Teoría de Lenguajes

Clase Teórica 9

Lema de Pumping, Propiedades de Clausura y Algoritmos de Decisión para Lenguajes Libres de Contexto

Primer Cuatrimestre 2024

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 ("Pumping" para lenguajes libres de contexto.)

Para todo lenguaje L libre 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

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

Demostración. Asumamos que L es libre 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 libre de contexto. Concluimos que no lo es.

Ejemplo

El lenguaje $L = \{ww : w \in \{a, b\}^*\}$ no es libre 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 libre de contexto. Concluimos que no lo es.

Ejemplo

El lenguaje $L = \{a^{m^2} : m \geq 0\}$ no es libre 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 libre de contexto. Concluimos que no lo es.

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

Definición (Árbol de derivación)

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

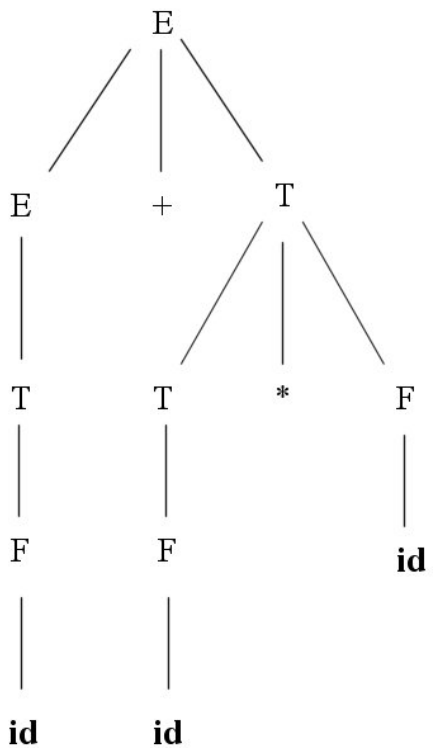
1. *la etiqueta de la raiz es el símbolo distinguido S.*
2. *cada vértice posee una etiqueta que pertenece al conjunto $V_N \cup V_T \cup \{\lambda\}$.*
3. *si un vértice es interior, su etiqueta debe pertenecer a V_N.*
4. *si un vértice n posee la etiqueta A y sus hijos $n_1, \ldots, n_k$ poseen etiquetas $X_1, \ldots, X_k$ 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 de un árbol de derivación

Sea $G = \langle V_N, V_T, P, S \rangle$ gramática libre de contexto dada por:

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

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



Definición

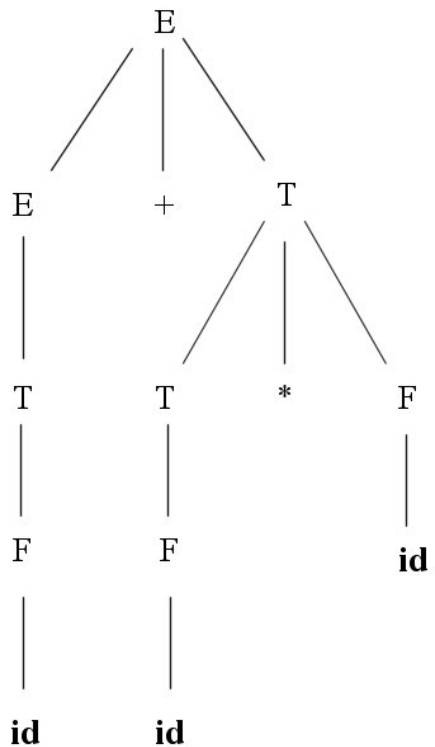
Dada una gramática $\langle V_N, V_T, P, S\rangle$ *y dados* $X \in V_T$ *o* $X \in V_N$*.*
Llamamos camino de X *en un árbol* $\mathcal{T}(A)$*, con* $A \in V_N$*, a la secuencia* $A, X_1, \ldots, X_k, X$ *tque corresponde a las etiquetas de los vértices de la rama del árbol. Una hoja de un árbol* $\mathcal{T}(A)$*, es un símbolo* $t \in V_T$ *para el cual hay un camino que empieza en* A *y termina en* t*.*

Definición

La altura de $\mathcal{T}(A)$ *es*

$$\text{máx}\{|\alpha| : \alpha x \text{ es un camino de } x \text{ y } x \text{ es una hoja de } \mathcal{T}(A)\}$$

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.

