

AUTÓMATAS Y LENGUAJES FORMALES LENGUAJES INDEPENDIENTES DEL CONTEXTO Y AUTÓMATAS DE PILA

Francisco Hernández Quiroz

Departamento de Matemáticas
Facultad de Ciencias, UNAM
E-mail: fhq@ciencias.unam.mx
Página Web: www.matematicas.unam.mx/fhq

Posgrado en Ciencia e Ingeniería de la Computación

Gramáticas independientes del contexto

- Una gramática $G = \langle \Sigma, \Gamma, S, \rightarrow \rangle$ es *independiente del contexto* sii todas las reglas de producción tienen una de las dos siguientes formas

$$A \rightarrow \alpha \quad A \rightarrow \epsilon,$$

donde $A \in \Gamma$.

- CFG designará el conjunto de gramáticas independientes del contexto.
- El lenguaje generado por G se denotará con $L(G)$.
- Un lenguaje L es *independiente del contexto* sii existe una gramática $G \in \text{CFG}$ tal que

$$L = L(G).$$

- CFL es el conjunto de lenguajes independientes del contexto.

Ejemplos

- Lenguaje de palindromas. Sea $G_P = \langle \{a, b\}, \{S\}, S, \rightarrow_P \rangle$, donde

$$S \rightarrow_P aSa \mid bSb \mid \epsilon.$$

- Lenguaje de paréntesis. Sea $G_D = \langle \{(,)\}, \{S\}, S, \rightarrow_D \rangle$, con las siguientes reglas

$$S \rightarrow_D (S) \mid SS \mid \epsilon.$$

Otras definiciones I

Considérese una derivación

$$\alpha_0 \Rightarrow \alpha_1 \Rightarrow \dots \Rightarrow \alpha_n.$$

Esta derivación presupone reglas de producción

$$A_1 \rightarrow \beta_1 \dots A_n \rightarrow \beta_n$$

tales que

$$\alpha_i = \gamma_i A_i \eta_i \quad \alpha_{i+1} = \gamma_i \beta_i \eta_i.$$

Esta derivación será *por la izquierda* sii para todo $i \leq n$

$$\gamma_i \leq \gamma_{i+1} \quad \gamma_i \in \Sigma^*$$

Otras definiciones II

y será por la derecha sii

$$\eta_i \leq \eta_{i+1} \quad \eta_i \in \Sigma^*.$$

Una gramática es *no ambigua* sii $\forall \alpha \in L(G)$, α tiene exactamente una derivación por la izquierda (o por la derecha).

Un lenguaje es *no ambiguo* sii existe una gramática no ambigua que lo genere. De lo contrario, será *inherentemente ambiguo*.

Una gramática es *pulcra* sii

(a) $\forall A \in \Gamma . L(A) \neq \emptyset$;

(b) $\forall A \in \Gamma . \exists \alpha, \beta \in \Sigma^* . S \rightarrow \alpha A \beta$ o $S = A$.

Otras definiciones III

Una producción es *terminal* sii no contiene símbolos no terminales del lado derecho.

Una producción es una regla- ϵ sii su lado derecho es ϵ .

Una producción es *unitaria* sii es de la forma de la forma $A \rightarrow B$, con $A, B \in \Gamma$.

Una gramática es *propia* sii no contiene regla- ϵ ni reglas unitarias.

Paréntesis balanceados

La gramática G_D genera el *lenguaje de paréntesis balanceados*, es decir, $\alpha \in L(G_D)$ sii

① $A(\alpha) = C(\alpha)$;

② $C(\beta) \leq A(\beta) \quad \forall \beta . \exists \gamma . \alpha = \beta \gamma$;

donde $A(\alpha)$ y $C(\alpha)$ son el número de paréntesis que abren y cierran que aparecen en α , respectivamente.

Demostración. Por inducción en la derivación de $S \Rightarrow \alpha$.

