Autómatas y Lenguajes Formales
Nota 08 Gramáticas regulares y Forma Normal de Chomsky*

Noé Salomón Hernández S.

1. Gramáticas regulares

Definición 1.1 Una gramática $G = (V, T, P, S)$ es **regular** si cada regla de producción es de la forma $A \longrightarrow \sigma B$ ó $A \longrightarrow \varepsilon$, donde $A, B \in V$ y $\sigma \in T$.

Teorema 1.2 *Para todo lenguaje $L \subseteq T^*$, L es regular si y solo si $L = \mathcal{L}(G)$ para alguna gramática regular G.*

Demostración.

$\Rightarrow$ Si L es un lenguaje regular, entonces sea $M = (Q, \Sigma, \delta, q_0, F)$ el autómata finito tal que $L = L(M)$. Definimos $G = (V, T, P, S)$ tomando $V = Q$, $T = \Sigma$, $S = q_0$, y P el conjunto que contiene a las producciones:

- $A \longrightarrow \sigma B$ para cada transición $\delta(A, \sigma) = B$ en M, y
- $A \longrightarrow \varepsilon$ para cada estado final A en M.

Así, G es la gramática regular que buscamos. Se deja como ejercicio verificar que para toda cadena w, $\delta(q_0, w) \in F \Leftrightarrow S \Rightarrow^* w$.

$\Leftarrow$ Si $G = (V, T, P, S)$ es una gramática regular con $L = \mathcal{L}(G)$, podemos construir un autómata finito (AF) $M = (Q, \Sigma, \delta, q_0, F)$ que reconozca el mismo lenguaje. Definimos $Q = V$, $\Sigma = T$, $q_0 = S$, F como el conjunto de estados (variables) para las cuales se tiene en G una producción con ε del lado derecho, y para cada producción $A \longrightarrow \sigma B$ hay una transición $B \in \delta(A, \sigma)$ en M. Como es posible tener producciones $A \longrightarrow \sigma B$ con distintas Bs del lado derecho, entonces el AF M es un AFN.

Veremos ahora que para toda cadena w, $S \Rightarrow^* wA$ syss $A \in \widehat{\delta}(S, w)$. La demostración es por inducción estructural sobre la cadena w.

- Base: $w = \varepsilon$. Así, $S \Rightarrow^* S$ syss $S \in \widehat{\delta}(S, \varepsilon)$, esto se cumple por definición de $\Rightarrow^*$ y de $\widehat{\delta}$.

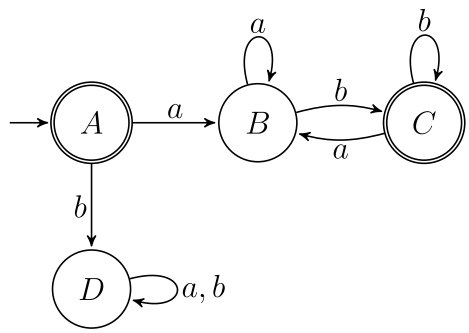
*Esta nota se basa en el libro: J. Martin. *Introduction to Languages and the Theory of Computation*, en las notas del prof. Favio Miranda y en las notas del prof. Rajeev Motwani.

- Paso inductivo: $w = x\sigma$. De modo que $S \Rightarrow^* xA \Rightarrow x\sigma B$, esto indica que se tiene la regla de producción $A \longrightarrow \sigma B$, por lo que $B \in \delta(A, \sigma)$. Además, por hipótesis de inducción, $S \Rightarrow^* wA$ syss $A \in \widehat{\delta}(S, w)$. Así se cumple que $B \in \widehat{\delta}(S, w\sigma)$. Por lo tanto, $S \Rightarrow^* x\sigma B$ syss $B \in \widehat{\delta}(S, x\sigma)$.

