Propiedades de los LLC.

Lema del Bombeo para LIC

Herramienta para demostrar que ciertos lenguajes no son LLC.

Sea L un LLC, entonces existe una constante n tal que si z es cualquier cadena de L de longitud mayor o igual a n ($|z| \geq n$) podemos escribir $z = uvwxy$ con las siguientes condiciones:

1) $|vwx| \leq n$
2) $vx \neq \epsilon$
3) $\forall\, i \geq 0 \quad uv^iwx^iy$ está en L

No se sabe donde cae x y v por tanto se debe demostrar que $\forall v$ y x. los v^i y x^i deben estar en L o no estar en L. Incluyendo $|v|$ ó $|x| = \emptyset$ $\quad v = \epsilon$ y $x = \epsilon$

Si una cadena de L es suficientemente larga, siempre, hay 2 subcadenas cortas lo suficientemente cerca que pueden ser bombeadas el mismo número de veces.

$$S \overset{*}{\Rightarrow} uAy \overset{*}{\Rightarrow} uvAxy \overset{*}{\Rightarrow} uvwxy \overset{*}{\Rightarrow} z$$

$A \overset{*}{\Rightarrow} vAx$ y $A \overset{*}{\Rightarrow} w$ tenemos que

$$A \overset{*}{\Rightarrow} vAx \overset{*}{\Rightarrow} v^2Ax^2 \overset{*}{\Rightarrow} \dots\; v^iAx^i \overset{k}{\Rightarrow} v^iwx^i$$

y $S \overset{*}{\Rightarrow} uv^iwx^iy$. para $i \geq 0$

$S \to uAy$
$A \to vAy \mid w$

Importante:

- Es importante seleccionar un z adecuado. Se puede escoger un z que no funcione al aplicar el lema.
- Es importante cerciorarse de que los casos cubren todas las posibilidades y que en realidad se obtenga una contradicción en cada caso.

(3) $L = \{ww \mid w \text{ en } (a+b)^*\}$

Asumo L como LLC y aplico el lema del bombeo

Escogemos $z = \underbrace{a^n b^n}_{w} \underbrace{a^n b^n}_{w}$, $|vx| > 0$ $|vwx| \leq n$

$z = uvwxy$

Hay mas casos posibles.

(4) $L = \{a^i b^j c^k \mid i < j < k\}$ $\begin{cases} L_1 = \{a^i b^j c^k \mid i < j\} \text{ si es LLC} \\ L_2 = \{a^i b^j c^k \mid j < k\} \text{ si es LLC} \end{cases}$ $L_1 \cap L_2$ no es LLC

(5) $L = \{a^n b^{2n} a^n \mid n \geq 0\}$ $\begin{cases} L_1 = \{a^n b^n, n \geq 0\} \text{ si es LLC} \\ L_2 = \{b^n a^n, n \geq 0\} \text{ si es LLC} \end{cases}$ $L_1 . L_2 \Rightarrow$ LLC

(6) $L = \{xcx \mid x \text{ en } (a+b)^*\}$

(7) $L = \{a^i b^j c^k \mid k = \min(i, j)\}$

(8) $L = \{a^i b^j c^k \mid k = i + j\}$

(9) $L = \{a^n b^{n^2} \mid n > 0\}$

(10) $L = \{a^i b^{i+k} a^k \mid k \neq i\}$

Con el lema del bombeo resultan muchos casos generalmente y es "difícil" ubicar las subcadenas a bombear en z. Existe una variante del lema del Bombeo denominado el lema de Ogden que permites distinguir las posiciones y reducir el número de casos.

Lema de Ogden:

Sea L un LLC entonces hay una constante n, tal que si z en L, $|z| \geq n$ y marcan n o mas posiciones como "distinguibles" podemos escribir $z = uvwxy$ tal que:

1) v y x tienen al menos una posición distinguible
2) vwx tiene a lo sumo n posiciones distinguibles
3) uv^iwx^iy en L $\forall i \geq 0$.

Nota: No hay restricción de la longitud de la cadena vx, sino que deben tener al menos un símbolo distinguible.