Lenguajes de Dyck

El lenguaje de paréntesis se puede generalizar a los lenguajes de Dyck. Sea A un alfabeto de símbolos no terminales y sea \bar{A} una copia de A , es decir:

(i) $A \cap \bar{A} = \emptyset$

(ii) existe una biyección de A a \bar{A} .

Por ejemplo, si $A = \{ \{, \} \}$, entonces $\bar{A} = \{ \} \}$.

El lenguaje de Dyck sobre A es el lenguaje generado por las reglas de producción:

$$S \rightarrow a_1 S \bar{a}_1 \mid \dots \mid a_n S \bar{a}_n \mid SS \mid \epsilon,$$

donde $A = \{ a_1, \dots, a_n \}$ y $\bar{a}_i \in \bar{A}$ es la copia de a_i . Este lenguaje se denotará con D_A^* .

Si $|A| = n$, una notación alternativa es D_n^* .

Formas normales de Chomsky y de Greibach

Sea $G = \langle \Sigma, \Gamma, S, \rightarrow \rangle$ un gramática. G tiene *forma normal de Chomsky* (CNF) sii todas las producciones tiene la forma

$$A \rightarrow BC \quad A \rightarrow a,$$

donde $A, B, C \in \Gamma$ y $a \in \Sigma$.

En cambio, G tiene *forma normal de Greibach* (GNF) sii todas las producciones tienen la forma

$$A \rightarrow bB_1 \dots B_k$$

con $0 \leq k, b \in \Sigma$ y $B_1, \dots, B_k \in \Gamma$.

Ejemplos

Los conjuntos de producciones siguientes tienen forma normal de Chomsky y de Greibach (respectivamente)

$$\begin{aligned} S &\rightarrow AB \mid AC \mid SS & C &\rightarrow SB & A &\rightarrow (\quad B \rightarrow) \\ S &\rightarrow (B \mid (SB \mid (BS \mid (SBS & B &\rightarrow) \end{aligned}$$

y generan el lenguaje de paréntesis balanceados.

Las producciones siguientes generan el lenguaje de palindromas

$$\begin{aligned} S &\rightarrow AA \mid BB \mid AC \mid BD & C &\rightarrow SA & D &\rightarrow SB & A &\rightarrow a & B &\rightarrow b \\ S &\rightarrow aA \mid aSA \mid bB \mid bSB & A &\rightarrow a & B &\rightarrow b \end{aligned}$$

Con excepción en ambos casos de ϵ .

Todas las CFG tienen equivalente en CNF o GNF

Teorema. Sea $G \in \text{CFG}$. Entonces existen $G_C, G_G \in \text{CFG}$ tales que tienen forma normal de Chomsky y de Greibach, respectivamente, y

$$L(G_C) = L(G_G) = L(G) - \{\epsilon\}.$$

Lema de gramáticas propias I

Sea $G \in \text{CFG}$. Entonces, existe una gramática propia G' tal que

$$L(G') = L(G) - \{\epsilon\}.$$

Demostración. Sea \rightarrow' la relación mínima que

- (i) contiene a \rightarrow ;
- (ii) si $A \rightarrow' \alpha B \beta$ y $B \rightarrow \epsilon$, entonces $A \rightarrow' \alpha \beta$
- (iii) si $A \rightarrow' B$ y $B \rightarrow' \gamma$, entonces $A \rightarrow' \gamma$.

Lema de gramáticas propias II

Y sea $G' \in \text{CFG}$ como G , pero con las reglas de producción \rightarrow' . Entonces

$$L(G) = L(G')$$

Obviamente, $L(G) \subseteq L(G')$. Para $L(G') \subseteq L(G)$, basta observar que toda derivación en G se puede realizar en G' , salvo tal vez en dos pasos.

Sea ahora G'' como G' pero con la relación \rightarrow'' que prescinde de las producciones unitarias y $-\epsilon$ que existan en \rightarrow' . Entonces

$$L(G'') = L(G') - \{\epsilon\} = L(G) - \{\epsilon\}.$$

Esta última afirmación se demuestra considerando que las derivaciones de longitud mínima en G' prescinden de producciones- ϵ y unitarias.

