Teoría de Lenguajes

Clase Teórica 7

Lema de Pumping para Lenguajes Independientes del Contexto

Primer cuartimestre 2016

Material compilado por el Profesor Julio Jacobo, a lo largo de distintas ediciones de la materia Teoría de Lenguajes en el Departamento de Computación, Facultad de Ciencias Exactas y Naturales, Universidad de Buenos Aires, revisado recientemente por Verónica Becher.

Bibliografía: Capítulos 5 (árboles de derivación), 6 y 7 *Introduction to Automata Theory, Languages and Computation*, J. Hopcroft, R. Motwani, J. Ullman, Second Edition, Addison Wesley, 2001.

Lema 1 ("Pumping" para lenguajes independientes del contexto.)**.** *Para todo lenguaje L independiente de contexto, existe $n > 0$ tal que*
para toda cadena α en L con $|\alpha| \geq n$,

- *Existe una descomposición de α en cadenas r, x, y, z, s, es decir $\alpha = rxyzs$.*
- $|xyz| \leq n$.
- $|xz| \geq 1$.
- *Para todo $i \geq 0$, la cadena rx^iyz^is pertenece a L.*

Ejemplo 1. *El lenguaje $L = \{a^m b^m c^m : m \geq 1\}$ no es independiente de contexto.*

Demostración. Asumamos que L es independiente de contexto.
Sea n dado por el Lema de Pumping y sea $\alpha = a^n b^n c^n$.
La longitud n no puede incluir al mismo tiempo $a$es y cs.

Por el Lema de Pumping, debe haber cadenas donde
(1) $a$es y bs se repitan, y c permanecen igual, o bien
(2) bs y cs se repitan y las $a$es permanecen igual.

Pero tales cadenas no están en L.
Llegamos a esta imposibilidad porque asumimos que L era independiente de contexto. Concluimos que no lo es.

Ejemplo 2. *El lenguaje $L = \{ww : w \in \{a, b\}^*\}$ no es independiente de contexto.*

Demostración. Asumamos que sí.
Sea n dado por el Lema de Pumping, y sea $\alpha = a^n b^n a^n b^n$.
La longitud n no puede incluir las $a$es en la primera mitad y las $a$es en la segunda mitad, Lo mismo ocurre para las bs.

Por el Lema de Pumping, hay palabrasen L donde a y b se repiten en *una* mitad, pero no en la otra. Pero por la definición de L esto es imposible.

Llegamos a esta imposibilidad porque asumimos L es independiente de contexto. Concluimos que no lo es.

Ejemplo 3. *El lenguaje $L = \{a^{m^2} : m \geq 0\}$ no es independiente de contexto.*

Demostración. Asumamos que sí.
Sean n dado por el Lema de Pumping. Sea $\alpha = a^{n^2}$.

La cadena de longitud inmediatamente mayor que la de α en L tiene longitud $(n + 1)^2$. Por el Lema de Pumping la cantidad de $a$es a ser repetidas puede ser menor o igual que n. Dado que

$$n^2 + n < (n + 1)^2,$$

hay palabras en L cuya longitud no es un cuadrado perfecto.

Por la definición de L esto es imposible. La imposibilidad provino de suponer que L es independiente de contexto. Concluimos que no lo es.

Para la demostración de Lema de Pumping para lenguajes independientes de contexto usaremos gramáticas libres de contexto.

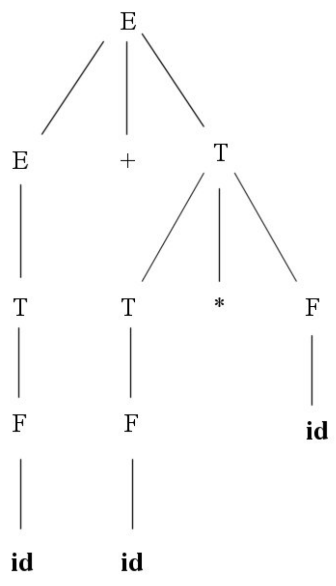
Definición 1 (Árbol de derivación)**.** *Sea $G = \langle V_N, V_T, P, S\rangle$ una gramática independiente de contexto y sea $\alpha \in V_T^*$. Un árbol de derivación para α en G es tal que*

1. *la etiqueta de la raiz es el símbolo distinguido S.*
2. *cada vértice posee una etiqueta que pertenece al conjunto $V_N \cup V_T \cup \{\lambda\}$.*
3. *si un vértice es interior, su etiqueta debe pertenecer a V_N.*
4. *si un vértice n posee la etiqueta A y sus hijos $n_1, \ldots, n_k$ poseen etiquetas $X_1, \ldots, X_k$ respectivamente, entonces $A \to X_1, \ldots, X_k$ debe ser una producción de P.*
5. *si un vértice posee la etiqueta λ, entonces es una hoja y es el único hijo de su padre.*