Proposición

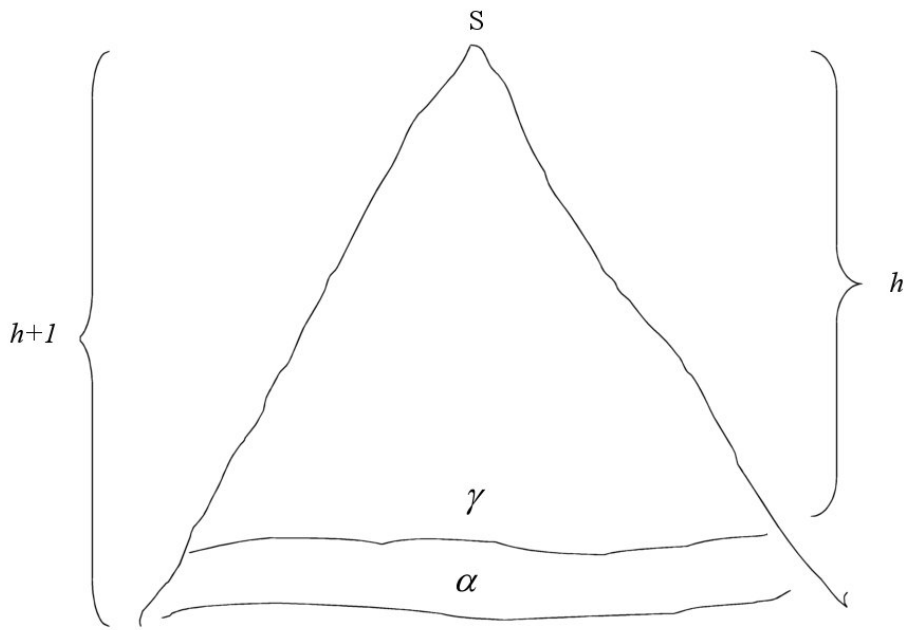
Sea $G = \langle V_N, V_T, P, S \rangle$ *una gramática libre de contexto con* $P \neq \emptyset$*, sea* $\alpha \in (V_N \cup V_T)^*$ *y sea* $\mathcal{T}(S)$ *un árbol de derivación para* α *en* G *cuya altura designaremos* h*. Sea*

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

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 ("Pumping" para lenguajes libres de contexto.)

Para todo lenguaje L libre 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 del Lema de Pumping para Libres de Contexto.

Sea $G = \langle V_N, V_T, P, S \rangle$ una gramática libre de contexto tal que $L = \mathcal{L}(G)$. Sea

$a = \text{máx}\left(\{2\} \cup \{|\beta| : A \to \beta \in P, A \in V_N, \beta \in (V_N \cup V_T)^*\}\right)$