Transformación a forma normal de Chomsky

Una vez que contamos con una gramática propia G'' , podemos construir la FNC:

- 1 Para cada $a \in \Sigma$, introducimos nuevos no terminales A_a y reglas de producción $A_a \rightarrow a$.
- 2 Sustituimos los terminales por los no terminales del punto anterior en todas las reglas de G'' que no tengan FNC todavía.
- 3 Dada una regla $A \rightarrow B_1 \dots B_k$ ($k > 2$), introducimos un nuevo terminal C_1 y las reglas

$$A \rightarrow B_1 C_1 \quad C_1 \rightarrow B_2 \dots B_k.$$

- 4 Repetimos este procedimiento hasta tener sólo reglas con dos no terminales del lado derecho.
- 5 Eliminamos las reglas del punto anterior que no tengan FNC.

Transformación a forma normal de Greibach

Sea G_C una gramática en FNC. Para convertirla a FNG:

- 1 Para cada $X \in \Gamma$, considérese el lenguaje

$$R_{X,x} = \{\beta \in \Gamma \mid X \Rightarrow_L x\beta\}.$$

Este lenguaje es regular. Sea $G_{X,x}$ una gramática lineal por la derecha que lo genere. Sea $S_{X,x}$ el símbolo inicial de esta gramática.

- 2 Entonces, sustituimos las reglas

$$A \rightarrow XY$$

por reglas

$$A \rightarrow xS_{X,x}Y,$$

y agregamos las producciones y símbolos no terminales de $G_{X,x}$ a nuestra gramática.

- 3 Eliminamos las reglas- ϵ .

Ejemplos I

Lenguaje de paréntesis. Empezamos con forma normal de Chomsky:

$$S \rightarrow AB \mid AC \mid SS \quad C \rightarrow SB \quad A \rightarrow (\quad B \rightarrow) .$$

Tenemos ahora los siguientes lenguajes $R_{X,x}$

$$R_{S,(} = (B + C)S^*$$

$$R_{C,(} = (B + C)S^*B$$

$$R_{A,(} = \{\epsilon\}$$

$$R_{B,)} = \{\epsilon\}$$

Estas producciones nos generan los lenguajes anteriores:

$$S_{S,(} \rightarrow BX \mid CX \quad X \rightarrow SX \mid \epsilon$$

$$S_{C,(} \rightarrow BY \mid CY \quad Y \rightarrow SY \mid BZ \quad Z \rightarrow \epsilon$$

$$S_{A,(} \rightarrow \epsilon$$

$$S_{B,)} \rightarrow \epsilon$$

Ejemplos II

Entonces, tenemos las reglas nuevas

$$\begin{array}{lll}
 S \rightarrow (S_A(B \mid (S_A(C \mid (S_S(S & C \rightarrow (S_S(B & A \rightarrow (\\
 S_S,(\rightarrow)S_{B,})X \mid (S_{C,}(X & X \rightarrow (S_S(X \mid \epsilon & B \rightarrow) \\
 S_{C,}(\rightarrow)S_{B,})Y \mid (S_{C,}(Y & Y \rightarrow (S_S(Y \mid)S_{B,})Z & Z \rightarrow \epsilon \\
 S_{A,}(\rightarrow \epsilon & S_{B,}) \rightarrow \epsilon &
 \end{array}$$

Teorema del bombeo II

- Este no terminal puede usarse para definir las cadenas γ y θ .