Ejemplo 4. *Sea $G = \langle V_N, V_T, P, S\rangle$ gramática independiente de contexto dada por:*

$$\begin{aligned} E &\to E + T \mid T \\ T &\to T * F \mid F \\ F &\to \mathbf{id} \mid \mathbf{const} \mid (E) \end{aligned}$$

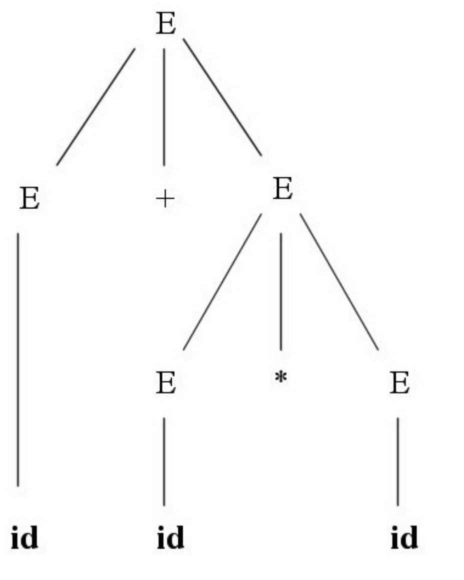
Este es un árbol de derivación para $\mathbf{id} + \mathbf{id} * \mathbf{id}$ *en G (y es único):*

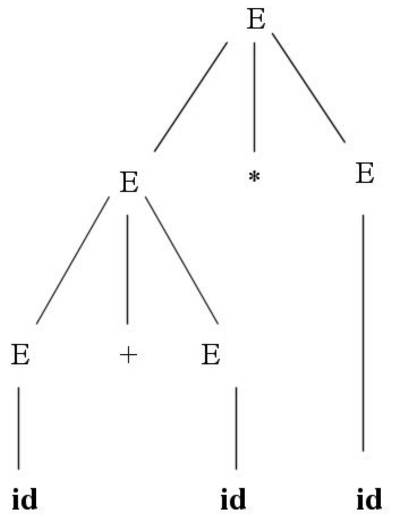


Ejemplo 5. *Dada la gramática $G = \langle \{E\}, \{+, *, \mathbf{id}, \mathbf{const}\}, P, E\rangle$ con*

$$P = \{ \quad \begin{aligned} & E \to E + E, \\ & E \to E * E, \\ & E \to \mathbf{id}, \\ & E \to \mathbf{const}\}, \end{aligned}$$

*Para G podemos dar dos árboles de derivación distintos para **id + id * id**:*





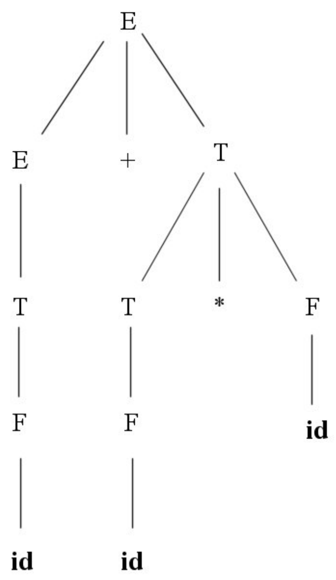
Definición 2. *Dada una gramática $\langle V_N, V_T, P, S\rangle$ y dados $X \in V_T$ o $X \in V_N$. Llamamos camino de X en un árbol $\mathcal{T}(A)$, con $A \in V_N$, a la secuencia $A, X_1, \ldots, X_k, X$ tal que*

$$A \Rightarrow \ldots X_1 \ldots \Rightarrow \ldots X_2 \ldots \Rightarrow \ldots\ldots \Rightarrow \ldots X_k \ldots \Rightarrow \ldots X \ldots.$$

Una hoja de un árbol $\mathcal{T}(A)$, con $A \in V_N$, es una cadena $x \in V_T^*$ para la cual hay un camino (que empieza en A y termina en x).

Definición 3. *La altura de $\mathcal{T}(A)$ es $\max\{|\alpha| : \alpha x$ es un camino de x y x es una hoja de $\mathcal{T}(A)\}$*

Consideremos este árbol de derivación para $id + id * id$ en la gramática G de uno de los ejemplos.



Altura de $\mathcal{T}(E) = \max\{|\alpha| : \alpha x$ es un camino de x y x es una hoja de $\mathcal{T}(A)\}$
Por lo tanto, la altura de $\mathcal{T}(E)$ es 4.