  Observe que si se continua la derivación $S \Rightarrow^* wA$ usando por último una regla de producción $A \longrightarrow \varepsilon$ para llegar a $S \Rightarrow^* w$, entonces $w \in \mathcal{L}(G)$. También ocurre que $w \in L(M)$ pues se tiene que $A \in \widehat{\delta}(S, w)$, y A tiene que ser un estado final pues $A \longrightarrow \varepsilon$. $\dashv$

Ejemplo 1.3 Dados los siguientes autómatas finitos encuentre gramáticas regulares correspondientes que generen el mismo lenguaje. Estos ejemplos se desarrollan a dos columnas.

a) Sea M_1 el siguiente autómata:

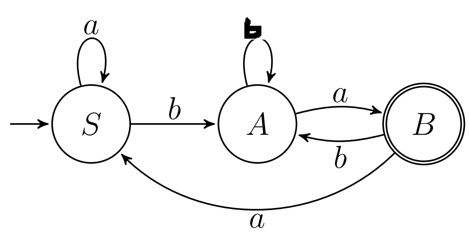


De acuerdo a la construcción descrita en la demostración de $\Rightarrow$ del teorema anterior, definimos la gramática regular $G_1 = (V_1, T_1, S_1, P_1)$ como sigue. Así $L(M_1) = \mathcal{L}(G_1)$.

- $V_1 = \{A, B, C, D\}$,
- $T_1 = \{a, b\}$,
- $S_1 = A$,
- $P_1 = \begin{cases} A \to aB \mid bD \mid \varepsilon & B \to aB \mid bC \\ C \to aB \mid bC \mid \varepsilon & D \to aD \mid bD \end{cases}$

---

b) Sea M_2 el siguiente autómata:



Siguiendo la construcción descrita en la demostración de $\Rightarrow$ del teorema anterior, definimos la gramática regular $G_2 = (V_2, T_2, S_2, P_2)$ como:

- $V_2 = \{S, A, B\}$,
- $T_2 = \{a, b\}$,
- $S_2 = S$,
- $P_2 = \begin{cases} S \to aS \mid bA & A \to bA \mid aB \\ B \to bA \mid aS \mid \varepsilon \end{cases}$

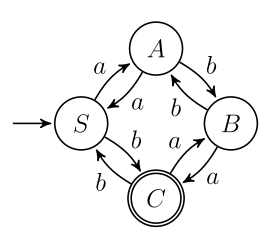
Por lo tanto, $L(M_2) = \mathcal{L}(G_2)$.

Ejemplo 1.4 A partir de las siguientes gramáticas regulares encuentre autómatas finitos que reconozcan el mismo lenguaje. Estos ejemplos se desarrollan a dos columnas.

I) Considere la gramática

$$\begin{array}{llll} G_3: & S & \to & a\,A \mid b\,C \\ & A & \to & a\,S \mid b\,B \\ & B & \to & a\,C \mid b\,A \\ & C & \to & a\,B \mid b\,S \mid \varepsilon \end{array}$$

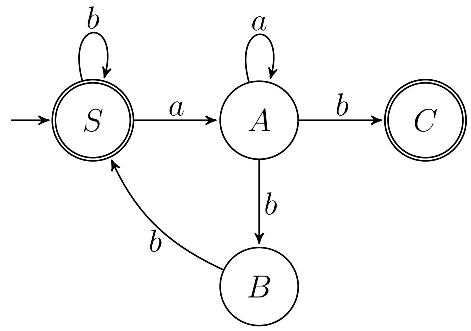
Por la construcción descrita en la demostración de $\Leftarrow$ del teorema anterior, encontramos el AF M_3 que está abajo. Así $\mathcal{L}(G_3) = L(M_3)$.



II) Considere la gramática:

$$
\begin{array}{llcl}
G_4: & S & \to & b\,S \,|\, a\,A \,|\, \varepsilon \\
 & A & \to & a\,A \,|\, b\,B \,|\, b\,C \\
 & B & \to & b\,S \\
 & C & \to & \varepsilon
\end{array}
$$

Siguiendo la construcción dada en la demostración de $\Leftarrow$ del teorema anterior, encontramos el AF M_4 de la derecha. Así $\mathcal{L}(G_4) = L(M_4)$.



2. Simplificando gramáticas

Nuestra meta es eliminar construcciones de gramáticas que retrasan a los analizadores sintácticos, como resultado estableceremos una forma normal.

2.1. Variables inútiles

Una variable útil es cualquier $X \in V$ tal que $S \Rightarrow^* \alpha X \beta \Rightarrow^* w$ con $w \in T^*$ y $\alpha, \beta \in (V \cup T)^*$. Así, variables inútiles son aquellas que no participan en derivaciones de cadenas en el lenguaje, por lo que pueden ser eliminadas de manera segura sin alterar el lenguaje.

Definición 2.1 $X \in V$ es variable **generadora** si $X \Rightarrow^* w$, con $w \in T^*$.