Teorema del bombeo I

Sea $L \in CFL$. Entonces, existe $k \in \mathbb{N}$ tal que para toda $\alpha \in L$, si $k \leq |\alpha|$, entonces

$$\alpha = \beta\gamma\eta\theta\phi \quad \text{y} \quad \gamma\theta \neq \epsilon$$

y para toda $i \in \mathbb{N}$

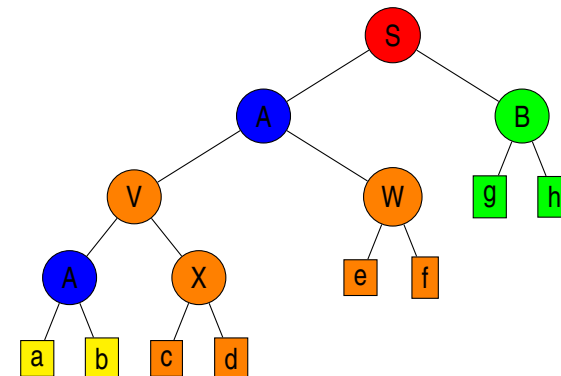
$$\beta\gamma^i\eta\theta^i\phi \in L.$$

Demostración.

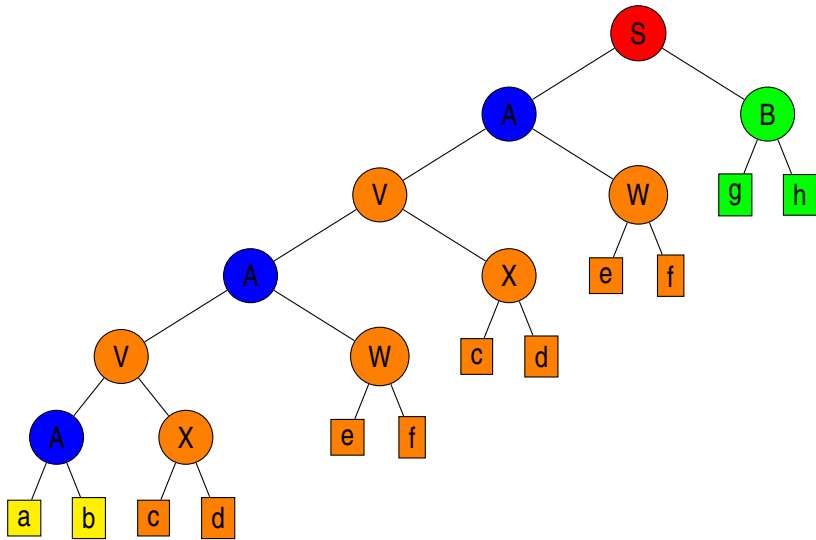
- Sea $G \in CFG$ en FNC y $L(G) = L - \epsilon$.
- Sea Γ el alfabeto de símbolos no terminales de G , con $|\Gamma| = n$ y sea $k = 2^{n+1}$.
- Entonces, el árbol sintáctico de toda cadena de longitud igual o superior a k debe tener profundidad de $n + 1$ al menos.
- Entonces en un camino de longitud máxima de la raíz a las hojas debe aparecer dos veces, al menos, el mismo no terminal.

Ejemplo

Considérense los siguientes árboles de derivación:



Árbol de la cadena $\beta\gamma\eta\theta\phi = abcdefgh$



Árbol de la cadena $\beta\gamma^2\eta\theta^2\phi = abcdefcdefgh$

Aplicaciones

Como con los lenguajes regulares, este teorema se puede utilizar para demostrar que un lenguaje dado no es independiente del contexto. Ejemplos:

$$\{a^n b^n c^n\},$$

por una aplicación directa del teorema, y

$$\{\alpha\alpha\}$$

pues los CFL y los regulares son cerrados bajo intersección y

$$\{a^n b^m a^n b^m\} = \{\alpha\alpha\} \cap L(a^* b^* a^* b^*)$$

y, obviamente, $\{a^n b^m a^n b^m\}$ no es independiente del contexto

Autómatas de pila

Un autómata de pila no determinista A está formado por

- un conjunto de estados Q ;
- un alfabeto de entrada Σ ;
- un alfabeto de pila Γ ;
- una relación de transición $\delta \subseteq (Q \times (\Sigma \cup \{\epsilon\}) \times \Gamma) \times (Q \times \Gamma^*)$
- un estado inicial s ;
- un símbolo inicial de la pila \perp ;
- un subconjunto de estados finales $F \subseteq Q$.

