

Gramáticas independientes de contexto

Autómatas y lenguajes formales

4 de octubre de 2023

Diseño de gramáticas

Para enfrentar el problema de diseñar una gramática a partir de ciertas condiciones van unos tips:

1. Puedes partir de pedazos más sencillos y luego unirlos, por ejemplo: $\{w \in \{a, b, c\}^* \mid w \text{ contenga } a \text{ un número par de veces}\} \cup \{w \in \{a, b, c\}^* \mid w \in b^n c^m, n \geq 0, m = 2n\}$. Tenemos para el primer (S_1) y segundo (S_2) caso:

$$S_1 \rightarrow aS_1a|\epsilon$$

$$S_2 \rightarrow bS_2cc|\epsilon$$

Uniéndolos para el caso dado al inicio:

$$S \rightarrow S_1|S_2$$

$$S_1 \rightarrow aS_1a|\epsilon$$

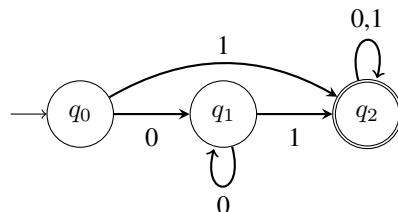
$$S_2 \rightarrow bS_2cc|\epsilon$$

2. Si se quiere pasar de un autómata finito determinista (DFA) a un lenguaje libre de contexto, se parte del DFA y se construye la gramática. Para cada estado del autómata se asigna una variable ($q_i \Rightarrow R_i$), o poniéndolo en el lenguaje apropiado:

$$R_i \rightarrow aR_j \Rightarrow \exists \delta(q_i, a) = q_j$$

$$R_i \rightarrow \epsilon \Rightarrow q_i \in F \text{ (estado final),}$$

y R_0 es la variable inicial, que corresponde al estado inicial q_0 . Por ejemplo, si tenemos el DFA:



Que se traduciría a la gramática

$$\begin{aligned}
R_0 &\rightarrow 0R_1|1R_2 \\
R_1 &\rightarrow 0R_1|1R_2 \\
R_2 &\rightarrow 1R_2|0R_2|\epsilon.
\end{aligned}$$

3. Para llevar la cuenta de los símbolos siempre es útil la producción $R \rightarrow uRv$. No doy ejemplo pues lo dejo para que ustedes lo apliquen.

1. Formas normales

Si me permiten me saltaré la parte de paréntesis balanceados, no dejen de revisarlo por su cuenta. Me pasaré directo a las formas normales.

1.1. Forma normal de Chomsky (CNF)

En la forma normal de Chomsky un lenguaje libre de contexto se escribe de la forma:

$$\begin{aligned}
A &\rightarrow BC \\
A &\rightarrow a,
\end{aligned}$$

donde A, B, C son variables y $a \in \Sigma$ (un símbolo del alfabeto). Es decir que el CFL está en forma normal de Chomsky siempre que del lado derecho de las producciones haya sólo un par de variables distintas a la inicial o sólo un símbolo terminal, incluido ϵ . Esto hace que sea más sencillo definir la gramática.

Cualquier lenguaje independiente de contexto puede convertirse en forma normal de Chomsky con un análisis detallado:

1. Dar una nueva variable inicial (así nos aseguramos que la variable inicial no aparezca del lado derecho de las producciones).
2. Se eliminan todas las producciones de la forma $A \rightarrow \epsilon$
3. Se eliminan las reglas unitarias de la forma $A \rightarrow B$
4. Ponemos parches en las secciones que dejamos huecas por nuestro frenesí desechador
5. Ya teniendo las reglas, lo acomodamos para que quede en la forma normal de Chomsky

Como siempre, un ejemplo. Tenemos el lenguaje libre de contexto dado por las variables A, B sobre el alfabeto $\Sigma = \{0\}$:

$$\begin{aligned}
A &\rightarrow BAB|B|\epsilon \\
B &\rightarrow 00|\epsilon
\end{aligned}$$