Lema 2. *Sea $G = \langle V_N, V_T, P, S\rangle$ una gramática independiente de contexto con $P \neq \emptyset$, sea $\alpha \in (V_N \cup V_T)^*$ y sea $\mathcal{T}(S)$ un árbol de derivación para α en G cuya altura designaremos h. Sea*

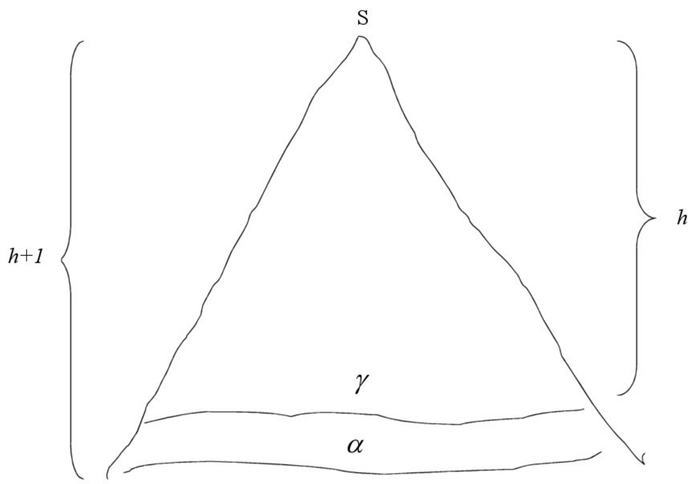
$$a = \max\{k : (k = |\beta|, A \to \beta \in P, \beta \neq \lambda) \text{ o } (k = 1, A \to \lambda \in P)\}.$$

Entonces $a^h \geq |\alpha|$.

Demostración. Por inducción en h.

Caso base, $h = 0$. El único árbol de derivación posible es el símbolo distinguido S, cuya altura es 0. Por lo tanto $a^h = a^0 = 1 = |S|$.

Caso inductivo. Sea γ la base del árbol $\mathcal{T}(S)$ para la altura h. Asumamos HI: $a^h \geq |\gamma|$. Sea α la base de $\mathcal{T}(S)$ para la altura $h+1$.



Luego $a\,|\gamma| \geq |\alpha|$.

Pero, por H.I. $a^h \geq |\gamma|$, entonces $a^{h+1} = aa^h \geq a\,|\gamma| \geq |\alpha|$. □

Recodemos el enunciado del Lema de Pumping que debemos demostrar.

Lema 3 ("Pumping" para lenguajes independientes del contexto.). *Para todo lenguaje L independiente de contexto, existe $n > 0$ tal que*

para toda cadena α en L con $|\alpha| \geq n$,

- *Existe una descomposición de α en cadenas r, x, y, z, s, es decir $\alpha = rxyzs$.*
- $|xyz| \leq n$.
- $|xz| \geq 1$.
- *Para todo $i \geq 0$, la cadena rx^iyz^is pertenece a L.*

Demostración.

Sea $G = \langle V_N, V_T, P, S\rangle$ una gramática independiente de contexto tal que $L = \mathcal{L}(G)$. Sea $a = \max\left(\{2\} \cup \{|\beta| : A \to \beta \in P, A \in V_N, \beta \in (V_N \cup V_T)^*\}\right)$

Tomemos $n = a^{|V_N|+1}$.

Sea α una cadena en L tal que $|\alpha| \geq n$.

Sea $\mathcal{T}(S)$ un árbol de derivación para α en G, de altura mínima.

Por el Lema anterior, $a^h \geq |\alpha|$.

Por lo tanto, $a^h \geq |\alpha| \geq n = a^{|V_N|+1}$.

Luego, $h \geq |V_N| + 1$.

Como $h \geq |V_N| + 1$, hay un camino a terminales de α de longitud mayor o igual a $|V_N| + 1$.

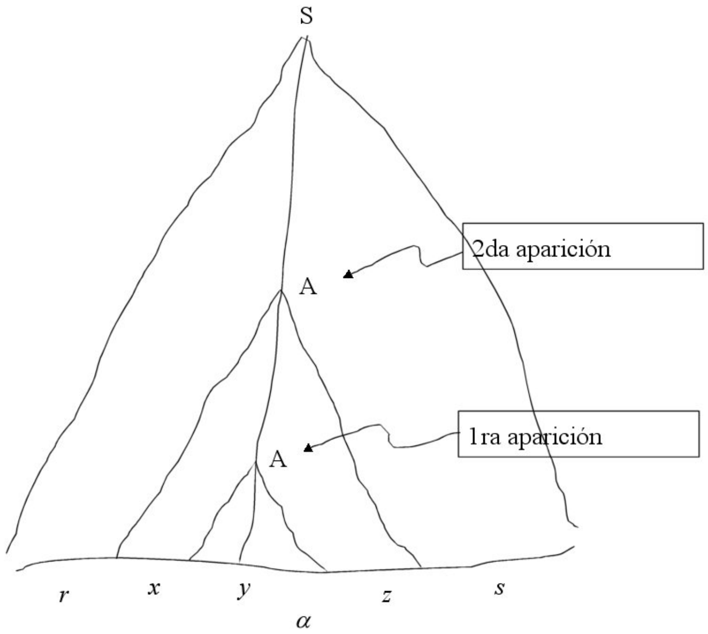
Como la cantidad de símbolos no terminales es $|V_N|$, entonces en ese camino existe un no-terminal repetido. Llamémoslo A.