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

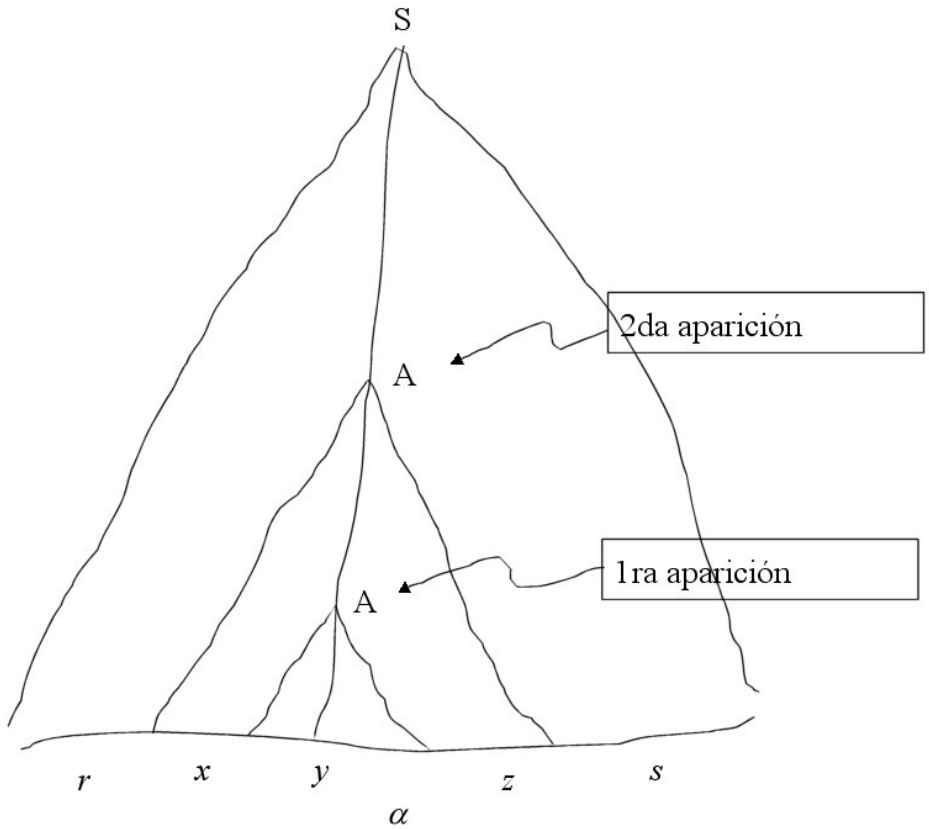
Por lo tanto, $a^h \geq |\alpha| \geq n = a^{|V_N|+1}$.

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.

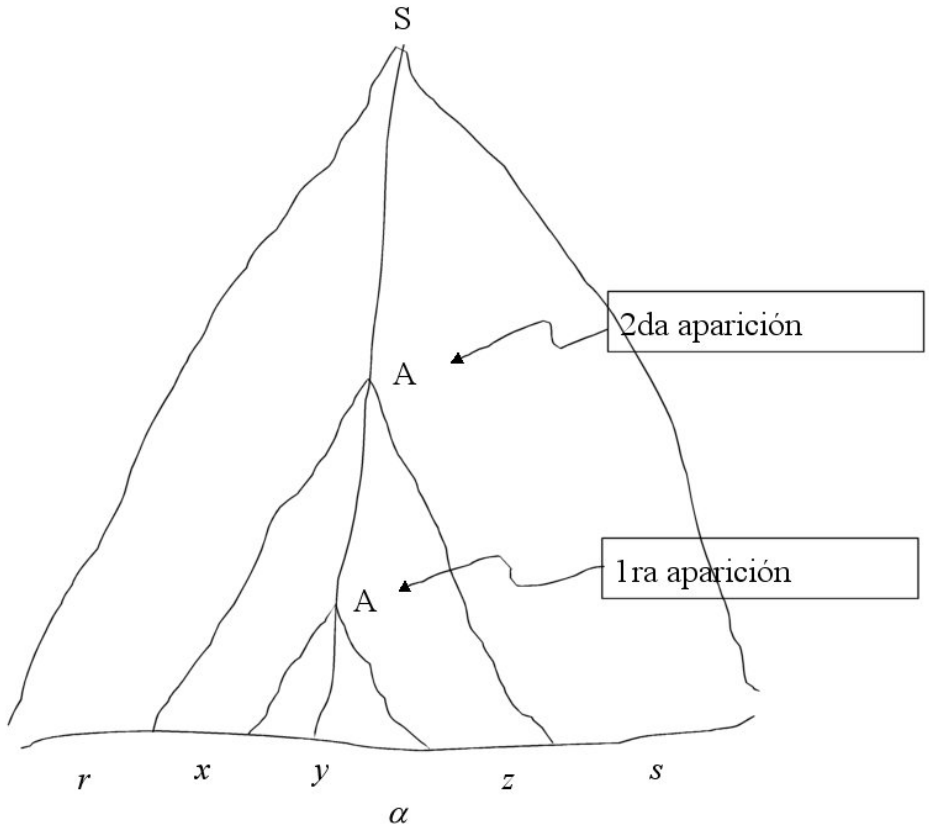
Recorriendo dicho camino en forma ascendente consideremos la primera y la segunda aparición de A, como se ve en la figura.