Ejemplo:

① $L = \{a^i b^j c^k d^l \mid i=0 \text{ ó } j=k=l\}$

Asumo que L es LLC y aplico L.Ogden

Para $i=0$ $z = b^n c^n d^n$ y ya demostramos por el lema del bombeo que no es LLC.

Para $i \neq 0$ Sea $z = a^k b^n c^n d^n$ y b distinguibles

	a^k	b^n	c^n	d^n
①	v	x		
②		vx		
③		v	x	
④		v		x

En todos los casos ① ②, ③, y ④ al bombear el z sale del conjunto L por tanto es una contradicción y No es un LLC.

② $L = \{a^i b^j c^k \mid i \neq j, j \neq k, i \neq k\}$

$z = a^n b^{n+n!} c^{n+2n!}$ a distinguibles

③ $L = \{a^i b^i c^j \mid i, j \geq 0 \text{ y } j \neq i\}$

$z = a^n b^n c^{n+n!}$ a distinguibles

Asumo a L como LLC, aplico lema de Ogden

$z = a^n b^n c^{n+n!}$

Casos posibles

$a^n \quad b^n \quad c^{n+n!}$

① vx

② $v \quad x$

③ $v \quad\quad x$

① vx en a^+

② v en a^+ y x en b^*

③ v en a^+ y x en c^*

③ $z = uv^iwx^iy$, $i \geq 0$, hay que conseguir un valor de i para sacar a z del lenguaje L.

$i = |v|$, $1 \leq i \leq n$ (ya que vwx tiene a lo sumo n posiciones distinguibles)

i divide a $n!$

Sea q tal que $iq = n!$

Consideramos $uv^{q+1}wx^{q+1}y$

Al bombear $uv^{(q+1)i}wx^{(q+1)i}y \Rightarrow uv^{iq+i}wx^{iq+i}y$

El # a es $\underbrace{(q+1)i}_{v} + \underbrace{(n-i)}_{u} \Rightarrow qi + i + n - i = qi + n$

$n! + n \Rightarrow \#a = \#c$.

① $p = |vx|$

② $p = |v|$

Para demostrar que un lenguaje no es LLC, se supone que L es LLC y se aplican propiedades cerradas en los LLC, si se llega a algo que no es un LLC entonces no se puede haber partido de un LLC.

<u>Teorema:</u> Los LLC son cerrados bajo la unión, concatenación, clausura de Kleene, sustitución, homomorfismo inverso

__teorema:__ Los LLC no son cerrados bajo intersección y complemento.

__teorema__ : Si L es un LLC y R es un conjunto regular, entonces $L \cap R$ es un LLC.

Algoritmos de Decisión de Los LLC

Los Problemas son decidibles si tienen algoritmos.

En los lenguajes Regulares se podía decidir "todo". En los LLC solo se puede decidir:

1. Si el lenguaje es vacío
2. Si L es finito o infinito
3. Si una cadena w está en L

① Si el lenguaje es vacío $G = (V, T, P, S)$ $L = L(G)$

Dada una GLC se aplica el algoritmo para determinar las variables generadoras o derivadoras de cadenas w en T^* (Para eliminar símbolos inútiles). Cuando termina el algoritmo se obtiene un conjunto V' y P' con las variables generadoras y las producciones sin símbolos inútiles. Si S está en el conjunto final de variables V', entonces L es no vacío.

② Si L es finito o infinito

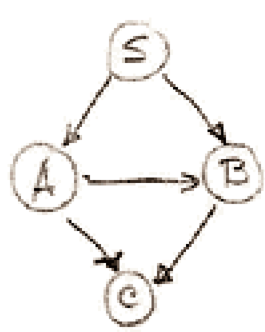
Dada una GLC $G = (V, T, P, S)$ tal que $L = L(G)$.

- Convertimos G a la FNC (sin ε-producciones, producciones unitarias, ni símbolos inútiles, con producciones de la forma $A \to BC$ y $A \to a$).
- Se construye un grafo con un vértice o nodo por cada variable en V. Se agrega un arco entre A y B si $A \to BC$ o $A \to CB$ está en P. Nota: Es un digrafo (grafo dirigido)
- Si el grafo contiene ciclos L es infinito
  Sino L es finito.

también puede hacerse a partir del AP a través de su diagrama de estados??