Recorriendo dicho camino en forma ascendente consideremos la primera y la segunda aparición de A, como se ve en la figura.

La segunda aparición de A da lugar a la cadena xyz. La altura del árbol generado por esta segunda aparición $\mathcal{T}(A)$ es menor o igual que $|V_N| + 1$, por lo tanto,

$$|xyz| \leq a^{|V_N|+1} = n$$

Notar que x y z no pueden ser simultáneamente nulas, ya que sino el árbol de derivación para α no sería de altura mínima.



Luego, existen cadenas $r, x, y, z, s \in V_T^*$ tales que
$\alpha = rxyzs$, $|xyz| \leq n$, $|xz| \geq 1$, y

$$S \stackrel{*}{\Rightarrow} rAs \stackrel{*}{\Rightarrow} rxAzs \stackrel{*}{\Rightarrow} rxyzs.$$

Por lo tanto tenemos $A \stackrel{*}{\Rightarrow} y$ y también $A \stackrel{*}{\Rightarrow} xAz$.
Demostremos que $\forall i \geq 0$, $rx^iyz^is \in L$, por inducción en i.

Caso Base, $i = 0$. Tenemos que $S \stackrel{*}{\Rightarrow} rAs \stackrel{*}{\Rightarrow} rys$. Por lo tanto, $rys = rx^0yz^0s$ está en L.

Caso inductivo, $i \geq 0$.Asumamos HI: $rx^iyz^is \in L$.
Veamos que se cumple para $i + 1$.Sabemos que

$$S \stackrel{*}{\Rightarrow} rx^iAz^is \stackrel{*}{\Rightarrow} rx^iyz^is.$$

Pero también se tiene que cumplir que

$$S \stackrel{*}{\Rightarrow} rx^iAz^is \stackrel{*}{\Rightarrow} rx^ixAzz^is \stackrel{*}{\Rightarrow} rx^ixyzz^is = rx^{i+1}yz^{i+1}s.$$

Por lo tanto $rx^{i+1}yz^{i+1}s$ está en L. $\square$

Algoritmos de decisión

Teorema 1. *Hay un algoritmo para decidir si un lenguaje independiente de contexto es vacío.*

Sea n la constante del Lema de Pumping. L es vacío si y solo si ninguna palabra de longitud menor que n pertenece a L. (Si hubiera una de longitud n o más, por el Lema de Pumping también habría una de longitud menor que n.)

Algoritmos de decisión

Teorema 2. *Hay un algoritmo para decidir si un lenguaje independientes de contexto es finito.*

Sea n la constante del Lema de Pumping. L es finito si y solo si ninguna palabra de longitud entre n y $2n - 1$. (izq a derecha) Supongamos L es finito pero acepta una palabra de longitud entre n y $2n - 1$. Lema de Pumping también L tiene infinitas de longitud mayor que n. Contradicción.

(derecha a izquierda) L no acepta ninguna palabra de longitud entre n y $2n - 1$ pero es infinito. Sea w la palabra en L más corta de longitud mayor o igual que $2n$. El lema de Pumping afirma que existe otra palabra en L más corta que w.

Algoritmos de decisión

Teorema 3. *Hay un algoritmo para decidir la pertenecia de una palabra a un lenguaje independiente de contexto.*

Problema: Necesitamos una cota del tamaño de los árboles de derivación. Las producciones de la forma $A \to B$ y $B \to A$ son un problema.

La solución es una gramática equivalente tal que todas las produccines con excepción de $S \to \lambda$, sean: $A \to a$ con $a \in V_T$

$A \to BC$, con $B, C \in V_N$.

Esta gramática equivalente se llama *Forma Normal de Chomsky* para gramáticas independientes de contexto (ver Capítulo 7 del libro de Hoproft y Ulmann).

Usando la gramática en forma normal, el árbol de derivación tiene una altura menor o igual a $2 * |w| - 1$. Luego, el algoritmo consiste en chequear exhaustivaemnte todos los árboles de derivación de tamaño 1 hasta la cota.