Lo convertimos a la forma normal de Chomsky siguiendo nuestros pasos descritos, primero damos una nueva variable inicial que llamaremos S :

$$\begin{aligned}
S &\rightarrow A \\
A &\rightarrow BAB|B|\epsilon \\
B &\rightarrow 00|\epsilon
\end{aligned}$$

El siguiente paso es quitar todas las producciones que tienen del lado derecho ϵ , para hacer las cosas bien de una vez vamos parchando los desperfectos que provocamos, cualquier variable que en una producción lleve a ϵ en sus apariciones la cambiamos por la ausencia de cadena, es decir, si hay una producción del tipo $A \rightarrow BAB|AA|a|\epsilon$ entonces se quedan las producciones que no llevan a ϵ , y a continuación se escriben las producciones en la que la variable es sustituida por la cadena vacía: $A \rightarrow BAB|AA|a|\mathbf{B}\mathbf{B}|\mathbf{A}$ (las últimas dos producciones son las primeras dos cambiando A por la cadena vacía, en el caso de AA tendría que sustituirse cada A , pero como llevan a lo mismo basta con poner una).

De manera similar, para nuestro ejemplo:

$$\begin{aligned}
S &\rightarrow A \\
A &\rightarrow BAB|B|BB|AB|BA|A \\
B &\rightarrow 00
\end{aligned}$$

Ahora eliminamos las reglas unitarias, como en el caso de la eliminación de ϵ hay que estar un paso adelante, si $S \rightarrow A$, nos saltamos a donde va A , entonces queda $S \rightarrow BAB|B|BB|AB|BA|A$, ya nos comimos una producción unitaria, pero faltan ver $A \rightarrow B$ ($A \rightarrow A$ es la producción trivial, A siempre va a A la podemos borrar incluso del CFL y no perdemos nada, la borramos sin más), así podemos escribir:

$$\begin{aligned}
S &\rightarrow BAB|B|BB|AB|BA \\
A &\rightarrow BAB|B|BB|AB|BA \\
B &\rightarrow 00,
\end{aligned}$$

luego

$$\begin{aligned}
S &\rightarrow BAB|00|BB|AB|BA \\
A &\rightarrow BAB|00|BB|AB|BA \\
B &\rightarrow 00
\end{aligned}$$

Este caso es un poco más sencillo, pues no aparece $B \rightarrow A$, de aparecer tendríamos que sustituir esa A por toda la producción del segundo renglón, ténganlo en cuenta.

El último paso es una manita de gato para ponerlo de forma que sólo haya dos variables o un terminal del lado derecho de la producción. Cuando haya tres variables, agrupamos dos en una nueva y damos la regla de producción correspondiente, cuando haya dos terminales agregamos una variable que lleve al par de variables que van a cada símbolo terminal, entonces:

$$\begin{aligned}
S &\rightarrow \mathbf{U}B|B_1B_1|BB|AB|BA \\
A &\rightarrow \mathbf{U}B|B_1B_1|BB|AB|BA \\
\mathbf{U} &\rightarrow BA \\
B &\rightarrow B_1B_1 \\
B_1 &\rightarrow 0
\end{aligned}$$

Noten que en las reglas de producción de S y A van a la par de variables a las que lleva U , pero no nos conviene cambiar esa regla por $A|S \rightarrow U$ ya que no es forma normal de Chomsky.

Y ya, esa última es nuestra forma normal de Chomsky del lenguaje libre de contexto dado, cumple los requisitos. Reciban un caluroso saludo de Chomsky por ese logro en la figura 1.

