_2 \rightarrow \alpha'_2 \in P$$

Y luego, n-pasos de derivación:

$$\alpha \xRightarrow[n]{G} \beta \text{ sii } \exists \alpha_1, \dots, \alpha_{n+1} \in (V \cup \Sigma)^* \text{ tq}$$

$$\alpha = \alpha_1 \xRightarrow{G} \dots \xRightarrow{G} \alpha_{n+1} = \beta$$

Y finalmente, escribimos $\alpha \xRightarrow{*}{G} \beta$ sii $\alpha \xRightarrow[n]{G} \beta$ para algún $n \in \mathbb{N}$.

Lenguaje Generado de una Gramática

Son todas las cadenas de terminales que pueden ser derivadas en la gramática a partir de su variable inicial:

$$L(G) = \{\alpha \in \Sigma^* : S \xRightarrow[G]{*} \alpha\}$$

Lenguaje Generado de una Gramática

Son todas las cadenas de terminales que pueden ser derivadas en la gramática a partir de su variable inicial:

$$L(G) = \{\alpha \in \Sigma^* : S \xRightarrow[G]{*} \alpha\}$$

Para la gramática anterior:

$$G_1 : S \rightarrow aS | \epsilon$$

Tenemos que $L(G_1) = \{a^n : n \in \mathbb{N}\}$.

Lenguaje Generado de una Gramática

Son todas las cadenas de terminales que pueden ser derivadas en la gramática a partir de su variable inicial:

$$L(G) = \{\alpha \in \Sigma^* : S \xrightarrow[G]{*} \alpha\}$$

Para la gramática anterior:

$$G_1 : S \rightarrow aS | \epsilon$$

Tenemos que $L(G_1) = \{a^n : n \in \mathbb{N}\}$.

Sea G_2 la siguiente gramática:

$$\begin{aligned} G_2 : S &\rightarrow \epsilon | bS | aA \\ A &\rightarrow aS | bA \end{aligned}$$

Tenemos que $L(G_2) = \{\alpha \in \Sigma_2^* : |\alpha|_a \text{ es par}\}$.

Sea G_3 la siguiente gramática:

$$G_3 : S \rightarrow aSb \mid \epsilon$$

Tenemos que $L(G_3) = \{a^m b^m : m \in \mathbb{N}\}$.

Gramáticas Regulares

Surgen de imponer una fuerte restricción en la forma de las producciones

$\alpha \rightarrow \beta \in P$:

Gramáticas Regulares

Surgen de imponer una fuerte restricción en la forma de las producciones

$\alpha \rightarrow \beta \in P$:

- ▶ α sólo puede ser una única variable.
- ▶ β sólo puede ser una cadena de símbolos terminales seguida de a lo sumo una variable o viceversa.

Gramáticas Regulares

Surgen de imponer una fuerte restricción en la forma de las producciones $\alpha \rightarrow \beta \in P$:

- ▶ α sólo puede ser una única variable.
- ▶ β sólo puede ser una cadena de símbolos terminales seguida de a lo sumo una variable o viceversa.

Definición (Gramática Regular)

Una gramática $G = (V, \Sigma, S, P)$ es **Lineal por Derecha** si y solo si todas sus producciones son de la forma $A \rightarrow \alpha B$ o $A \rightarrow \alpha$ con $\alpha \in \Sigma^*$ y $A, B \in V$. Análogamente, una gramática $G = (V, \Sigma, S, P)$ es **Lineal por Izquierda** si y solo si todas sus producciones son de la forma $A \rightarrow B\alpha$ o $A \rightarrow \alpha$ con $\alpha \in \Sigma^*$ y $A, B \in V$.

Gramáticas Regulares

Surgen de imponer una fuerte restricción en la forma de las producciones $\alpha \rightarrow \beta \in P$:

- ▶ α sólo puede ser una única variable.
- ▶ β sólo puede ser una cadena de símbolos terminales seguida de a lo sumo una variable o viceversa.

Definición (Gramática Regular)

Una gramática $G = (V, \Sigma, S, P)$ es **Lineal por Derecha** si y solo si todas sus producciones son de la forma $A \rightarrow \alpha B$ o $A \rightarrow \alpha$ con $\alpha \in \Sigma^*$ y $A, B \in V$. Análogamente, una gramática $G = (V, \Sigma, S, P)$ es **Lineal por Izquierda** si y solo si todas sus producciones son de la forma $A \rightarrow B\alpha$ o $A \rightarrow \alpha$ con $\alpha \in \Sigma^*$ y $A, B \in V$.

Una gramática $G = (V, \Sigma, S, P)$ es una **Gramática Regular (GR)** si y solo si es una gramática lineal por derecha o una gramática lineal por izquierda.

$$GR^{\Sigma} = \{ G : G \text{ es GR con símbolos terminales } \Sigma \}$$

Ejemplos de Gramáticas Regulares

De los ejemplos anteriores, G_1 y G_2 son gramáticas regulares, mientras que G_3 no lo es pues la producción $S \rightarrow aSb$ no es lineal.

Ejemplos de Gramáticas Regulares

De los ejemplos anteriores, G_1 y G_2 son gramáticas regulares, mientras que G_3 no lo es pues la producción $S \rightarrow aSb$ no es lineal.