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

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

Por lo tanto tenemos $A \overset{*}{\Rightarrow} y$ y también $A \overset{*}{\Rightarrow} xAz$.

Demostremos que $\forall i \geq 0$, $rx^iyz^is \in L$, por inducción en i.

Caso Base, $i = 0$. Tenemos que $S \overset{*}{\Rightarrow} rAs \overset{*}{\Rightarrow} rys$. 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

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

Pero también se tiene que cumplir que

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

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

Cota en la cantidad de pasos en una derivación

Una gramática libre de contexto es recursiva a izquierda si hay algún símbolo $A \in V_N$ y algún $\alpha \in (V_N \cup V_T)^*$ tal que $A \underset{L}{\overset{+}{\Rightarrow}} A\alpha$.

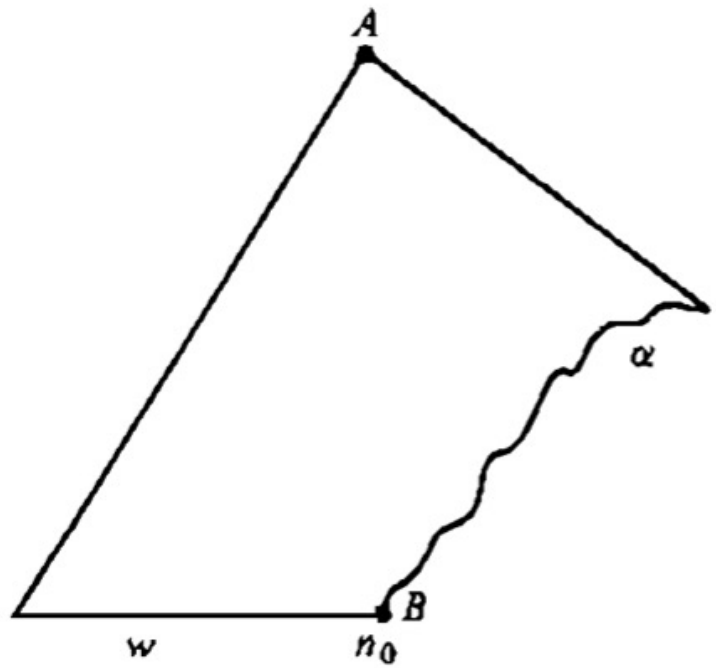
Lema (Lema 4.1 Aho-Ullman vol. 1)

Sea $G = (N, T, P, S)$ libre de contexto y no recursiva a izquierda.
Existe una constante c tal que si $A \underset{L}{\overset{i}{\Rightarrow}} wB\alpha$ y $|w| = n$ entonces $i \leq c^{n+2}$.

Se puede demostrar un resultado mucho más ajustado, con i lineal en n pero la misma constante.

Demostración del lema

Llamemos k a la cantidad de símboos no-terminales.
Sea $\mathcal{A}$ el ábol de la derivación más a la izquierda para $A \underset{L}{\overset{i}{\Rightarrow}} wB\alpha$.



Sea n_0 el nodo con etiqueta B en la derivación $A \underset{L}{\overset{i}{\Rightarrow}} wB\alpha$. Notemos que, por tratarse de la derivación más a la izquierda, todos los caminos a la derecha del camino desde la raíz a n_0 son más cortos, o del mismo largo.