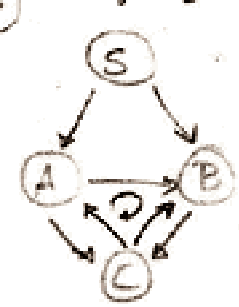
Ejemplo:

Ⓐ $S \to AB$
$A \to BC \mid a$
$B \to CC \mid b$
$C \to a$



No hay ciclos es finito

Ⓑ Agregando $C \to AB$



Hay un ciclo $A \to B \to C$

es infinito

③ Pertenencia

Dada $G=(V,T,P,S)$ y w en T^*, ¿w en $L(G)$?

Obtenemos G en FNG. Si w es de tamaño n $|w|=n$, se necesitan n derivaciones para generar w (si w esta en $L(G)$).

Si hay a lo sumo k producciones por variable entonces tenemos como máximo $k^{|w|}$ posibilidades de derivar w. Es muy costoso computacionalmente.

$$w = a_1\ a_2\ a_3\ \cdots\ a_n$$
$$K\quad K\quad K \qquad k = K^{|w|}$$

Existe un algoritmo menos costoso, basado en la idea de "programación dinámica" que se conoce también como "algoritmo de relleno de tablas" o "tabulación". El algoritmo se conoce por las 3 personas que lo descubrieron de manera independiente (Cocke, Younger, Kasami) y tiene sus iniciales CYK ($O(n^3)$)

Algoritmo CYK

- Dada $G=(V,T,P,S)$ una GLC y w en T^*
- Obtenemos a G en FNC.
- Se construye una tabla, con el eje horizontal = posiciones de la cadena $w = a_1 a_2 \cdots a_n$

n	X_{1n}				
$n-1$	$\vdots$	X_{2n}			
$n-2$	$\vdots$	$\vdots$	X_{3n}		
$\vdots$			$\vdots$		
2	X_{12}	X_{23}	X_{34}		
1	X_{11}	X_{22}	X_{33}	$\cdots$	X_{nn}
	a_1	a_2	a_3	$\cdots$	a_n

$$X_{ij} = \{A \mid A \overset{*}{\Rightarrow} a_i a_{i+1} \cdots a_j\}$$

BASE $i=j$ $(A \to a)$

$$X_{ii} = \{A \mid A \to a_i\}$$

INDUCTIVO $j \neq i$ $(A \to BC)$

$A \overset{*}{\Rightarrow} a_i a_{i+1} \cdots a_j$

$$\left.\begin{array}{l} B \overset{*}{\Rightarrow} a_i a_{i+1} \cdots a_k \\ C \overset{*}{\Rightarrow} a_{k+1} a_{k+2} \cdots a_j \end{array}\right\} k<j \qquad \Rightarrow$$

Para A este X_{ij} debemos encontrar B y C y k tal que:

1) $i \leq k < j$
2) B esta en X_{ik}
3) C esta en $X_{k+1\,j}$
4) $A \to BC$ esta en P.

- Si S pertenece al conjunto X_{1n} entonces $S \overset{*}{\Rightarrow} \omega$, es decir ω esta en L.

Nota :

Para rellenar la tabla, trabajamos fila x fila hacia arriba (comenzamos x el caso base y luego aplicamos el paso inductivo). En otras palabras la fila 1 es para cadenas de longitud 1, la fila 2 para cadenas de longitud 2, asi sucesivamente hasta llegar a la fila n que corresponde a la cadena ω ($|\omega| = n$).

teorema : El algoritmo CYK calcula correctamente X_{ij} para todo i, j, por lo tanto, ω está en $L(G)$ si y solo si S esta en X_{1n}. El tiempo de ejecución del algoritmo es $O(n^3)$.