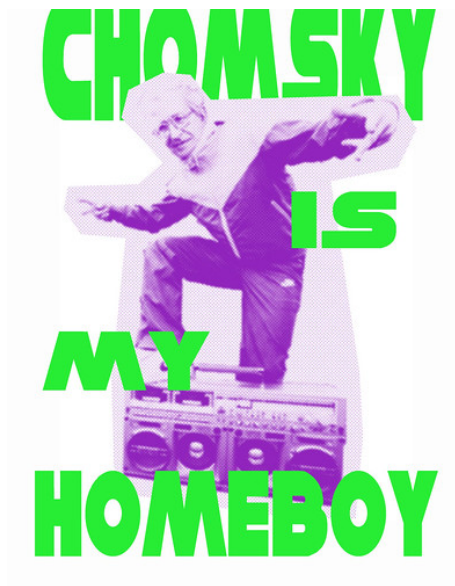


Figura 1: "chomskiezzz3" por xantiblingx tiene una licencia CC BY-NC 2.0

1.2. Forma Normal de Greibach (GNF)

En la forma normal de Greibach las producciones son de la forma:

$$A \rightarrow aB_1B_2\dots B_k,$$

para alguna $k > 0$, B_1, B_2, \dots, B_k son variables y $a \in \Sigma$, algún símbolo del alfabeto.

Para llegar a la forma normal de Greibach seguimos los pasos:

1. Pasar el lenguaje independiente de contexto a la forma normal de Chomsky.

2. Para cada variable A y a un símbolo del alfabeto se tiene un nuevo lenguaje:

$$R_{A,a} = \{\beta \in \Gamma^* \mid A \xrightarrow[G]{L} a\beta\},$$

que quiere decir que la regla de producción para la variable A puede obtenerse con producciones solo aplicadas al símbolo de la extrema izquierda de la forma enunciada. Si lo vemos de la forma que hemos enunciado las gramáticas independientes de contexto, partiendo de una variable inicial, que va a combinaciones de variables, y esas variables eventualmente van a símbolos terminales, es un poco caminar para atrás (normalmente armamos los términos de izquierda a derecha, para la forma normal de Greibach los armamos de derecha a izquierda).

Un método más detallado puede encontrarse en el libro de Shyamalendu[3], libro que recomiendo ver con precaución, pues según yo tiene algunos errores (muy probablemente yo soy el del error), pero para este caso me parece un procedimiento más detallado para llegar a la forma normal de Greibach¹.

Antes de empezar necesitamos unos lemas que serán útiles en la conversión.

Lema 1 Si $G = (\Sigma, \Gamma, S, \rightarrow)$ es una gramática independiente de contexto, y si $A \rightarrow A\alpha$ y $A \rightarrow \beta_1|\beta_2|\dots|\beta_n$ pertenecen a las reglas de producción (\rightarrow) de G , entonces una nueva gramática $G'(\Sigma, \Gamma, S, \rightarrow_{G'})$ puede construirse reemplazando $A \rightarrow \beta_1|\beta_2|\dots|\beta_n$ en $A \rightarrow A\alpha$, resultando

$$A \rightarrow \beta_1\alpha|\beta_2\alpha|\dots|\beta_n\alpha$$

Esta producción pertenece a ($\rightarrow_{G'}$) y puede probarse que $L(G) = L(G')$.

Que nos da una manera de pasar de una producción con puros símbolos terminales aislados a una con parejas de símbolos terminales.

Otro lema útil

Lema 2 Sea $G = (\Sigma, \Gamma, S, \rightarrow)$ es una gramática independiente de contexto y el conjunto de producciones A que pertenecen a (\rightarrow) sean $A \rightarrow A\alpha_1|A\alpha_2|\dots|A\alpha_m|\beta_1|\beta_2|\dots|\beta_m$. Si se introduce una nueva no terminal X . Sea $G' = (\Sigma, \Gamma', S, \rightarrow_{G'})$, donde $\Gamma' = \Gamma \cup X$ y ($\rightarrow_{G'}$) puede construirse reemplazando las producciones A por

1.

$$\begin{aligned} A &\rightarrow \beta_i \quad 1 \leq i \leq n \\ A &\rightarrow \beta_i X \end{aligned}$$

2.

$$\begin{aligned} X &\rightarrow \alpha_j \quad 1 \leq j \leq m \\ X &\rightarrow \alpha_j X \end{aligned}$$

Se puede probar que $L(G) = L(G')$.

Con estos se pueden dar pasos para pasar a la forma normal de Greibach:

¹Por cierto, la forma normal de Greibach es derivada del trabajo de Sheila Greibach, computóloga estadounidense que trabaja en la UCLA, vale la pena reconocer el trabajo de las computólogas.