Llamaremos NPDA a los autómatas de pila no deterministas.

Lenguajes aceptados por NPDA

Una configuración de un autómata $A \in \text{NPDA}$ es una terna (q, α, β) donde

$$q \in Q$$

$$\alpha \in \Sigma^*$$

$$\beta \in \Gamma^*$$

La configuración inicial es (s, α, \perp) . Definiremos ahora una relación entre configuraciones:

$$\text{si } \delta((q, a, B), (r, \eta)) \text{ entonces } (q, a\alpha, B\beta) \Rightarrow (r, \alpha, \eta\beta)$$

$$\text{si } \delta((q, \epsilon, B), (r, \eta)) \text{ entonces } (q, \alpha, B\beta) \Rightarrow (r, \alpha, \eta\beta)$$

El lenguaje aceptado por un $A \in \text{NPDA}$ es

$$L(A) = \{\alpha \in \Sigma^* \mid \exists f \in F. (s, \alpha, \perp) \Rightarrow^* (f, \epsilon, \beta)\}$$

Aceptación por pila vacía

Una definición alternativa del lenguaje aceptado por un $A \in \text{NPDA}$ es

$$L(A) = \{\alpha \in \Sigma^* \mid (s, \alpha, \perp) \Rightarrow^* (q, \epsilon, \epsilon)\}$$

Ambas definiciones son equivalentes y puede construirse un $A' \in \text{NPDA}$ que acepte por pila vacía el mismo lenguaje que un $A \in \text{NPDA}$ que acepte por estado final (y viceversa).

Teorema de equivalencia entre CFG y NPDA I

(a) Sea $G \in \text{CFG}$. Entonces, existe $A \in \text{NPDA}$ tal que

$$L(G) = L(A).$$

(b) Sea $A \in \text{NPDA}$. Entonces, existe $G \in \text{CFG}$ tal que

$$L(G) = L(A).$$

Demostración

(a) Sea $G = \langle \Sigma, \Gamma, S \rightarrow \rangle$ en FNG. Sea

$$A = (\{q\}, \Sigma, \Gamma, \delta, q, S, q),$$

donde

$$\delta((q, a, A), (q, B_1 \cdots B_k)) \quad \text{sii} \quad A \rightarrow aB_1 \cdots B_k.$$

Teorema de equivalencia entre CFG y NPDA II

Si puede demostrar por inducción en la longitud de las derivaciones (por la izquierda) que

$$(q, \alpha\beta, A) \Rightarrow_A^n (q, \beta, \gamma) \quad \text{sii} \quad A \Rightarrow_G^n \alpha\gamma.$$

Entonces

$$\begin{aligned} \alpha \in L(G) & \quad \text{sii} \quad S \Rightarrow_G^* \alpha \\ & \quad \text{sii} \quad (q, \alpha, S) \Rightarrow_A^* (q, \epsilon, \epsilon) \\ & \quad \text{sii} \quad \alpha \in L(A). \end{aligned}$$

(b) Se demuestra primero que para todo $A \in \text{NPDA}$ existe un $A' \in \text{NPDA}$ con un solo estado tal que

$$L(A) = L(A')$$

y una vez construido A' , los argumentos del caso (a) se pueden aplicar en sentido inverso.