Teorema 4. *Los lenguajes independientes del contexto están cerrados por unión, concatenación y clausura de Kleene.*

No están cerrados por intersección, diferencia, complemento.

Teorema 5. *Si L_1 y L_2 son dos lenguajes independientes del contexto, entonces $L_1 \cup L_2$ lo es también.*

Demostración. Como L_1 y L_2 son dos lenguajes independientes del contexto, entonces existen dos gramáticas independientes del contexto $G_1 = \langle V_{N_1}, \Sigma, P_1, S_1 \rangle$ y $G_2 = \langle V_{N_2}, \Sigma, P_2, S_2 \rangle$ tales que $L_1 = \mathcal{L}(G_1)$ y $L_2 = \mathcal{L}(G_2)$.

Sin pérdida de generalidad, supongamos que $V_{N_1} \cap V_{N_2} = \emptyset$.

Definamos entonces

$$G = \langle V_{N_1} \cup V_{N_2} \cup \{S\}, \Sigma, P_1 \cup P_2 \cup \{S \to S_1, S \to S_2\}, S \rangle .$$

Puede demostrarse que $\forall \alpha \in \Sigma^*$, $\alpha \in \mathcal{L}(G)$ si y solo si $\alpha \in \mathcal{L}(G_1) \cup \mathcal{L}(G_2)$. □

Teorema 6. *Si L_1 y L_2 son dos lenguajes independientes del contexto, entonces $L_1 L_2$ lo es también.*

Demostración. Como L_1 y L_2 son dos lenguajes independientes del contexto, entonces existen dos gramáticas independientes del contexto $G_1 = \langle V_{N_1}, \Sigma, P_1, S_1 \rangle$ y $G_2 = \langle V_{N_2}, \Sigma, P_2, S_2 \rangle$ tales que $L_1 = \mathcal{L}(G_1)$ y $L_2 = \mathcal{L}(G_2)$.

Supongamos además, sin pérdida de generalidad que $V_{N_1} \cap V_{N_2} = \emptyset$. Definamos entonces

$$G = \langle V_{N_1} \cup V_{N_2} \cup \{S\}, \Sigma, P_1 \cup P_2 \cup \{S \to S_1 S_2\}, S \rangle .$$

Puede demostrarse que $\forall \alpha \in \Sigma^*$, $\alpha \in \mathcal{L}(G)$ si y solo si $\alpha \in \mathcal{L}(G_1) \mathcal{L}(G_2)$. □

Teorema 7. *Si L_1 es un lenguaje independiente de contexto entonces L_1^+ lo es también.*

Demostración. Como L_1 es un lenguaje independiente de contexto, entonces existe una gramática independiente de contexto $G_1 = \langle V_{N_1}, \Sigma, P_1, S_1 \rangle$ tal que $L_1 = \mathcal{L}(G_1)$.

Definamos entonces

$$G = \langle V_{N_1} \cup \{S\}, \Sigma, P_1 \cup \{S \to S_1 S, S \to S_1\}, S \rangle .$$

Puede demostrarse que $\forall \alpha \in \Sigma^*$, $\alpha \in \mathcal{L}(G)$ si y solo si $\alpha \in \mathcal{L}(G_1)^+$. □

Teorema 8. *Si L_1 y L_2 son dos lenguajes independientes del contexto, entonces $L_1 \cap L_2$ no siempre es un lenguaje independiente de contexto.*

Demostración.Sean

$$L_1 = \left\{a^n b^m c^l : m, n, l \geq 0 \quad \wedge \quad n = m\right\}$$
$$L_2 = \left\{a^n b^m c^l : n, m, l \geq 0 \quad \wedge \quad m = l\right\} .$$

Son lenguajes independientes del contexto ya que, $L_1 = \mathcal{L}(G_1)$ donde

$$G_1 = \langle \{S, A, C\}, \{a, b, c\}, \{S \to AC \mid C,\ A \to aAb \mid ab,\ C \to cC \mid \lambda\}, S \rangle$$

Lo mismo puede hacerse para L_2. Sin embargo, el lenguaje

$$L_1 \cap L_2 = \{a^m b^m c^m : m \geq 0\} ,$$

no es independiente de contexto (primer ejemplo de esta clase).□ Por lo tanto los lenguajes independientes de contexto

- no están cerrados por complementación, ya que $(A^c \cup B^c)^c = A \cap B$.
- no están cerrados por diferencia, ya que $A^c = \Sigma^* - A$.