1. Pasar la gramática a forma normal de Chomsky de acuerdo a las instrucciones líneas arriba.
2. Renombrar los no terminales V_N como (A_1, A_2, \dots, A_n) con el símbolo inicial A_1 .
3. Usando el ?? modificar las producciones para que todas queden de la forma $A_i \rightarrow A_j V$, con $i \leq j$.
4. Aplicando de manera repetida ambos lemas todas las producciones se modificarán hasta quedar en la forma normal de Greibach.

Recordemos como quedó nuestra Gramática en forma normal de Chomsky:

$$\begin{aligned}
 S &\rightarrow \mathbf{U}B|B_1B_1|BB|AB|BA \\
 A &\rightarrow \mathbf{U}B|B_1B_1|BB|AB|BA \\
 \mathbf{U} &\rightarrow BA \\
 B &\rightarrow B_1B_1 \\
 B_1 &\rightarrow 0
 \end{aligned}$$

Ya está en forma normal de Chomsky por lo que pasamos al paso 2 renombrar en orden:

$$\begin{aligned}
 A_1 &\rightarrow A_3A_4|A_5A_5|A_4A_4|A_2A_4|A_4A_2 \\
 A_2 &\rightarrow A_3A_4|A_5A_5|A_4A_4|\mathbf{A}_2\mathbf{A}_4|A_4A_2 \\
 A_3 &\rightarrow A_4A_2 \\
 A_4 &\rightarrow A_5A_5 \\
 A_5 &\rightarrow 0
 \end{aligned}$$

Por suerte aquí cumple el orden de que $A_i \rightarrow A_j V$ con $i \leq j$, no hay que renombrar mucho más, pero notemos que en la producción A_2 aparece la producción $A_2 \rightarrow A_2A_4$ que cumple estar en la forma $A \rightarrow A\alpha$ mientras las demás cumplen $A \rightarrow \beta_1|\beta_2|\dots|\beta_n$. De acuerdo a 1 se sustituye por $A \rightarrow \beta_1\alpha|\beta_2\alpha|\dots|\beta_n\alpha$, es decir:

$$\begin{aligned}
 A_1 &\rightarrow A_3A_4|A_5A_5|A_4A_4|A_2A_4|A_4A_2 \\
 A_2 &\rightarrow A_3A_4|A_5A_5|A_4A_4|\mathbf{A}_3\mathbf{A}_4\mathbf{A}_4|\mathbf{A}_5\mathbf{A}_5\mathbf{A}_4|\mathbf{A}_4\mathbf{A}_4\mathbf{A}_4|\mathbf{A}_4\mathbf{A}_2\mathbf{A}_4|A_4A_2 \\
 A_3 &\rightarrow A_4A_2 \\
 A_4 &\rightarrow A_5A_5 \\
 A_5 &\rightarrow 0
 \end{aligned}$$

Lo que sigue es sustituir los primeros terminales del lado derecho de cada producción hasta que el primer símbolo sea un terminal:

$$\begin{aligned}
A_1 &\rightarrow A_4 A_2 A_4 | 0 A_5 | A_5 A_5 A_4 | A_3 A_4 A_4 | A_5 A_5 A_4 | A_4 A_4 A_4 | \\
&\quad | A_3 A_4 A_4 A_4 | A_5 A_5 A_4 A_4 | A_4 A_4 A_4 A_4 | A_4 A_2 A_4 A_4 | A_4 A_2 A_4 | A_5 A_5 A_2 \\
A_2 &\rightarrow A_4 A_2 A_4 | 0 A_5 | A_5 A_5 A_4 | A_4 A_2 A_4 A_4 | 0 A_5 A_4 | A_5 A_5 A_4 A_4 | \\
&\quad A_5 A_5 A_2 A_4 | A_5 A_5 A_2 \\
A_3 &\rightarrow A_5 A_5 A_2 \\
A_4 &\rightarrow 0 A_5 \\
A_5 &\rightarrow 0
\end{aligned}$$