Determinismo

- A diferencia de los lenguajes regulares, la clase CFL no determinista es estrictamente mayor que la determinista (DCFL).
- Un autómata determinista de pila $D = (Q, \Sigma, \Gamma, \delta, \perp, \neg, s, F)$ incluye el símbolo especial \neg que sirve para indicar el final de la cadena de entrada.
- Las transiciones están indicadas por la función

$$\delta : Q \times \Sigma \cup \{\epsilon, \neg\} \rightarrow Q \times \Gamma^*.$$

- δ no puede eliminar de la pila a \perp , es decir

$$\delta(q, a, \perp) = (r, \alpha\perp).$$

- La aceptación es por estados finales. La aceptación por pila vacía define un conjunto de lenguajes diferente.
- Un ejemplo de un lenguaje en CFL pero no en DCFL es

$$\{a, b\}^* - \{\alpha\alpha \mid \alpha \in \{a, b\}^*\}$$

El teorema de Chomsky y Schützenberger I

Sea $L \in \text{CFL}$. Entonces existen $n \in \mathbb{N}$, $R \in \text{REG}$ y un homomorfismo h tal que

$$L = h(D_n^* \cap R).$$

Nota: un homomorfismo $h : \Sigma^* \rightarrow \Gamma^*$ es una función tal que

$$h(\alpha\beta) = h(\alpha)h(\beta).$$

Demostración. Sea $G = \langle \Sigma, \Gamma, S, \rightarrow \rangle$ en FNC. Asignaremos letras griegas minúsculas a las producciones de G :

$$\pi, \rho, \sigma, \dots$$

Definiremos nuevas producciones

$$\pi' = \begin{cases} A \rightarrow \frac{1}{\pi} B \frac{1}{\pi} C \frac{2}{\pi} & \text{si } \pi = A \rightarrow BC \\ A \rightarrow \frac{1}{\pi} \frac{1}{\pi} \frac{2}{\pi} \frac{2}{\pi} & \text{si } \pi = A \rightarrow a \end{cases}$$

El teorema de Chomsky y Schützenberger II

y sean

$$\begin{aligned} \Sigma' &= \left\{ \frac{1}{\pi}, \frac{1}{\pi}, \frac{2}{\pi}, \frac{2}{\pi} \mid \pi \in \rightarrow \right\} \\ \rightarrow' &= \{ \pi' \mid \pi \in \rightarrow \} \\ G' &= \langle \Sigma', \Gamma, S, \rightarrow' \rangle \end{aligned}$$

Se puede definir fácilmente un isomorfismo con un lenguaje D_n^* (con un par de tipos de paréntesis por cada regla en \rightarrow). Obviando el isomorfismo, es claro que

$$L(G') \subseteq D_n^*.$$

$L(G')$ cumple además con las siguientes restricciones adicionales:

- 1 Para toda $\pi \in \rightarrow$, el símbolo $\frac{1}{\pi}$ está seguido de $\frac{2}{\pi}$.

El teorema de Chomsky y Schützenberger III

- 2 Ningún $\frac{2}{\pi}$ está seguido de $\frac{1}{\rho}$.
- 3 Si $\pi = A \rightarrow BC$ entonces $\frac{1}{\pi}$ está seguido de $\frac{1}{\rho}$, con $\rho = B \rightarrow \beta$.
Análogamente para $\frac{2}{\pi}$.
- 4 Si $\pi = A \rightarrow a$, entonces $\frac{1}{\pi}$ está seguido por la cadena $\frac{1}{\pi} \frac{2}{\pi} \frac{2}{\pi}$.
- 5 Si $A \Rightarrow_{G'}^* \alpha$, entonces α comienza con $\frac{1}{\pi}$ para alguna $\pi = A \rightarrow \gamma$.

Considérese el siguiente lenguaje regular:

$$R_A = \{ \alpha \in \Sigma'^* \mid \alpha \text{ satisface 1-5} \}$$

Lema. $A \Rightarrow_{G'}^* \alpha$ sii $\alpha \in D_n^* \cap R_A$.