Recordemos

Definición 4 (Lenguaje independiente de contexto determinístico)**.** *Un lenguaje L es independiente de contexto determinístico si existe un autómata de pila determinístico (APD) M tal que $L = \mathcal{L}(M)$.*

Teorema 9. *El complemento de un lenguaje independiente de contexto determinìstico es otro lenguaje independiente de contexto determinístico.*

Idea ingenua para la demostración, que no funciona.
Dar un autómata de pila M que reconozca el lenguaje, y definir otro, llamémoslo M', donde los estados finales pasen a ser no-finales y viceversa. No funciona porque:

* El autómata M puede no consumir totalmente la cadena de entrada.
Esto puede deberse a:

1. el autómata M alcanza una configuración desde la cual ninguna transición es posible.
2. el autómata M entra en un ciclo infinito de transiciones λ

En ambos casos, la cadena no consumida es rechazada por ambos autómatas. Por lo tanto, el lenguaje aceptado porM' no es el complemento del lenguaje aceptado por M.

* El autómata M consume toda las cadena de entrada, pero hay alguna cadena que, luego de consumida, deja al autómata en una configuración tal que éste pueda hacer transiciones λ pasando por estados finales y no-finales. El lenguaje aceptado por M' no es el complemento del lenguaje aceptado por M.

Lema 4. *Para todo APD M, existe otro otro APD M' equivalente que siempre consume la cadena de entrada.*

Demostración. Dado el APD $M = \langle Q, \Sigma, \Gamma, \delta, q_0, Z_0, F\rangle$, definimos $M' = \langle Q \cup \{q_0', d, f\}, \Sigma, \Gamma \cup \{X_0\}, \delta', q_0', X_0, F \cup \{f\}\rangle$ donde

1. $\delta'(q_0', \lambda, X_0) = (q_0, Z_0X_0)$ para evitar que el APD se detenga por haberse vaciado la pila.
2. para todo $q \in Q, a \in \Sigma, Z \in \Gamma$,
   si $\delta(q, a, Z) = \delta(q, \lambda, Z) = \phi$ entonces $\delta'(q, a, Z) = (d, Z)$

   $\forall q \in Q, a \in \Sigma, \delta'(q, a, X_0) = (d, X_0)$
3. para todo $a \in \Sigma, Z \in \Gamma \cup \{X_0\}, \delta'(d, a, Z) = (d, Z)$ para setear el estado trampa.
4. (loop λ) Si para $q \in Q, Z \in \Gamma$, ocurre que

$$\forall i \geq 1, \exists q_i, \gamma_i \text{ para los cuales } (q, \lambda, Z) \overset{i}{\vdash} (q_i, \lambda, \gamma_i), \text{ entonces}$$

   si ningún q_i es final, $\delta'(q, \lambda, Z) = (d, Z)$
   si algún q_i es final, $\delta'(q, \lambda, Z) = (f, Z)$.
5. para todo $Z \in \Gamma \cup \{X_0\}$ $\delta'(f, \lambda, Z) = (d, Z)$.
6. para todo $q \in Q, a \in \Sigma \cup \{\lambda\}, Z \in \Gamma$,
   si $\delta'(q, a, Z)$ no ha sido definida aún, $\delta'(q, a, Z) = \delta(q, a, Z)$.

El APD M' así definido consume siempre toda la cadena de entrada. Para verlo consideremos lo siguiente: supongamos que existe una cadena de entrada xy tal que M' no consume más allá del prefijo x, entonces debe ocurrir que

$$(q_0', xy, X_0) \underset{M'}{\overset{*}{\vdash}} (q, y, Z_1, \ldots Z_kX_0)$$

y que a partir de esa configuración instantánea, no se consume más entrada. Pero esto no puede ocurrir, proque siempre hay transiciones posibles y no hay ciclos λ. $\square$ Ahora sí, repetimos el enunciado del teorema y damos la demostración.

Teorema 10. *El complemento de un lenguaje independiente de contexto determinìstico es otro lenguaje independiente de contexto determinístico.*

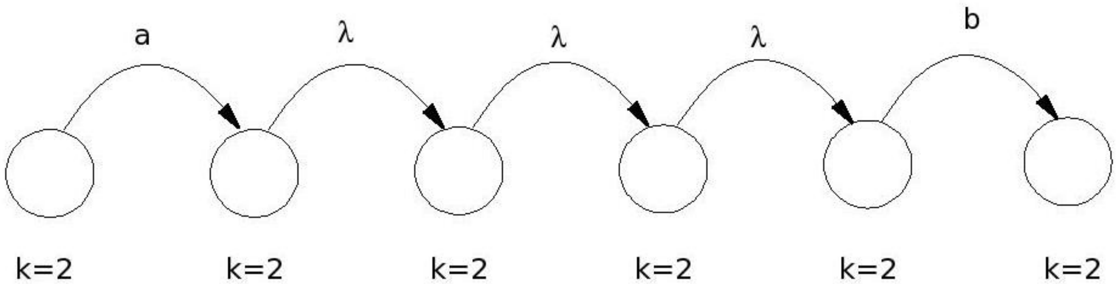
Demostración del Teorema. Dado el APD $M = \langle Q, \Sigma, \Gamma, \delta, q_0, Z_0, F \rangle$, que por lo ya visto podemos considerar que consume toda la cadena de entrada, definimos $M' = \langle Q', \Sigma, \Gamma, \delta', q_0', Z_0, F' \rangle$ un APD donde