Supongamos que hay un camino de longitud mayor o igual que $k(n+2)$ arcos de la raíz a la hoja,

$$A = \alpha_0 \underset{L}{\Rightarrow} \alpha_1 \underset{L}{\Rightarrow} \ldots \underset{L}{\Rightarrow} \alpha_{k(n+2)-1} \underset{L}{\Rightarrow} \alpha_{k(n+2)} = wB\alpha$$

Visualicemos esta derivación por segmentos así:

$$\alpha_0 \underset{L}{\Rightarrow} \alpha_1 \underset{L}{\Rightarrow} \ldots \underset{L}{\Rightarrow} \alpha_k$$

$$\alpha_k \underset{L}{\Rightarrow} \ldots \underset{L}{\Rightarrow} \alpha_{2k}$$

$$\ldots$$

$$\alpha_{(n+1)k} \underset{L}{\Rightarrow} \ldots \underset{L}{\Rightarrow} \alpha_{(n+2)k}.$$

Son $(n+2)$ segmentos de derivaciones. Es imposible que cada uno de estos produzca uno o más símbolos de wB, porque $|wB| = n+1$. Entonces ¡hay al menos uno de estos segmentos que no produce ningun símbolo!

Entonces en el árbol de derivación $\mathcal{A}$ hay un segmento, digamos el iésimo,

$$\alpha_{ik} \underset{L}{\Rightarrow} \cdots \underset{L}{\Rightarrow} \alpha_{(i+1)k}$$

que no produce ningún símbolo de wB.

Llamemos $v_0, .., v_k$ a los vértices cuyas etiquetas son $\alpha_{ik}, .., \alpha_{(i+1)k}$. Entonces, el subarbol $v_0, .., v_k$ deriva solamente λ. Como cada uno de $\alpha_{ik}, .., \alpha_{(i+1)k}$ es una cadena de símbolos no terminales y son en total $k+1$, entonces, necesariamente hay dos que empiezan con el mismo símbolo. Pero esto contradice que la gramática no es recursiva a izquierda. Entonces nuestra suposición de que el árbol de derivación $\mathcal{A}$ tiene un camino de longitud mayor igual que $k(n+2)$ es imposible.

Sea ℓ el máximo número de símbolos en la parte derecha de una producción de la gramática. La cantidad de nodos del árbol de derivación $\mathcal{A}$ es a lo sumo

$$\ell^{k(n+2)}$$

Por lo tanto, si $A \overset{i}{\underset{L}{\Rightarrow}} wB\alpha$, entonces $i \leq \ell^{k(n+2)}$.

Para finalizar la demostración basta tomar $c = \ell^k$. □

Algoritmos de decisión

Teorema

Hay un algoritmo para decidir si un lenguaje libres de contexto es finito.

Demostración

Sea n la constante del Lema de Pumping. L es finito si y solo si ninguna palabra de longitud entre n y $2n-1$.

(izquierda a derecha) Supongamos L es finito pero tiene 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) Supongamos L no tiene 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 una palabra más corta que w que está en L:
Hay una factorización $w = rxyzs$ con $|xyz| \leq n$, $|xz| \geq 1$ y tenemos que $rx^0yz^0s = rys$ está en L. Dado que $|xz| \geq 1$ y $|xyz| \leq n$

$$|w| - n \leq |rys| = |w| - |xz| \leq |w| - 1$$

Si $|w| = 2n$, $|rys|$ es entre n y $2n-1$ contradiciendo la suposición.
Si $|w| > 2n$, $|rys|$ es estrictamente menor que w. O bien $|rys|$ es entre n y $2n-1$ y llegamos a contradicción, o bien $|rys|$ es mayor o igual que $2n$ contradiciendo que w era la más corta de longitud mayor o igual que $2n$.
□

Algoritmos de decisión

Teorema

Hay un algoritmo para decidir si un lenguaje libre 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.)