Sea G_4 la gramática regular:

$$\begin{aligned} G_4 : S &\rightarrow aS \mid B \\ B &\rightarrow bB \mid \epsilon \end{aligned}$$

Tenemos que $L(G_4) = \{a^n b^m : n, m \in \mathbb{N}\} = a^* b^*$

Ejemplos de Gramáticas Regulares

De los ejemplos anteriores, G_1 y G_2 son gramáticas regulares, mientras que G_3 no lo es pues la producción $S \rightarrow aSb$ no es lineal.

Sea G_4 la gramática regular:

$$\begin{aligned} G_4 : S &\rightarrow aS \mid B \\ B &\rightarrow bB \mid \epsilon \end{aligned}$$

Tenemos que $L(G_4) = \{a^n b^m : n, m \in \mathbb{N}\} = a^* b^*$

Sea G_5 la gramática regular:

$$\begin{aligned} G_5 : S &\rightarrow aS \mid bS \mid A \\ A &\rightarrow aA \mid a \end{aligned}$$

Tenemos que $L(G_5) = (a + b)^* a^+$

Probando por Inducción que $L(G) = L$

Probando por Inducción que $L(G) = L$

Sea $L = \{\alpha \in \Sigma^* : |\alpha| \text{ par}\}$ y sea G la gramática regular:

$$G : S \rightarrow aaS | abS | baS | bbS | \epsilon$$

Tenemos que $L(G) = L$.

Probando por Inducción que $L(G) = L$

Sea $L = \{\alpha \in \Sigma^* : |\alpha| \text{ par}\}$ y sea G la gramática regular:

$$G : S \rightarrow aaS | abS | baS | bbS | \epsilon$$

Tenemos que $L(G) = L$.

Pero como podemos asegurar con certeza matemática que efectivamente $L(G) = L$?

Probando por Inducción que $L(G) = L$

Sea $L = \{\alpha \in \Sigma^* : |\alpha| \text{ par}\}$ y sea G la gramática regular:

$$G : S \rightarrow aaS | abS | baS | bbS | \epsilon$$

Tenemos que $L(G) = L$.

Pero como podemos asegurar con certeza matemática que efectivamente $L(G) = L$? Debemos ver que vale la inclusión mutua $L(G) \subseteq L$ y $L \subseteq L(G)$.

Probando por Inducción que $L(G) = L$

Sea $L = \{\alpha \in \Sigma^* : |\alpha| \text{ par}\}$ y sea G la gramática regular:

$$G : S \rightarrow aaS | abS | baS | bbS | \epsilon$$

Tenemos que $L(G) = L$.

Pero como podemos asegurar con certeza matemática que efectivamente $L(G) = L$? Debemos ver que vale la inclusión mutua $L(G) \subseteq L$ y $L \subseteq L(G)$. Y lo probaremos por inducción!

Probando por Inducción que $L(G) = L$

Sea $L = \{\alpha \in \Sigma^* : |\alpha| \text{ par}\}$ y sea G la gramática regular:

$$G : S \rightarrow aaS | abS | baS | bbS | \epsilon$$

Tenemos que $L(G) = L$.

Pero como podemos asegurar con certeza matemática que efectivamente $L(G) = L$? Debemos ver que vale la inclusión mutua $L(G) \subseteq L$ y $L \subseteq L(G)$. Y lo probaremos por inducción!

Veamos que $L(G) \subseteq L$.

Probando por Inducción que $L(G) = L$

Sea $L = \{\alpha \in \Sigma^* : |\alpha| \text{ par}\}$ y sea G la gramática regular:

$$G : S \rightarrow aaS | abS | baS | bbS | \epsilon$$

Tenemos que $L(G) = L$.

Pero como podemos asegurar con certeza matemática que efectivamente $L(G) = L$? Debemos ver que vale la inclusión mutua $L(G) \subseteq L$ y $L \subseteq L(G)$. Y lo probaremos por inducción!

Veamos que $L(G) \subseteq L$.

Sea $\alpha \in L(G) \therefore S \xrightarrow[G]{*} \alpha$, con $\alpha \in \Sigma^* \therefore S \xrightarrow[G]{n} \alpha$, para algún $n \geq 1$.

Hacemos inducción en n (i.e, en la cantidad de pasos de la derivación).

Probando por Inducción que $L(G) = L$

Sea $L = \{\alpha \in \Sigma^* : |\alpha| \text{ par}\}$ y sea G la gramática regular:

$$G : S \rightarrow aaS | abS | baS | bbS | \epsilon$$

Tenemos que $L(G) = L$.

Pero como podemos asegurar con certeza matemática que efectivamente $L(G) = L$? Debemos ver que vale la inclusión mutua $L(G) \subseteq L$ y $L \subseteq L(G)$. Y lo probaremos por inducción!

Veamos que $L(G) \subseteq L$.

Sea $\alpha \in L(G) \therefore S \xrightarrow[G]{*} \alpha$, con $\alpha \in \Sigma^* \therefore S \xrightarrow[G]{n} \alpha$, para algún $n \geq 1$.

Hacemos inducción en n (i.e, en la cantidad de pasos de la derivación).

HI: $S \xrightarrow[G]{n} \alpha$, entonces $|\alpha| \in L$ (o sea, $|\alpha|$ par).