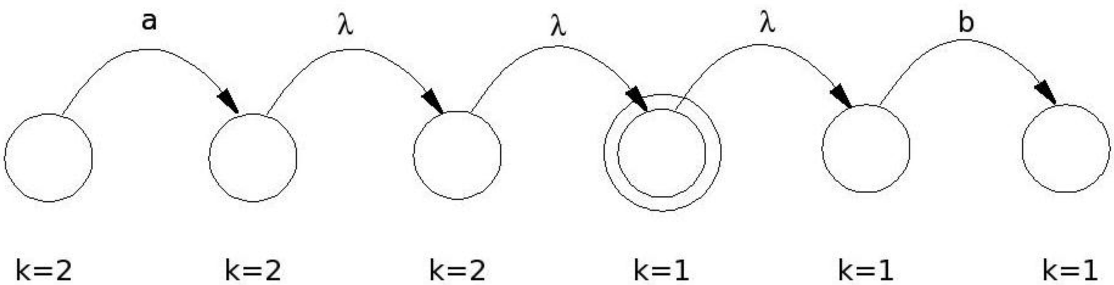
$$Q' = \{[q, k] : q \in Q \ \text{ y } \ k = 1, 2 \text{ ó } 3\},$$

El propósito de k es detectar si entre transiciones con consumo de entrada pasó o no por un estado final. Si pasó por un estado final, entonces se hará $k = 1$, sino se hará $k = 2$.

$$F' = \{[q, 3] : q \in Q\} \ \text{ y}$$

$$q_0' = \begin{cases} [q_0, 1] & \text{si } q_0 \in F \\ [q_0, 2] & \text{si } q_0 \notin F \end{cases}$$

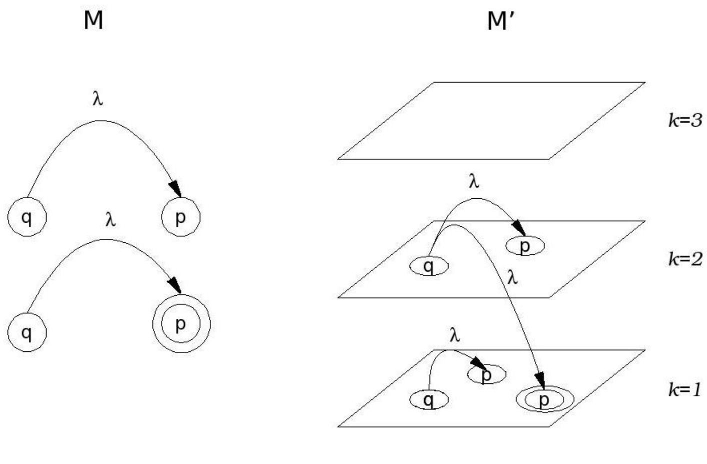




La función de transición δ' se define por

1. Si $\delta(q, \lambda, Z) = (p, \gamma)$ entonces, para $k = 1, 2$

$$\delta'([q, k], \lambda, Z) = \begin{cases} ([p, 1], \gamma) & \text{si } k = 1 \text{ ó } p \in F \\ ([p, 2], \gamma) & \text{si no} \end{cases}$$

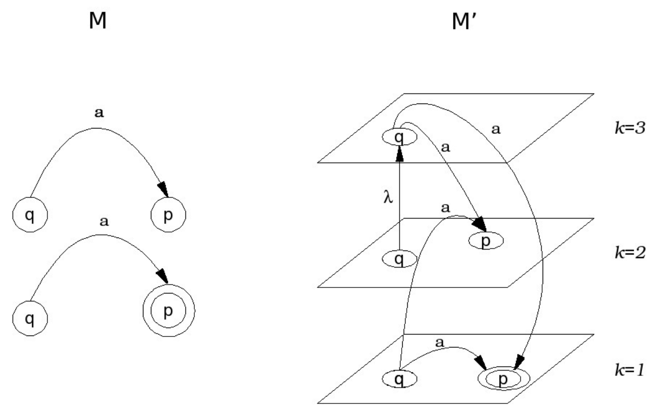


2. Si $\delta(q, a, Z) = (p, \gamma)$ entonces, para $k = 1, 2$

$$\delta'([q, 2], \lambda, Z) = ([q, 3], Z)$$

y

$$\delta'([q,1],a,Z) = \delta'([q,3],a,Z) = \begin{cases} ([p,1],\gamma) & \text{si } p \in F \\ ([p,2],\gamma) & \text{si } p \notin F \end{cases}$$