Suponiendo que el lema anterior es verdadero, entonces definimos el homomorfismo

$$h : \Sigma'^* \rightarrow \Sigma^*.$$

El teorema de Chomsky y Schützenberger IV

Si $\pi = A \rightarrow BC$ entonces

$$h\left(\frac{1}{\pi}\right) = h\left(\frac{1}{\pi}\right) = h\left(\frac{2}{\pi}\right) = h\left(\frac{2}{\pi}\right) = \epsilon.$$

Si $\pi = A \rightarrow a$ entonces

$$h\left(\frac{1}{\pi}\right) = a \quad \text{y} \quad h\left(\frac{1}{\pi}\right) = h\left(\frac{2}{\pi}\right) = h\left(\frac{2}{\pi}\right) = \epsilon.$$

El teorema se sigue directamente de esta definición.

Lema $A \Rightarrow_G^* \alpha$ sii $\alpha \in D_n^* \cap R_A$

Demostración. Por inducción en n para la dirección \Rightarrow .

Para el caso inverso, por inducción en la longitud de las cadenas en $D_n^* \cap R_A$. Basta considerar que

$$\alpha = \begin{matrix} 1 & 1 & 2 & 2 \\ \pi & \beta & \pi & \gamma \\ \pi & \pi & \pi & \pi \end{matrix}$$

y analizar los dos casos posibles para π , a saber, $A \rightarrow BC$ o bien $A \rightarrow a$.

En el primer caso, por hipótesis inductiva,

$$B \Rightarrow_G^* \beta \quad \text{y} \quad C \Rightarrow_G^* \gamma$$

y además β y γ satisfacen las condiciones 1–5. El segundo caso es aún más simple.

Algoritmo de Cocke, Kasami y Younger

- Es un algoritmo para *decidir* de manera eficiente si una cadena pertenece a un CFL.
- Sea L un CFL y sea $G = \langle \Sigma \Gamma, S, \rightarrow \rangle$ una CFG en forma normal de Chomsky tal que

$$L(G) = L.$$

- Sea $\alpha \in L$ con longitud n . Sean $0 \leq i < j \leq n$ y sea

$$\alpha_{i,j}$$

la subcadena de α que va de la posición $i + 1$ a la j .

- Sea $T_{i,j} \subseteq \Gamma$ el conjunto de no terminales que generaron $\alpha_{i,j}$.
- El algoritmo calcula el valor de todos los $T_{i,j}$ de manera inductiva. En particular, $S \in T_{0,n}$ si $S \Rightarrow_G^* \alpha$.
- El algoritmo es $O(pn^3)$, p el número de producciones en G .

for $i := 0$ to $n - 1$ do

$T_{i,i+1} := \emptyset$;
 for $A \rightarrow a \in G$ do
 if $a = \alpha_{i,i+1}$ then
 $T_{i,i+1} := T_{i,i+1} \cup \{A\}$

for $m := 2$ to n do

 for $i := 0$ to $n - m$ do
 $T_{i,i+m} := \emptyset$; for $j := i + 1$ to $i + m - 1$ do
 for $A \rightarrow BC \in G$ do
 if $B \in T_{i,j} \wedge C \in T_{j,i+m}$ then
 $T_{i,i+m} := T_{i,i+m} \cup \{A\}$

El teorema de Parikh

- Sean $\Sigma = \{a_1, \dots, a_n\}$, $\alpha \in \Sigma^*$ y $a \in \Sigma$.
- Sea $\#a(\alpha)$ es el número de apariciones del símbolo a en α .
- El mapa de Parikh es la función $\psi : \Sigma^* \rightarrow \mathbb{N}^n$ definida por

$$\psi(\alpha) = (\#a_1(\alpha), \dots, \#a_n(\alpha)).$$

- Sea $\bar{0} = (0, \dots, 0)$.
- Entonces, \mathbb{N}^n y $+$: $\mathbb{N}^n \times \mathbb{N}^n \rightarrow \mathbb{N}^n$ (suma por coordenadas) forman un monoide, *i.e.*, un conjunto con una operación binaria asociativa y un elemento neutro de esta operación, a saber, $\bar{0}$.
- Además, Σ^* y \cdot también forman un monoide.
- ψ es un homomorfismo entre Σ^* y \mathbb{N}^n .