Definición 2.2 $X \in V$ es variable **alcanzable** si $S \Rightarrow^* \alpha X \beta$, con $\alpha, \beta \in (V \cup T)^*$.

Claramente, una variable útil es tanto generadora como alcanzable. Así, nuestro objetivo es eliminar variables inútiles al quitar:

1. Variables no generadoras, y todas sus producciones.
2. Variables no alcanzables, y todas sus producciones.

Este orden debe respetarse para efectivamente quitar las variable inútiles.

Estos dos pasos se realizan de manera recursiva como se indica enseguida:

Paso 1. Eliminar variables no generadoras.

Base. Etiquetar toda variable A como *generadora* si se tiene una regla $A \longrightarrow w$, con w una cadena, posiblemente vacía, de terminales únicamente.

Inducción. Etiquetar a la variable X como *generadora* si tiene una producción $X \longrightarrow w$, donde w es una cadena compuesta únicamente de terminales y/o variables etiquetadas previamente como *generadoras*.

Término. Cuando no se etiqueten nuevas variables *generadoras*. Eliminar aquellas variables no etiquetadas y sus producciones.

Paso 2. Eliminar variables no alcanzables.

Base. Etiquetar a S como *alcanzable*.

Inducción. Etiquetar a Y como *alcanzable* si hay una producción $X \longrightarrow w$, donde X es una variable previamente etiquetada como *alcanzable* y w es una cadena donde figura Y.

Término. Cuando no se etiqueten nuevas variables *alcanzables*. Eliminar aquellas variables no etiquetadas y sus producciones.

Ejemplo 2.3 Elimine las variables inútiles de la siguiente gramática:

$$\begin{array}{rcl} G_1 : S & \longrightarrow & AB \,|\, AC \,|\, CD \\ A & \longrightarrow & BB \\ B & \longrightarrow & AC \,|\, ab \\ C & \longrightarrow & Ca \,|\, CC \\ D & \longrightarrow & BC \,|\, b \,|\, d \end{array}$$

1. Eliminar variables no generadoras de G_1.

   Base Las variables etiquetadas son $\{B, D\}$.

   Inducción Los pasos de inducción etiquetan las siguientes variables:

   $$\{B, D, A\}$$
   $$\{B, D, A, S\}$$

   Término Ya no se etiquetan más variables. El conjunto de variables generadoras es $\{B, D, A, S\}$. La variable C no es generadora, la eliminamos junto con sus producciones. Llegamos a:

   $$\begin{array}{rcl} G_2 : S & \longrightarrow & AB \\ A & \longrightarrow & BB \\ B & \longrightarrow & ab \\ D & \longrightarrow & b \,|\, d \end{array}$$

2. Eliminar variables no alcanzables de G_2.

   Base La variables S es alcanzable.

   Inducción Las variables etiquetadas como alcanzables al ejecutar los pasos de inducción son:

   $$\{S, A, B\}$$

   Término No hay nuevas variables alcanzables. Por lo tanto, las variables alcanzables de la gramática G_2 son $\{S, A, B\}$. Eliminamos D por no ser alcanzable, también quitamos sus producciones. Así, obtenemos la gramática G_3 sin variables inútiles.

   $$\begin{array}{rcl} G_3 : S & \longrightarrow & AB \\ A & \longrightarrow & BB \\ B & \longrightarrow & ab \end{array}$$

El orden es importante, si primero elimináramos las variables no alcanzables y luego las no generadoras obtendríamos una gramática con símbolos inútiles aún. Por ejemplo, en la gramática G_1 todas sus variables son alcanzables, si ahora quitamos las variables no generadoras obtendríamos como gramática sin variables inútiles G_2, lo cual es incorrecto.

2.2. Producciones-ε

Una producción-ε es de la forma $A \longrightarrow \varepsilon$. Al igual que las variables inútiles, las producciones ε retrasan a los analizadores sintácticos.

Definición 2.4 $X \in V$ es **anulable** si $X \Rightarrow^* \varepsilon$.

La siguiente subrutina encuentra las variables anulables recursivamente:

Subrutina. Identificar las variables anulables.

Base. Si el conjunto de producciones P contiene a $A \longrightarrow \varepsilon$, entonces etiquetamos a A como *anulable*.