Probando por Inducción que $L(G) = L$

Sea $L = \{\alpha \in \Sigma^* : |\alpha| \text{ par}\}$ y sea G la gramática regular:

$$G : S \rightarrow aaS | abS | baS | bbS | \epsilon$$

Tenemos que $L(G) = L$.

Pero como podemos asegurar con certeza matemática que efectivamente $L(G) = L$? Debemos ver que vale la inclusión mutua $L(G) \subseteq L$ y $L \subseteq L(G)$. Y lo probaremos por inducción!

Veamos que $L(G) \subseteq L$.

Sea $\alpha \in L(G) \therefore S \xrightarrow[G]{*} \alpha$, con $\alpha \in \Sigma^* \therefore S \xrightarrow[G]{n} \alpha$, para algún $n \geq 1$.

Hacemos inducción en n (i.e, en la cantidad de pasos de la derivación).

HI: $S \xrightarrow[G]{n} \alpha$, entonces $|\alpha| \in L$ (o sea, $|\alpha|$ par).

Caso base $n = 1$:

Si $S \xrightarrow[G]{1} \alpha$, entonces $\alpha = \epsilon$ y $|\epsilon| = 0$ y es par $\therefore \epsilon \in L$.

Caso inductivo $n + 1$:

Si $S \xrightarrow[G]{n+1} \alpha \therefore S \xrightarrow[G]{1} x_1 x_2 S \xrightarrow[G]{n} \alpha$, con $x_1, x_2 \in \{a, b\} \therefore \alpha = x_1 x_2 \alpha'$, con $\alpha' \in \Sigma^* \therefore S \xrightarrow[G]{n} \alpha'$, luego por HI, $\alpha' \in L \therefore |\alpha'|$ es par $\therefore |\alpha| = |x_1 x_2 \alpha'| = 2 + |\alpha'|$ es par $\therefore \alpha \in L$.

Caso inductivo $n + 1$:

Si $S \xrightarrow[n+1]{G} \alpha \therefore S \xrightarrow[1]{G} x_1 x_2 S \xrightarrow[n]{G} \alpha$, con $x_1, x_2 \in \{a, b\} \therefore \alpha = x_1 x_2 \alpha'$, con $\alpha' \in \Sigma^* \therefore S \xrightarrow[n]{G} \alpha'$, luego por HI, $\alpha' \in L \therefore |\alpha'|$ es par $\therefore |\alpha| = |x_1 x_2 \alpha'| = 2 + |\alpha'|$ es par $\therefore \alpha \in L$.

Veamos que $L \subseteq L(G)$.

Caso inductivo $n + 1$:

Si $S \xrightarrow[n+1]{G} \alpha \therefore S \xrightarrow[1]{G} x_1 x_2 S \xrightarrow[n]{G} \alpha$, con $x_1, x_2 \in \{a, b\} \therefore \alpha = x_1 x_2 \alpha'$, con $\alpha' \in \Sigma^* \therefore S \xrightarrow[n]{G} \alpha'$, luego por HI, $\alpha' \in L \therefore |\alpha'|$ es par $\therefore |\alpha| = |x_1 x_2 \alpha'| = 2 + |\alpha'|$ es par $\therefore \alpha \in L$.

Veamos que $L \subseteq L(G)$.

Sea $\alpha \in L$ y sea $n = |\alpha|$, con $n \geq 0$.

Hacemos inducción en n (i.e, en la longitud de la cadena).

Caso inductivo $n + 1$:

Si $S \xrightarrow[G]{n+1} \alpha \therefore S \xrightarrow[G]{1} x_1 x_2 S \xrightarrow[G]{n} \alpha$, con $x_1, x_2 \in \{a, b\} \therefore \alpha = x_1 x_2 \alpha'$, con $\alpha' \in \Sigma^* \therefore S \xrightarrow[G]{n} \alpha'$, luego por HI, $\alpha' \in L \therefore |\alpha'|$ es par $\therefore |\alpha| = |x_1 x_2 \alpha'| = 2 + |\alpha'|$ es par $\therefore \alpha \in L$.

Veamos que $L \subseteq L(G)$.

Sea $\alpha \in L$ y sea $n = |\alpha|$, con $n \geq 0$.

Hacemos inducción en n (i.e, en la longitud de la cadena).

HI: $\alpha \in L$ y $|\alpha| = n$, entonces $S \xrightarrow[G]{*} \alpha$ (o sea, $\alpha \in L(G)$)

Caso inductivo $n + 1$:

Si $S \xrightarrow[G]{n+1} \alpha \therefore S \xrightarrow[G]{1} x_1 x_2 S \xrightarrow[G]{n} \alpha$, con $x_1, x_2 \in \{a, b\} \therefore \alpha = x_1 x_2 \alpha'$, con $\alpha' \in \Sigma^* \therefore S \xrightarrow[G]{n} \alpha'$, luego por HI, $\alpha' \in L \therefore |\alpha'|$ es par $\therefore |\alpha| = |x_1 x_2 \alpha'| = 2 + |\alpha'|$ es par $\therefore \alpha \in L$.

Veamos que $L \subseteq L(G)$.

Sea $\alpha \in L$ y sea $n = |\alpha|$, con $n \geq 0$.