Ejemplo $G = (V, T, P, S)$ $V = \{A, B, C, S\}$ $T = \{0, 1\}$

$$P = \begin{cases} S \to AB \mid BC \\ A \to BA \mid 0 \\ B \to CC \mid 1 \\ C \to AB \mid 0 \end{cases}$$

Comprobar la pertenencia de la cadena $\omega = 10010$

$\omega = a_1 a_2 a_3 a_4 a_5$

5	$\{S,A,C\}$				
4	–	$\{S,C,A\}$			
3	–	$\{B\}$	$\{B\}$		
2	$\{A,S\}$ X_{12}	$\{B\}$ X_{23}	$\{S,C\}$ X_{34}	$\{A,S\}$ X_{45}	
1	$\{B\}$	$\{A,C\}$	$\{A,C\}$	$\{B\}$	$\{A,C\}$
	1 a_1	0 a_2	0 a_3	1 a_4	0 a_5

fila 1 : X_{ii} caso base

$X_{ii} = \{A \mid A \to a_i\}$

Derivan $1 \to \{B\}$

Derivan $0 \to \{A, C\}$

fila 2 : X_{ij} caso inductivo

$X_{12} = 10$ $(BA, BC) = \{A, S\}$

$X_{23} = 00$ $(AA, AC, CA, CC) = \{B\}$

$X_{34} = 01$ $(AB, CB) = \{S, C\}$

$X_{45} = 10$ $(BA, BC) = \{A, S\}$

fila 3 : X_{ij} caso inductivo

$X_{13} = 100 \to \{\overset{k=1}{X_{11} \cdot X_{23}}\} \cup \{\overset{k=2}{X_{12} \cdot X_{33}}\} = \{B\}.\{B\} \cup \{A,S\}.\{A,C\} = \emptyset$

$X_{24} = 001 \to \{\overset{k=2}{X_{22} \cdot X_{34}}\} \cup \{\overset{k=3}{X_{23} \cdot X_{44}}\} = \{A,C\}\{S,C\} \cup \{B\}.\{B\} = \{B\}$

$X_{35} = 010 \to \{\overset{k=3}{X_{33} \cdot X_{45}}\} \cup \{\overset{k=4}{X_{34} \cdot X_{55}}\} = \{A,C\}.\{A,S\} \cup \{S,C\}.\{A,C\} = \{B\}$

$i \leq K < j$ $2 \leq K < 4$

$X_{iK} \cdot X_{K+1\,j}$

$i \leq k < j$

fila 4 : X_{ij} caso inductivo

$$X_{14} = 1001 \quad \overset{k=1}{\{X_{11}.X_{24}\}} \cup \overset{k=2}{\{X_{12}.X_{34}\}} \cup \overset{k=3}{\{X_{13}.X_{44}\}} =$$

$$= \{B\}.\{B\} \cup \{A,S\}.\{S,C\} \cup \Phi.\{B\} = \Phi$$

$$X_{25} = 0010 \quad \overset{k=2}{\{X_{22}.X_{35}\}} \cup \overset{k=3}{\{X_{23}.X_{45}\}} \cup \overset{k=4}{\{X_{24}.X_{55}\}} =$$

$$\{A,C\}\{B\} \cup \{B\}\{A,S\} \cup \{B\}\{A,C\} = \{S,C,A\}$$

fila 5 : X_{ij} caso inductivo

$$X_{15} = 10010 = \overset{k=1}{\{X_{11}.X_{25}\}} \cup \overset{k=2}{\{X_{12}.X_{35}\}} \cup \overset{k=3}{\{X_{13}.X_{45}\}} \cup \overset{k=4}{\{X_{14}.X_{55}\}}$$

$$= \{B\}.\{S,C,A\} \cup \{A,S\}.\{B\} \cup \Phi.\{A,S\} \cup \Phi.\{A,C\} = \{S,A,C\}$$

Como X_{15} contiene a S entonces w esta en $L(G)$.

Problemas indecidibles sobre LLC

Los Problemas sobre los LLC que no tienen ningún algoritmo son problemas "indecidibles". Algunos de ellos son, dada una $G=(V,T,P,S)$ GLC, donde L es LLC y $L = L(G)$:

- (a) ¿G es ambigua?
- (b) ¿L es inherentemente ambiguo?
- (c) ¿La intersección de 2 LLC es vacio?
- (d) ¿Dos LLC son iguales?
- (e) ¿Dos GLC son equivalentes?
- (f) ¿Es un LLC igual a Σ^*?