Inducción. Para toda producción $X \longrightarrow X_1X_2\cdots X_k$, si cada X_i es *anulable*, entonces etiquetamos a X como a *anulable*.

Término. Cuando no se etiqueten nuevas variables *anulables*.

Esta subrutina es empleada por el siguiente algoritmo para eliminar producciones-ε.

Algoritmo. Eliminar producciones-ε.

---

a) Identificar las variables anulables usando la subrutina arriba descrita.

b) Reemplazar cada producción $X \longrightarrow X_1X_2\cdots X_k$ por el conjunto de producciones de la forma $X \longrightarrow \alpha_1\alpha_2\cdots\alpha_k$, donde

- $\alpha_i = X_i$, si X_i no es anulable.
- $\alpha_i \in \{X_i, \varepsilon\}$, si X_i es anulable, de modo que no todas las $\alpha_i s$ son ε.

c) Eliminar las producciones-ε.

Ejemplo 2.5 Elimine las producciones-ε de la siguiente gramática G:

$$\begin{array}{rcl} S & \longrightarrow & ABC \mid BCB \\ A & \longrightarrow & aB \mid a \end{array} \qquad \begin{array}{rcl} B & \longrightarrow & CC \mid b \\ C & \longrightarrow & S \mid \varepsilon \end{array}$$

Para este fin, ejecutamos la subrutina que nos permite reconocer a las variables anulables.

Base Etiquetamos a C como anulable.

Inducción Las variables que resultan anulables ejecutando los pasos de inducción son:

$$\{C, B\}$$
$$\{C, B, S\}$$

Término Ya no hay más variables anulables, éstas son $\{C, B, S\}$.

De acuerdo con el algoritmo que elimina producciones-ε, examinamos las producciones donde figuran variables anulables analizando los casos donde dichas variables se conservan o son retiradas, generando así nuevas producciones, cuidando que no se retiren todos los elementos del lado derecho de la producción. Luego eliminamos todas las producciones-ε. A continuación se muestra el resultado de realizar este proceso sobre la gramática original G.

$$\begin{array}{lllllll} S & \longrightarrow & ABC\,|\,AB\,|\,AC\,|\,A\,|\,BCB\,|\,CB\,|\,BB\,|\,BC\,|\,B\,|\,C & \qquad & B & \longrightarrow & CC\,|\,b\,|\,C \\ A & \longrightarrow & aB\,|\,a & & C & \longrightarrow & S \end{array}$$

Notemos un inconveniente ya que si originalmente $S \Rightarrow^* \varepsilon$ era posible, después de haber ejecutado el algoritmo quitamos a ε de $\mathcal{L}(G)$, esto es inevitable.

2.3. Producciones unitarias

Definición 2.6 Una producción es **unitaria** si es de la forma $A \longrightarrow B$. Este tipo de producciones también retrasan a los analizadores sintácticos.

Para eliminar producciones unitarias primero necesitamos encontrar las variables Y etiquetadas con $unitaria_X$. Tales variables son el resultado de derivaciones unitarias $X \Rightarrow^* Y$. Observemos que al eliminar producciones-ε primero, hay un sólo modo de derivar $X \Rightarrow^* Y$, el cual es: $X \Rightarrow Y_1 \Rightarrow Y_2 \Rightarrow \ldots \Rightarrow Y_k \Rightarrow Y$, con $X, Y_1, Y_2, \ldots, Y_k, Y \in V$. Las variables Y para las que se tienen derivaciones unitarias $X \Rightarrow^* Y$ se encuentran mediante la subrutina:

Subrutina. Identificar las variables etiquetadas con unitaria$_X$.

Base. La variable X se etiqueta como $unitaria_X$.

Inducción. Etiquetar a Y como $unitaria_X$ si existe $Z \in V$ tal que $Z \longrightarrow Y$ y Z es $unitaria_X$.

Término. Cuando no se etiqueten nuevas variables como $unitaria_X$. Se regresa el conjunto de variables con etiqueta $unitaria_X$.

El siguiente algoritmo elimina las producciones unitarias.

Algoritmo. Eliminar producciones unitarias.

Paso 1. Eliminar producciones-ε.

Paso 2. Para toda $X, Y \in V$, con $X \neq Y$, agregar la producción nueva $X \longrightarrow \alpha$ si se tiene $Y \longrightarrow \alpha$ como una producción no unitaria, dado que Y tiene etiqueta $unitaria_X$ de acuerdo a la subrutina anterior.