3

□

Definición 5. *Para todo $A \in V_N$, $\alpha_1, \alpha_2 \in (V_T \cup V_N)^*$,*

- $\alpha_1 A \alpha_2 \underset{L}{\Rightarrow} \alpha_1 \alpha' \alpha_2$ *es una drivación más a la izquierda si $A \to \alpha' \in P$ y $\alpha_1 \in V_T^*$.*
- $\alpha_1 A \alpha_2 \underset{R}{\Rightarrow} \alpha_1 \alpha' \alpha_2$ *es una derivación más a la derecha si $A \to \alpha' \in P$ y $\alpha_2 \in V_T^*$.*

Definición 6 (Gramáticas ambiguas)**.** *Una gramática independiente de contexto G es ambigua si existe $\alpha \in \mathcal{L}(G)$ con más de una derivación más a la izquierda.*

Definición 7. *Un lenguaje independiente de contexto es inherentemente ambiguo si solamente admite gramáticas ambiguas.*

CFL Determinístico implica CFL no ambiguo

Recordemos

Definición 8. *Un lenguaje L es independiente de contexto determinístico si existe un autómata de pila determinístico (APD) M tal que $L = \mathcal{L}(M)$.*

Los lenguajes determinísticos independientes del contexto admiten una gramática independiente de contexto no ambigua.

Y hay una clase importante de lenguajes no determinísticos independientes del contexto que también admiten una gramática independiente de contexto no ambigua.

Ejemplo 6. *Este lenguaje es independiente de contexto es inherentemente ambiguo:*

$$\{a^n b^m c^m d^n | n, m > 0\} \cup \{a^n b^n c^m d^m | n, m > 0\}.$$

Es independiente de contexto porque es la unión de dos conjuntos que lo son.

Hopcroft y Ullman (1979) mostraron que no hay forma de derivar de manera no ambigua las cadenas de la intersección de ambos lenguajes, es decir del conjunto

$$\{a^n b^n c^n d^n | n > 0\},$$

(además este conjunto intersección no es independiente de contexto).

El problema de si una gramática es inherentemente ambigua es indecidible. Es equivalente al problema de correspondencia de Post.

Dadas dos listas de palabras sobre un alfabeto (con al menos dos símbolos) $\alpha_1, \ldots, \alpha_N$ y $\beta_1, \ldots, \beta_N$ decidir si hay una secuencia de índices $(i_k)_{1 \le k \le K}$ con $K \geq 1$ y $1 \leq i_k \leq N$ tal que $\alpha_{i_1} \ldots \alpha_{i_K} = \beta_{i_1} \ldots \beta_{i_K}$.

Ejemplo

$\alpha_1 = a, \alpha_2 = ab, \alpha_3 = bba$ $\beta_1 = baa, \beta_2 = aa, \beta_3 = bb$.

Solución= (3, 2, 3, 1) porque

$\alpha_3\alpha_2\alpha_3\alpha_1 = bba + ab + bba + a = bbaabbbaa = bb + aa + bb + baa = \beta_3\beta_2\beta_3\beta_1$.

Preguntas

1. Sea L un lenguaje. Si todas las palabras de L validan el Lema de Pumping para lenguajes independientes del contexto, ¿Podemos concluir que L es un lenguaje independendiente del contexto?
2. Sea L un lenguaje regular. Demostrar que todas las palabras de L validan el Lema de Pumping para lenguajes independientes del contexto.
3. Mostrar que $L = \{a^p : p \text{ es número primo}\}$ no es independente del contexto. Ayuda: . Asumir L es independiente de contexto, y n es la longitud dada por el Lema de Pumping. Sea m el primer primo mayor o igual que n, considerar $\alpha = a^m$ y bombear $m + 1$ veces.
4. Demostrar que un lenguaje regular intersección un lenguaje independiente de contexto es independiente de contexto. Ayuda: Definir el autómata de pila que lo reconoce.
5. Demostrar que hay lenguajes que pueden ser reconocidos con un autómata con dos pilas pero no pueden ser reconocidos con un autómata con una sola pila. Ayuda: los ejemplos en esta clase.