¿Qué hace este algoritmo?

Input $G = (V_N, V_T, P, S)$ libre de contexto
$N_0 = \emptyset$
repetir

$$i = i + 1$$
$$N_i = \{A : A \to \alpha \in P, \alpha \in (N_{i-1} \cup T)^*\} \cup N_{i-1}$$

hasta que $N_i = N_{i-1}$
Si $S \in N_i$, output **sí**.

Algoritmos de decisión

Teorema

Hay un algoritmo para decidir la pertenecia de una palabra a un lenguaje libre de contexto.

Demostración: Algoritmo de parsing CYK o Earley (en una próxima clase).

Propiedades de Clausura

Teorema

Los lenguajes libres de contexto están cerrados por unión, concatenación, reversa y clausura de Kleene.

No están cerrados por intersección, diferencia, complemento.

Sin embargo la intersección de libre de contexto con regular es libre de contexto.

Demostración del teorema

union (gramática)

Supongamos $G_1 = (N_1, T_1, P_1, S_1)$ *y* $G_2 = (N_2, T_2, P_2, S_2)$.
Definimos $G = (N_1 \cup N_2 \cup \{S\}, T_1 \cup T_2, P, S)$ *donde*
$P = P_1 \cup P_2 \cup \{S \to S_1 | S_2\}$

concatenacion (gramática)

Supongamos $G_1 = (N_1, T_1, P_1, S_1)$ *y* $G_2 = (N_2, T_2, P_2, S_2)$.
Definimos $G = (N_1 \cup N_2 \cup \{S\}, T_1 \cup T_2, P, S)$ *donde*
$P = P_1 \cup P_2 \cup \{S \to S_1 S_2\}$

clausura Kleene (gramática)

Supongamos $G = (N, T, P, S)$
Definimos $G' = (N \cup \{S'\}, T, P', S')$ *donde*
$P' = P \cup \{S' \to SS' | \lambda\}$

interseccion con lenguaje regular (autómata de pila para lenguaje intersección)

Demostración del teorema, continuación

reversa (gramática invertiendo cuerpo gramática)

Supongamos $G = (V, T, P, S)$ *es libre de contexto,* $L = L(G)$.
Construimos $G' = (V, T, P', S)$ *para* L^R *así:*

Para cada producción $X \to \alpha$ *en* P *ponemos* $X \to \alpha^R$ *en* P'.

Supongamos P *tiene* $S \to uXv$ *y* $X \to \alpha$.
Entonces P' *tiene* $S \to v^R X u^R$ *y* $X \to \alpha^R$.
Luego $S \overset{*}{\underset{G}{\Rightarrow}} u\alpha v$ *y* $S \overset{*}{\underset{G'}{\Rightarrow}} v^R \alpha^R u^R$.
Dado que $v^R \alpha^R u^R = (u\alpha v)^R$, *tenemos* $S \overset{*}{\underset{G'}{\Rightarrow}} (u\alpha v)^R$.

Demostración del teorema, continuación

Los lenguajes libres de contexto no están clausurados por :

intersección

Sean los lenguajes libres de contexto
$L_1 = \{a^i b^j c^j\}$ *y* $L_2 = \{a^i b^i c^j\}$.
Notemos que $L_1 \cap L_2 = \{a^i b^i c^i\}$ *no es libre de contexto.*

complemento

Supongamos que el complemento fuera libre de contexto.
Entonces
$L_1 \cap L_2 = \overline{\overline{L_1} \cup \overline{L_2}}$ *sería libre de contexto.*

diferencia

Si lo fuera entonces $\Sigma^* - L$ *deberia ser libre de contexto*

□

Definición (Gramáticas ambiguas)