Hacemos inducción en n (i.e, en la longitud de la cadena).

HI: $\alpha \in L$ y $|\alpha| = n$, entonces $S \xrightarrow[G]{*} \alpha$ (o sea, $\alpha \in L(G)$)

Caso base $n = 0$:

Si $0 = |\alpha|$, entonces $\alpha = \epsilon$, entonces tomamos $S \xrightarrow[G]{1} \epsilon \therefore \epsilon \in L(G)$.

Caso inductivo $n + 1$:

Si $S \xrightarrow[n+1]{G} \alpha \therefore S \xrightarrow[1]{G} x_1 x_2 S \xrightarrow[n]{G} \alpha$, con $x_1, x_2 \in \{a, b\} \therefore \alpha = x_1 x_2 \alpha'$, con $\alpha' \in \Sigma^* \therefore S \xrightarrow[n]{G} \alpha'$, luego por HI, $\alpha' \in L \therefore |\alpha'|$ es par $\therefore |\alpha| = |x_1 x_2 \alpha'| = 2 + |\alpha'|$ es par $\therefore \alpha \in L$.

Veamos que $L \subseteq L(G)$.

Sea $\alpha \in L$ y sea $n = |\alpha|$, con $n \geq 0$.

Hacemos inducción en n (i.e, en la longitud de la cadena).

HI: $\alpha \in L$ y $|\alpha| = n$, entonces $S \xrightarrow[n]{*G} \alpha$ (o sea, $\alpha \in L(G)$)

Caso base $n = 0$:

Si $0 = |\alpha|$, entonces $\alpha = \epsilon$, entonces tomamos $S \xrightarrow[1]{G} \epsilon \therefore \epsilon \in L(G)$.

Caso inductivo $n + 2$:

Si $n + 2 = |\alpha| \therefore \alpha = x_1 x_2 \alpha'$, con $\alpha' \in \Sigma^*$, $|\alpha'|$ par y $x_1, x_2 \in \{a, b\} \therefore \alpha' \in L$, luego por HI, $\alpha' \in L(G) \therefore S \xrightarrow[n]{*G} \alpha'$, entonces tomamos

$S \xrightarrow[1]{G} x_1 x_2 S \xrightarrow[n]{*G} x_1 x_2 \alpha' = \alpha \therefore \alpha \in L(G)$.

Caso inductivo $n + 1$:

Si $S \xrightarrow[n+1]{G} \alpha \therefore S \xrightarrow[1]{G} x_1 x_2 S \xrightarrow[n]{G} \alpha$, con $x_1, x_2 \in \{a, b\} \therefore \alpha = x_1 x_2 \alpha'$, con $\alpha' \in \Sigma^* \therefore S \xrightarrow[n]{G} \alpha'$, luego por HI, $\alpha' \in L \therefore |\alpha'|$ es par $\therefore |\alpha| = |x_1 x_2 \alpha'| = 2 + |\alpha'|$ es par $\therefore \alpha \in L$.

Veamos que $L \subseteq L(G)$.

Sea $\alpha \in L$ y sea $n = |\alpha|$, con $n \geq 0$.

Hacemos inducción en n (i.e, en la longitud de la cadena).

HI: $\alpha \in L$ y $|\alpha| = n$, entonces $S \xrightarrow[n]{*G} \alpha$ (o sea, $\alpha \in L(G)$)

Caso base $n = 0$:

Si $0 = |\alpha|$, entonces $\alpha = \epsilon$, entonces tomamos $S \xrightarrow[1]{G} \epsilon \therefore \epsilon \in L(G)$.

Caso inductivo $n + 2$:

Si $n + 2 = |\alpha| \therefore \alpha = x_1 x_2 \alpha'$, con $\alpha' \in \Sigma^*$, $|\alpha'|$ par y $x_1, x_2 \in \{a, b\} \therefore \alpha' \in L$, luego por HI, $\alpha' \in L(G) \therefore S \xrightarrow[n]{*G} \alpha'$, entonces tomamos

$S \xrightarrow[1]{G} x_1 x_2 S \xrightarrow[n]{*G} x_1 x_2 \alpha' = \alpha \therefore \alpha \in L(G)$.

(Este no fue el caso, pero en general, en aquellas gramáticas que utilizan más variables que la variable inicial S , debemos fortalecer la hipótesis inductiva!!!).

Equivalencia entre Lenguajes Regulares y Gramáticas Regulares

- ▶ Queremos probar que los lenguajes regulares y las gramáticas regulares son equivalentes, $LR^{\Sigma} \equiv GR^{\Sigma}$.

Equivalencia entre Lenguajes Regulares y Gramáticas Regulares

- ▶ Queremos probar que los lenguajes regulares y las gramáticas regulares son equivalentes, $LR^{\Sigma} \equiv GR^{\Sigma}$.
- ▶ Para probarlo de manera sencilla necesitamos primero introducir el concepto de gramática regular de paso único (GRPU).

Equivalencia entre Lenguajes Regulares y Gramáticas Regulares

- ▶ Queremos probar que los lenguajes regulares y las gramáticas regulares son equivalentes, $LR^{\Sigma} \equiv GR^{\Sigma}$.
- ▶ Para probarlo de manera sencilla necesitamos primero introducir el concepto de gramática regular de paso único (GRPU).
- ▶ Probaremos que las GR y las GRPU son equivalentes.