Paso 3. Eliminar todas las producciones unitarias.

Ejemplo 2.7 Elimine las producciones unitarias de la siguiente gramática:

$$\begin{aligned} S &\longrightarrow A \mid B \\ A &\longrightarrow Sa \mid a \\ B &\longrightarrow S \mid b \end{aligned}$$

Primeramente, eliminamos las producciones ε, lo cual ya está hecho. Enseguida encontramos:

$$\begin{aligned} unitaria_S &= \{S, A, B\} \\ unitaria_A &= \{A\} \\ unitaria_B &= \{B, S, A\} \end{aligned}$$

Agregamos las producciones de acuerdo al Paso 2 y eliminamos las producciones unitarias como dice el Paso 3. Llegamos a la gramática siguiente sin producciones unitarias:

$$\begin{aligned} S &\longrightarrow Sa \mid a \mid b \\ A &\longrightarrow Sa \mid a \\ B &\longrightarrow Sa \mid a \mid b \end{aligned}$$

Observe que las variables A y B se volvieron inútiles al no poder ser alcanzables.

Ejemplo 2.8 Elimine las producciones unitarias de la gramática G que aparece a continuación:

$$\begin{aligned} S &\longrightarrow XYZ \\ X &\longrightarrow aY \mid Z \mid b \\ Y &\longrightarrow bX \mid aZ \\ Z &\longrightarrow aa \mid bY \mid Y \end{aligned}$$

Como en G no hay producciones ε, hallamos primero los conjuntos $unitaria_A$, para toda variable A de la gramática. En tal conjunto ponemos a A y a todas las variables a las que se llega desde A siguiendo producciones unitarias. Entonces,

$$\begin{aligned} unitaria_S &= \{S\}, \\ unitaria_X &= \{X, Z, Y\}, \text{ pues tenemos la producción } X \longrightarrow Z, \\ & \qquad\qquad\qquad \text{y al seguirla vemos que } Z \longrightarrow Y. \\ unitaria_Y &= \{Y\}, \\ unitaria_Z &= \{Z, Y\}. \end{aligned}$$

Las producciones de S y de Y se mantienen igual. Para las producciones de X agregamos las producciones **no** unitarias de Y y Z. Para las producciones de Z agregamos las producciones no unitarias de Y.

$$\begin{aligned} S &\longrightarrow XYZ \\ X &\longrightarrow aY \mid Z \mid b \mid bX \mid aZ \mid aa \mid bY \\ Y &\longrightarrow bX \mid aZ \\ Z &\longrightarrow aa \mid bY \mid Y \mid bX \mid aZ \end{aligned}$$

Quitamos las producciones unitarias. **Atención**: no se eliminan variables, sólo producciones. Quedando las producciones como:

$$\begin{array}{rcl} S & \longrightarrow & XYZ \\ X & \longrightarrow & aY \,|\, b \,|\, bX \,|\, aZ \,|\, aa \,|\, bY \\ Y & \longrightarrow & bX \,|\, aZ \\ Z & \longrightarrow & aa \,|\, bY \,|\, bX \,|\, aZ \end{array}$$

La gramática anterior ya no tiene producciones unitarias.

¿Importa el orden al eliminar variables inútiles, producciones-ε y producciones unitarias? Notemos que:

- Al eliminar variables inútiles se quitan elementos pero no se pueden agregar producciones-ε ó producciones unitarias.
- Al eliminar producciones-ε se podrían agregar producciones unitarias.
- El eliminar producciones unitarias requiere que se eliminen producciones-ε primero, y se podrían generar variables inútiles.

Por lo que el orden de eliminación es el siguiente:

A Producciones-ε.

B Producciones unitarias (no agrega producciones-ε).

C Variables inútiles (no agrega ninguna producción).

2.4. Forma Normal de Chomsky

Definición 2.9 Una GLC G está en Forma Normal de Chomsky (FNC) si todas sus producciones son de la forma:

- $A \longrightarrow a$,
- $A \longrightarrow XY$,

donde $A, X, Y \in V$ y $a \in T$.

La FNC tiene aplicaciones importantes, como son: en el algoritmo CKY, en la demostración del lema del bombeo para LLC, en encontrar la forma normal de Greibach, y una gramática en FNC puede derivar una cadena de tamaño n en exactamente $2n-1$ pasos, como lo indica la siguiente proposición.