El siguiente paso:

$$\begin{aligned}
A_1 &\rightarrow A_5 A_5 A_2 A_4 | 0 A_5 | 0 A_5 A_4 | A_5 A_5 A_2 A_4 A_4 | 0 A_5 A_4 | 0 A_5 A_4 A_4 | \\
&\quad | A_5 A_5 A_2 A_4 A_4 A_4 | 0 A_5 A_4 A_4 | 0 A_5 A_4 A_4 A_4 | 0 A_5 A_2 A_4 A_4 | 0 A_5 A_2 A_4 | 0 A_5 A_2 \\
A_2 &\rightarrow 0 A_5 A_2 A_4 | 0 A_5 | 0 A_5 A_4 | 0 A_5 A_2 A_4 A_4 | 0 A_5 A_4 | 0 A_5 A_4 A_4 | \\
&\quad 0 A_5 A_2 A_4 | 0 A_5 A_2 \\
A_3 &\rightarrow 0 A_5 A_2 \\
A_4 &\rightarrow 0 A_5 \\
A_5 &\rightarrow 0
\end{aligned}$$

Por último:

$$\begin{aligned}
A_1 &\rightarrow 0 A_5 A_2 A_4 | 0 A_5 | 0 A_5 A_4 | 0 A_5 A_2 A_4 A_4 | 0 A_5 A_4 | 0 A_5 A_4 A_4 | \\
&\quad | 0 A_5 A_2 A_4 A_4 A_4 | 0 A_5 A_4 A_4 | 0 A_5 A_4 A_4 A_4 | 0 A_5 A_2 A_4 A_4 | 0 A_5 A_2 A_4 | 0 A_5 A_2 \\
A_2 &\rightarrow 0 A_5 A_2 A_4 | 0 A_5 | 0 A_5 A_4 | 0 A_5 A_2 A_4 A_4 | 0 A_5 A_4 | 0 A_5 A_4 A_4 | \\
&\quad 0 A_5 A_2 A_4 | 0 A_5 A_2 \\
A_3 &\rightarrow 0 A_5 A_2 \\
A_4 &\rightarrow 0 A_5 \\
A_5 &\rightarrow 0
\end{aligned}$$

Y listo, está en forma normal de Greibach.

1.3. Extra: Chomsky y su disco punk

Esto no tiene que ver con el tema, es un dato curioso para que no les sea tan aburrida la lectura. Noam Chomsky es muy conocido por ser, quizá, el intelectual estado unidense más famoso en el mundo, de una ideología cercana al anarquismo es un fuerte crítico al imperialismo de su país, un defensor a ultranza de la libertad de expresión y un pacifista (seguro a estas alturas ustedes saben que aunque muchos medios y persona suelen emparentar al anarquismo con la violencia, no necesariamente van pegadas).

Ha visitado México para hablar de su trabajo en política, sus libros al respecto son famosos, y por ser un intelectual anarquista suele ser muy conocido en el ambiente punk y anarcopunk (aunque tiene sus detractores también ahí).

En 1990 la banda punk californiana Bad Religion le pidió hacer un *split* con ellos. Un *split* es una forma común de sacar discos en el ambiente punk, dos bandas graban

un mismo disco. Cuando eran acetatos, una banda de un lado, la otra del otro, así podías descubrir dos bandas en un mismo disco. La tradición siguió al casete, al cd y ahora al disco digital.

Bad Religion es una banda aún en activo, su vocalista Greg Graffin tiene un doctorado en paleontología evolutiva y es académico en la UCLA. Aprovecha las vacaciones académicas para seguir presentándose con la banda.

Pueden oír el *split* en la siguiente liga de youtube: <https://www.youtube.com/watch?v=-GLIdS0jZrw>.

Referencias

- [1] Kozen, Dexter C. “Automata and Computability” Springer (1997)
- [2] Sipser, Michael “Introduction to the Theory of Computation” 2a ed., Thomson Course Technology (2006)
- [3] Shyamalendu, Kandar. “Introduction to Automata Theory, Formal Languages and Computation”, Dorling Kindersley (India) - Pearson (2013)