Equivalencia entre Lenguajes Regulares y Gramáticas Regulares

- ▶ Queremos probar que los lenguajes regulares y las gramáticas regulares son equivalentes, $LR^{\Sigma} \equiv GR^{\Sigma}$.
- ▶ Para probarlo de manera sencilla necesitamos primero introducir el concepto de gramática regular de paso único (GRPU).
- ▶ Probaremos que las GR y las GRPU son equivalentes.
- ▶ Por último, probaremos que los AF y las GRPU son equivalentes, y así, por transitividad, habremos demostrado que $LR^{\Sigma} \equiv GR^{\Sigma}$.

Gramáticas Regulares de Paso Unico (GRPU)

- ▶ Surgen de imponer una restricción aún más fuerte a la forma de las producciones de una gramática regular.

Gramáticas Regulares de Paso Unico (GRPU)

- ▶ Surgen de imponer una restricción aún más fuerte a la forma de las producciones de una gramática regular.
- ▶ Por lo tanto, son un caso particular de gramáticas regulares.

Gramáticas Regulares de Paso Unico (GRPU)

- ▶ Surgen de imponer una restricción aún más fuerte a la forma de las producciones de una gramática regular.
- ▶ Por lo tanto, son un caso particular de gramáticas regulares.

Definición (Gramática Regular de Paso Unico)

Una gramática $G = (V, \Sigma, S, P)$ es una **Gramática Regular de Paso Unico (GRPU)** si y solo si todas sus producciones son de la forma $A \rightarrow aB$ o $A \rightarrow B$ o $A \rightarrow \epsilon$, con $a \in \Sigma$ y $A, B \in V$.

$$GRPU^{\Sigma} = \{ G : G \text{ es GRPU con simbolos terminales } \Sigma \}$$

Gramáticas Regulares de Paso Unico (GRPU)

- ▶ Surgen de imponer una restricción aún más fuerte a la forma de las producciones de una gramática regular.
- ▶ Por lo tanto, son un caso particular de gramáticas regulares.

Definición (Gramática Regular de Paso Unico)

Una gramática $G = (V, \Sigma, S, P)$ es una **Gramática Regular de Paso Unico (GRPU)** si y solo si todas sus producciones son de la forma $A \rightarrow aB$ o $A \rightarrow B$ o $A \rightarrow \epsilon$, con $a \in \Sigma$ y $A, B \in V$.

$$GRPU^{\Sigma} = \{ G : G \text{ es GRPU con simbolos terminales } \Sigma \}$$

De las anteriores, G_1 , G_2 , G_4 y G_5 son gramáticas regulares de paso unico.

Gramáticas Regulares de Paso Unico (GRPU)

- ▶ Surgen de imponer una restricción aún más fuerte a la forma de las producciones de una gramática regular.
- ▶ Por lo tanto, son un caso particular de gramáticas regulares.

Definición (Gramática Regular de Paso Unico)

Una gramática $G = (V, \Sigma, S, P)$ es una **Gramática Regular de Paso Unico (GRPU)** si y solo si todas sus producciones son de la forma $A \rightarrow aB$ o $A \rightarrow B$ o $A \rightarrow \epsilon$, con $a \in \Sigma$ y $A, B \in V$.

$$GRPU^{\Sigma} = \{ G : G \text{ es GRPU con simbolos terminales } \Sigma \}$$

De las anteriores, G_1 , G_2 , G_4 y G_5 son gramáticas regulares de paso unico.

Mientras que la gramática anterior G con $L(G) = \{ \alpha \in \Sigma^* : |\alpha| \text{ par} \}$ no es de paso único pero si es regular:

$$G : S \rightarrow aaS | abS | baS | bbS | \epsilon$$

GRPU vs GR

En una gramática regular podemos derivar con la aplicación de una sola producción una cadena α con k símbolos:

$$S \xRightarrow{1} \alpha A, \text{ con } \alpha = a_1 \dots a_k$$

GRPU vs GR

En una gramática regular podemos derivar con la aplicación de una sola producción una cadena α con k símbolos:

$$S \xRightarrow{1} \alpha A, \text{ con } \alpha = a_1 \dots a_k$$

Mientras que en una gramática regular de paso único derivamos α pero aplicando k producciones:

$$S \xRightarrow{1} a_1 A_1 \xRightarrow{1} a_1 a_2 A_2 \dots \xRightarrow{1} a_1 a_2 \dots a_k A_k \xRightarrow{1} a_1 a_2 \dots a_k A = \alpha A \quad \therefore$$

$$S \xRightarrow{k} \alpha A$$

GRPU vs GR

En una gramática regular podemos derivar con la aplicación de una sola producción una cadena α con k símbolos:

$$S \xRightarrow{1} \alpha A, \text{ con } \alpha = a_1 \dots a_k$$

Mientras que en una gramática regular de paso único derivamos α pero aplicando k producciones:

$$S \xRightarrow{1} a_1 A_1 \xRightarrow{1} a_1 a_2 A_2 \dots \xRightarrow{1} a_1 a_2 \dots a_k A_k \xRightarrow{1} a_1 a_2 \dots a_k A = \alpha A \quad \therefore$$

$$S \xRightarrow{k} \alpha A$$

Por lo tanto, las GRPU pueden generar las mismas cadenas que una GR pero símbolo a símbolo.