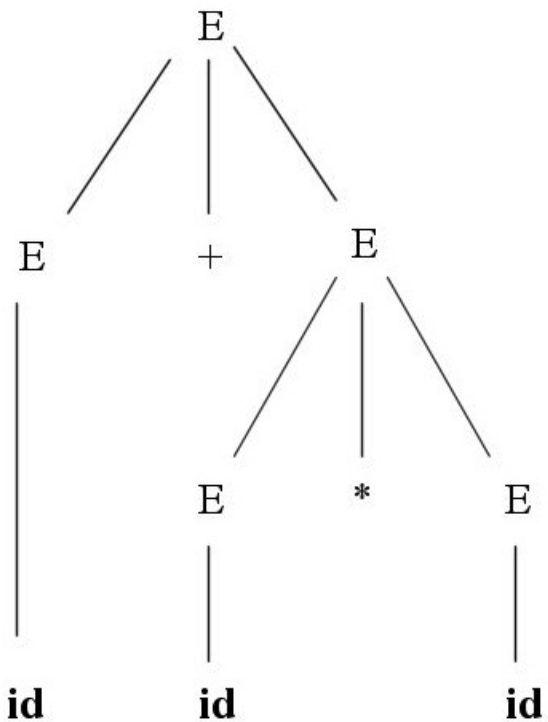
Una gramática libre de contexto G es ambigua si existe $\alpha \in \mathcal{L}(G)$ con más de una derivación más a la izquierda.

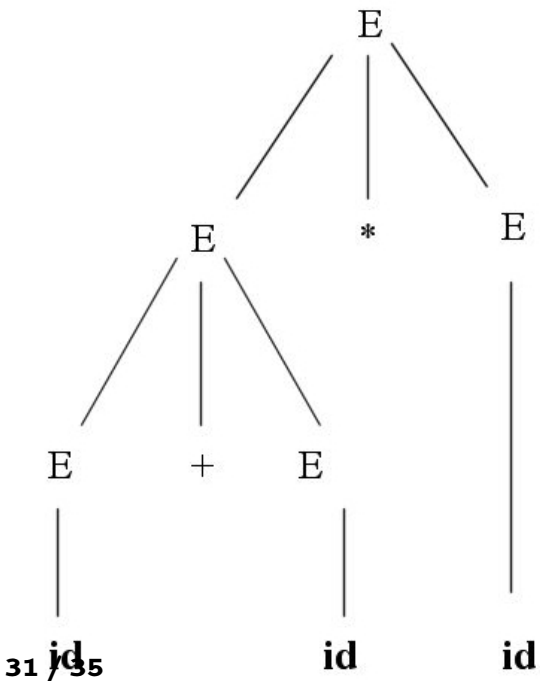
Ejemplo

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

$$P = \{ \begin{array}{l} E \to E + E, \\ E \to E * E, \\ E \to \mathbf{id}, \\ E \to \mathbf{const}\}, \end{array}$$

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





Definición

Un lenguaje libre de contexto es inherentemente ambiguo si solamente admite gramáticas ambiguas.

Ejemplo

Este lenguaje es libre 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 libre 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 libre 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 $(i_k)_{1 \leq k \leq K}$ 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

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

Resumen: decisión de lenguajes libres de contexto

Sea G gramática libre de contexto y sea $L = L(G)$.

Hay algoritmos para :
$L = \emptyset$?
L finito?
L infinito?
$w \in L$? en tiempo cúbico en la longitud de w (algoritmo CYK).

No hay algoritmos para:
L es regular?
G libre de contexto es ambigua?
$L_1 = L_2$?
$L = \Sigma^*$?
$L_1 \subseteq L_2$?
$L_1 \cap L_2 = \emptyset$?

Preguntas

1. Sea L un lenguaje. Si todas las palabras de L validan el Lema de Pumping para lenguajes libres de contexto, ¿Podemos concluir que L es un lenguaje libre contexto?
2. Sea L un lenguaje regular. Demostrar que todas las palabras de L validan el Lema de Pumping para lenguajes libres de contexto.
3. Mostrar que $L = \{a^p : p \text{ es número primo}\}$ no es libre de contexto. Ayuda: . Asumir L es libre 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 libre de contexto es libre 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.