Proposición 2.10 Sea $G = (V, T, P, S)$ una gramática en FNC, y $w \neq \varepsilon$ una cadena en $\mathcal{L}(G)$, con $|w| = n$. Entonces $S \Rightarrow^* w$ en exactamente $2n-1$ pasos.

Demostración. Como las producciones $A \longrightarrow XY$ incrementan en uno la cadena que se va derivando, entonces necesitamos aplicar tales producciones $n-1$ veces iniciando en S para generar n variables, a las cuales se les tiene que aplicar la producción $A \longrightarrow a$ para convertirlas a símbolos terminales, y así obtener w. En total, en número de aplicaciones de producciones en G para derivar w es $(n-1)+n = 2n-1$. $\dashv$

Teorema 2.11 *Dada una GLC G_1 con $\varepsilon \notin \mathcal{L}(G_1)$, existe una gramática en FNC G_2 tal que $\mathcal{L}(G_1) = \mathcal{L}(G_2)$.*

Demostración. Construimos la gramática G_2 en tres pasos.

Paso 1. Limpiar la gramática siguiendo este orden: (i) eliminamos las producciones-ε, (ii) eliminamos producciones unitarias y (iii) eliminamos las variables inútiles. Ahora, todas las producciones son de la forma:

- $A \longrightarrow a$,
- $A \longrightarrow X_1X_2 \dots X_k, \quad k \geq 2$,

con $A, X_1, X_2, \dots, X_k \in V \cup T$ y $a \in T$.

Paso 2. Eliminar el lado derecho mixto. Este paso tiene que ver con terminales presentes en las producciones. Para cada $a \in T$, agregamos variables nuevas V_a y producciones nuevas $V_a \longrightarrow a$. En cada producción $A \longrightarrow X_1X_2 \dots X_k$ reemplazamos a por V_a si $k \geq 2$. Ahora, todas las producciones son de la forma:

- $A \longrightarrow a$,
- $A \longrightarrow A_1A_2 \dots A_k, \quad k \geq 2$,

con $A, A_1, A_2, \dots, A_k \in V$ y $a \in T$.

Paso 3. Factorizar producciones largas. Para $A \longrightarrow A_1A_2 \dots A_k$ con $k \geq 3$, agregamos nuevas variables $B_1, B_2, \dots, B_{k-2}$ y reemplazamos $A \longrightarrow A_1A_2 \dots A_k$ por

$$\begin{array}{lcl} A & \longrightarrow & A_1B_1 \\ B_1 & \longrightarrow & A_2B_2 \\ B_2 & \longrightarrow & A_3B_3 \\ & \vdots & \\ B_{k-2} & \longrightarrow & A_{k-1}A_k \end{array}$$

Así obtenemos una GLC en Forma Normal de Chomsky que preserva el lenguaje de la gramática original. $\dashv$

Ejemplo 2.12 Encuentre la forma normal de Chomsky de la siguiente gramática:

$$\begin{aligned} G : S &\longrightarrow ABB \mid ab \\ A &\longrightarrow Ba \mid ba \\ B &\longrightarrow aAbS \end{aligned}$$

Paso 1 La gramática anterior está libre de producciones-ε, producciones unitarias y variables inútiles.

Paso 2 Agregamos las variables V_a y V_b, y sus producciones respectivas. También reemplazamos V_a y V_b en las producciones de la gramática G. Obtenemos:

$$\begin{aligned}
S &\longrightarrow ABB \mid V_aV_b \\
A &\longrightarrow BV_a \mid V_bV_a \\
B &\longrightarrow V_aAV_bS \\
V_a &\longrightarrow a \\
V_b &\longrightarrow b
\end{aligned}$$

Paso 3 Para factorizar las producciones largas necesitamos de las variables nuevas X_1, Y_1 y Y_2, que se ocupan de la siguiente manera.

$$\begin{aligned}
S &\longrightarrow AX_1 \mid V_aV_b & \qquad Y_1 &\longrightarrow AY_2 \\
X_1 &\longrightarrow BB & Y_2 &\longrightarrow V_bS \\
A &\longrightarrow BV_a \mid V_bV_a & V_a &\longrightarrow a \\
B &\longrightarrow V_aY_1 & V_b &\longrightarrow b
\end{aligned}$$

Esta gramática ya está en Forma Normal de Chomsky.