GRPU vs GR

En una gramática regular podemos derivar con la aplicación de una sola producción una cadena α con k símbolos:

$$S \xRightarrow{1} \alpha A, \text{ con } \alpha = a_1 \dots a_k$$

Mientras que en una gramática regular de paso único derivamos α pero aplicando k producciones:

$$S \xRightarrow{1} a_1 A_1 \xRightarrow{1} a_1 a_2 A_2 \dots \xRightarrow{1} a_1 a_2 \dots a_k A_k \xRightarrow{1} a_1 a_2 \dots a_k A = \alpha A \quad \therefore$$

$$S \xRightarrow{k} \alpha A$$

Por lo tanto, las GRPU pueden generar las mismas cadenas que una GR pero símbolo a símbolo.

Las GRPU pueden ser vistas como la versión “bit a bit” de las GR!!!

Equivalencia entre GRPU y GR

Las gramáticas regulares y las gramáticas regulares de paso único generan exactamente los mismos lenguajes.

Equivalencia entre GRPU y GR

Las gramáticas regulares y las gramáticas regulares de paso único generan exactamente los mismos lenguajes.

Teorema ($GRPU^{\Sigma} \equiv GR^{\Sigma}$)

1. (ida) Si $G \in GRPU^{\Sigma}$, entonces $\exists G' \in GR^{\Sigma}$ tal que $L(G') = L(G)$.
2. (vuelta) Si $G \in GR^{\Sigma}$, entonces $\exists G' \in GRPU^{\Sigma}$ tal que $L(G') = L(G)$.

Equivalencia entre GRPU y GR

Las gramáticas regulares y las gramáticas regulares de paso único generan exactamente los mismos lenguajes.

Teorema ($GRPU^{\Sigma} \equiv GR^{\Sigma}$)

1. (ida) Si $G \in GRPU^{\Sigma}$, entonces $\exists G' \in GR^{\Sigma}$ tal que $L(G') = L(G)$.
2. (vuelta) Si $G \in GR^{\Sigma}$, entonces $\exists G' \in GRPU^{\Sigma}$ tal que $L(G') = L(G)$.

Demo 1. Trivial, pues una GRPU es en particular una GR.

Equivalencia entre GRPU y GR

Las gramáticas regulares y las gramáticas regulares de paso único generan exactamente los mismos lenguajes.

Teorema ($GRPU^{\Sigma} \equiv GR^{\Sigma}$)

1. (ida) Si $G \in GRPU^{\Sigma}$, entonces $\exists G' \in GR^{\Sigma}$ tal que $L(G') = L(G)$.
2. (vuelta) Si $G \in GR^{\Sigma}$, entonces $\exists G' \in GRPU^{\Sigma}$ tal que $L(G') = L(G)$.

Demo 1. Trivial, pues una GRPU es en particular una GR.

Demo 2. Debemos reemplazar con producciones de paso único aquellas producciones de G que no lo son (y además, preservar las que si lo son).

Equivalencia entre GRPU y GR

Las gramáticas regulares y las gramáticas regulares de paso único generan exactamente los mismos lenguajes.

Teorema ($GRPU^{\Sigma} \equiv GR^{\Sigma}$)

1. (ida) Si $G \in GRPU^{\Sigma}$, entonces $\exists G' \in GR^{\Sigma}$ tal que $L(G') = L(G)$.
2. (vuelta) Si $G \in GR^{\Sigma}$, entonces $\exists G' \in GRPU^{\Sigma}$ tal que $L(G') = L(G)$.

Demo 1. Trivial, pues una GRPU es en particular una GR.

Demo 2. Debemos reemplazar con producciones de paso único aquellas producciones de G que no lo son (y además, preservar las que si lo son).

Sean P y P' las producciones de G y G' , respectivamente.

Si $(A \rightarrow a_1 \dots a_k B) \in P$, con $k \geq 2$, entonces en P' hacemos:

$$A \rightarrow A_{a_1}$$

$$A_{a_1} \rightarrow a_1 A_{a_2}$$

$$A_{a_2} \rightarrow a_2 A_{a_3}$$

$$\vdots$$

$$A_{a_k} \rightarrow a_k A_{a_{k+1}}$$

$$A_{a_{k+1}} \rightarrow B$$

Si $(A \rightarrow a_1 \dots a_k B) \in P$, con $k \geq 2$, entonces en P' hacemos:

$$A \rightarrow A_{a_1}$$

$$A_{a_1} \rightarrow a_1 A_{a_2}$$

$$A_{a_2} \rightarrow a_2 A_{a_3}$$

$$\vdots$$

$$A_{a_k} \rightarrow a_k A_{a_{k+1}}$$

$$A_{a_{k+1}} \rightarrow B$$

Si $(A \rightarrow a_1 \dots a_k) \in P$, con $k \geq 2$, entonces en P' hacemos:

$$A \rightarrow A_{a_1}$$

$$A_{a_1} \rightarrow a_1 A_{a_2}$$

$$A_{a_2} \rightarrow a_2 A_{a_3}$$

$$\vdots$$

$$A_{a_k} \rightarrow a_k A_{a_{k+1}}$$

$$A_{a_{k+1}} \rightarrow \epsilon$$

Si $(A \rightarrow a_1 \dots a_k B) \in P$, con $k \geq 2$, entonces en P' hacemos:

$$\begin{aligned} A &\rightarrow A_{a_1} \\ A_{a_1} &\rightarrow a_1 A_{a_2} \\ A_{a_2} &\rightarrow a_2 A_{a_3} \\ &\vdots \\ A_{a_k} &\rightarrow a_k A_{a_{k+1}} \\ A_{a_{k+1}} &\rightarrow B \end{aligned}$$

Si $(A \rightarrow a_1 \dots a_k) \in P$, con $k \geq 2$, entonces en P' hacemos:

$$\begin{aligned} A &\rightarrow A_{a_1} \\ A_{a_1} &\rightarrow a_1 A_{a_2} \\ A_{a_2} &\rightarrow a_2 A_{a_3} \\ &\vdots \\ A_{a_k} &\rightarrow a_k A_{a_{k+1}} \\ A_{a_{k+1}} &\rightarrow \epsilon \end{aligned}$$

Claramente, G' es GRPU, y además, $\alpha \in L(G)$ sii $\alpha \in L(G') \therefore L(G) = L(G')$.

Probando que $LR^\Sigma \equiv GR^\Sigma$

Un lenguaje es regular si y solo si es generado por una gramática regular.

Probando que $LR^\Sigma \equiv GR^\Sigma$

Un lenguaje es regular si y solo si es generado por una gramática regular.

“Una interpretación posible de este resultado es que las GR son el modelo computacional, además de los AF, que caracterizan a los lenguajes regulares.”

- ▶ Ya probamos que $LR^\Sigma \equiv AF^\Sigma$ y $GR^\Sigma \equiv GRPU^\Sigma$.

Probando que $LR^\Sigma \equiv GR^\Sigma$

Un lenguaje es regular si y solo si es generado por una gramática regular.

“Una interpretación posible de este resultado es que las GR son el modelo computacional, además de los AF, que caracterizan a los lenguajes regulares.”

- ▶ Ya probamos que $LR^\Sigma \equiv AF^\Sigma$ y $GR^\Sigma \equiv GRPU^\Sigma$.
- ▶ Ahora, probaremos que $GRPU^\Sigma \equiv AF^\Sigma$.

Probando que $LR^\Sigma \equiv GR^\Sigma$

Un lenguaje es regular si y solo si es generado por una gramática regular.

“Una interpretación posible de este resultado es que las GR son el modelo computacional, además de los AF, que caracterizan a los lenguajes regulares.”

- ▶ Ya probamos que $LR^\Sigma \equiv AF^\Sigma$ y $GR^\Sigma \equiv GRPU^\Sigma$.
- ▶ Ahora, probaremos que $GRPU^\Sigma \equiv AF^\Sigma$.
- ▶ Y por transitividad, tendremos que $LR^\Sigma \equiv GR^\Sigma$.

Probando que $LR^\Sigma \equiv GR^\Sigma$

Un lenguaje es regular si y solo si es generado por una gramática regular.

“Una interpretación posible de este resultado es que las GR son el modelo computacional, además de los AF, que caracterizan a los lenguajes regulares.”

- ▶ Ya probamos que $LR^\Sigma \equiv AF^\Sigma$ y $GR^\Sigma \equiv GRPU^\Sigma$.
- ▶ Ahora, probaremos que $GRPU^\Sigma \equiv AF^\Sigma$.
- ▶ Y por transitividad, tendremos que $LR^\Sigma \equiv GR^\Sigma$.

Teorema ($GRPU^\Sigma \equiv AF^\Sigma$)

1. (ida) Si $G \in GRPU^\Sigma$, entonces $\exists M \in AF^\Sigma$ tal que $L(G) = L(M)$.
2. (vuelta) Si $M \in AF^\Sigma$, entonces $\exists G \in GRPU^\Sigma$ tal que $L(G) = L(M)$.

Demo 1. Sea $G = (V, \Sigma, S, P)$, entonces podemos tomar $M = (\Sigma, Q, q_0, F, \Delta)$ donde:

Demo 1. Sea $G = (V, \Sigma, S, P)$, entonces podemos tomar $M = (\Sigma, Q, q_0, F, \Delta)$ donde:

$$Q = V$$

$$q_0 = S$$

$$F = \{A \in V : A \rightarrow \epsilon \in P\}$$

$$\Delta(A, a) = \{B \in V : A \rightarrow aB \in P\}$$

$$\Delta(A, \epsilon) = \{B \in V : A \rightarrow B \in P\}$$

Demo 1. Sea $G = (V, \Sigma, S, P)$, entonces podemos tomar $M = (\Sigma, Q, q_0, F, \Delta)$ donde:

$$Q = V$$

$$q_0 = S$$

$$F = \{A \in V : A \rightarrow \epsilon \in P\}$$

$$\Delta(A, a) = \{B \in V : A \rightarrow aB \in P\}$$

$$\Delta(A, \epsilon) = \{B \in V : A \rightarrow B \in P\}$$

Claramente, M es un AFN ϵ .

Luego, $\alpha \in L(G)$ sii $\alpha \in L(M) \therefore L(G) = L(M)$.

Por ejemplo, sea G la siguiente GRPU:

$$G : S \rightarrow aS|B|\epsilon$$

$$B \rightarrow bB|C|\epsilon$$

$$C \rightarrow cC|\epsilon$$

con $L(G) = a^*b^*c^*$.

Por ejemplo, sea G la siguiente GRPU:

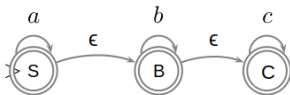
$$G : S \rightarrow aS \mid B \mid \epsilon$$

$$B \rightarrow bB \mid C \mid \epsilon$$

$$C \rightarrow cC \mid \epsilon$$

con $L(G) = a^*b^*c^*$.

Su AFN ϵ equivalente M sería:



Y claramente, $L(M) = a^*b^*c^*$.

Demo 2. Sea $M = (\Sigma, Q, q_0, F, \Delta)$ AFN ϵ , entonces podemos tomar $G = (V, \Sigma, S, P)$ donde:

Demo 2. Sea $M = (\Sigma, Q, q_0, F, \Delta)$ AFN ϵ , entonces podemos tomar $G = (V, \Sigma, S, P)$ donde:

$$V = Q$$

$$S = q_0$$

$$P = \{q \rightarrow aq' : q' \in \Delta(q, a)\} \cup \\ \{q \rightarrow q' : q' \in \Delta(q, \epsilon)\} \cup \\ \{q \rightarrow \epsilon : q \in F\}$$

Demo 2. Sea $M = (\Sigma, Q, q_0, F, \Delta)$ AFN ϵ , entonces podemos tomar $G = (V, \Sigma, S, P)$ donde:

$$V = Q$$

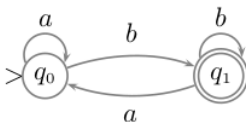
$$S = q_0$$

$$P = \{q \rightarrow aq' : q' \in \Delta(q, a)\} \cup \\ \{q \rightarrow q' : q' \in \Delta(q, \epsilon)\} \cup \\ \{q \rightarrow \epsilon : q \in F\}$$

Claramente, G es una GRPU.

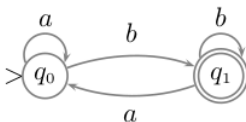
Luego, $\alpha \in L(M)$ sii $\alpha \in L(G) \therefore L(G) = L(M)$.

Por ejemplo, sea M el siguiente AF:



con $L(M) = (a + b)^* b$.

Por ejemplo, sea M el siguiente AF:



con $L(M) = (a + b)^* b$.

Su GRPU G equivalente con variable inicial q_0 sería:

$$\begin{aligned} G : q_0 &\rightarrow aq_0 \mid bq_1 \\ q_1 &\rightarrow bq_1 \mid aq_0 \mid \epsilon \end{aligned}$$

Y claramente, $L(G) = (a + b)^* b$.

Resumen

- ▶ Presentamos el modelo gramatical propuesto por Chomsky, el cual define cada una de las clases de lenguajes en su jerarquía.

Resumen

- ▶ Presentamos el modelo gramatical propuesto por Chomsky, el cual define cada una de las clases de lenguajes en su jerarquía.
- ▶ Luego, definimos la noción de gramática regular, a partir de una fuerte restricción en la forma de las producciones de la gramática.

Resumen

- ▶ Presentamos el modelo gramatical propuesto por Chomsky, el cual define cada una de las clases de lenguajes en su jerarquía.
- ▶ Luego, definimos la noción de gramática regular, a partir de una fuerte restricción en la forma de las producciones de la gramática.
- ▶ Probamos que un lenguaje es regular si y solo si es generado por una gramática regular.

Resumen

- ▶ Presentamos el modelo gramatical propuesto por Chomsky, el cual define cada una de las clases de lenguajes en su jerarquía.
- ▶ Luego, definimos la noción de gramática regular, a partir de una fuerte restricción en la forma de las producciones de la gramática.
- ▶ Probamos que un lenguaje es regular si y solo si es generado por una gramática regular.
- ▶ Por lo tanto, las gramáticas regulares son también, al igual que los AF, un modelo computacional caracterizador de los lenguajes regulares.

Resumen

- ▶ Presentamos el modelo gramatical propuesto por Chomsky, el cual define cada una de las clases de lenguajes en su jerarquía.
- ▶ Luego, definimos la noción de gramática regular, a partir de una fuerte restricción en la forma de las producciones de la gramática.
- ▶ Probamos que un lenguaje es regular si y solo si es generado por una gramática regular.
- ▶ Por lo tanto, las gramáticas regulares son también, al igual que los AF, un modelo computacional caracterizador de los lenguajes regulares.
- ▶ Tenemos una nueva herramienta para probar regularidad de un lenguaje simplemente dando una GR que lo genera.

Bibliografía



Sergio Balari.

“Teoría de los Lenguajes Formales”. Una introducción para lingüistas.

Capítulo II: sección 3.2.2.



Rodrigo De Castro Korgi.

“Teoría de la Computación”. Lenguajes, Autómatas, Gramáticas.

Capítulo 4: sección 4.7.