INF155 - Informática Teórica

Ayudantía #6

NP-Complete Warriors

Semestre 2022-2

UNIVERSIDAD TECNICA FEDERICO SANTA MARIA
DEPARTAMENTO DE INFORMÁTICA

Forma Normal de Chomsky (CNF)

Definición

Una CFG $G = (N, \Sigma, P, S)$ se dice en *forma normal de Chomsky* (abreviada CNF) si todas sus producciones tienen las formas:

$$\begin{aligned} A &\to BC \qquad && A, B, C \in N \\ A &\to a && A \in N, a \in \Sigma \end{aligned}$$

La forma normal de Chomsky nos sirve para realizar demostraciones con gramáticas, ya que es mucho más sencillo si hay pocos tipos de lados derechos.

Forma Normal de Chomsky (CNF)

Para transformar una gramática en su Forma Normal, se deben realizar los siguientes pasos:

- Eliminar producciones ε: para ello, se agrega la producción como si se hubiese reemplazado el no terminal por ε.
- Eliminar *producciones unitarias*: Son las de la forma $A \rightarrow B$, donde A y B son no-terminales.
- Convertir producciones a CNF: es decir, las producciones pueden tener **exactamente** dos no terminales o **exactamente** un terminal.

CNF - Ejemplo

Consideremos la siguiente gramática:

$$
\begin{aligned}
S &\rightarrow AC \mid BC \mid DE \mid DF \\
A &\rightarrow 0 \mid 0A \mid 0A1 \\
B &\rightarrow 1 \mid B1 \mid 0B1 \\
C &\rightarrow 2 \mid 2C \\
D &\rightarrow 0 \mid 0D \\
E &\rightarrow 1 \mid 1E \mid 1E2 \\
F &\rightarrow 2 \mid F2 \mid 1F2
\end{aligned}
$$

Note que gran parte de las producciones ya se encuentran en CNF, por lo que nuestro trabajo será sencillo.

CNF - Ejemplo

Comenzamos definiendo las siguientes producciones:

$$Z \to 0$$
$$O \to 1$$
$$T \to 2$$

Reescribiendo nuestra gramática, se obtiene lo siguiente:

$$S \to AC \mid BC \mid DE \mid DF$$
$$A \to 0 \mid ZA \mid ZAO$$
$$B \to 1 \mid BO \mid ZBO$$
$$C \to 2 \mid TC$$
$$D \to 0 \mid ZD$$
$$E \to 1 \mid OE \mid OET$$
$$F \to 2 \mid FT \mid OFT$$
$$Z \to 0$$
$$O \to 1$$
$$T \to 2$$

CNF - Ejemplo

Definimos 4 producciones más para transformar las producciones restantes a CNF:

$$X_1 \to AO$$
$$X_2 \to BO$$
$$X_3 \to ET$$
$$X_4 \to FT$$

CNF - Ejemplo

Así, nuestra linda gramática queda en Forma Normal de Chomsky:

$$
\begin{aligned}
S &\rightarrow AC \mid BC \mid DE \mid DF & Z &\rightarrow 0 \\
A &\rightarrow 0 \mid ZA \mid ZX_1 & O &\rightarrow 1 \\
B &\rightarrow 1 \mid BO \mid ZX_2 & T &\rightarrow 2 \\
C &\rightarrow 2 \mid TC & X_1 &\rightarrow AO \\
D &\rightarrow 0 \mid ZD & X_2 &\rightarrow BO \\
E &\rightarrow 1 \mid OE \mid OX_3 & X_3 &\rightarrow ET \\
F &\rightarrow 2 \mid FT \mid OX_4 & X_4 &\rightarrow FT
\end{aligned}
$$

Lema de Bombeo para CFL

Sea $\mathscr{L}$ un lenguaje de contexto libre. Entonces hay una constante $N > 1$ tal que para todo $\sigma \in \mathscr{L}$, si $|\sigma| \geq N$ hay $u, v, w, x, y \in \Sigma^*$ con los que podemos escribir:

$$\sigma = uvwxy$$

donde $|vwx| \leq N$, $vx \neq \varepsilon$, y para todo $k \geq 0$:

$$uv^k wx^k y \in \mathscr{L}$$

Consideraciones importantes

- Al igual que su variante regular, el lema de bombeo sólo puede utilizarse para demostrar que un lenguaje NO es de contexto libre.
- El *substring* $|vwx|$ puede *moverse* a lo largo de σ. Esto generará varios casos a analizar en el lema de bombeo. **Todos los casos deben fallar** (es decir, $uv^kwx^ky \notin \mathscr{L}$) para concluir que no se cumple el lema.
- Todos los casos deben probarse con un **ÚNICO** valor de k.

Lema de Bombeo - Ejemplo clásico

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

Demostración: por contradicción. Supongamos que $\mathscr{L}$ es de contexto libre, por lo que cumple el lema de bombeo.

Lema de Bombeo - Ejemplo clásico

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

Demostración: por contradicción. Supongamos que $\mathscr{L}$ es de contexto libre, por lo que cumple el lema de bombeo. Sea N la constante del lema y $\sigma = a^N b^N c^N$, con $|\sigma| = 3N \geq N$.

Lema de Bombeo - Ejemplo clásico

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

Demostración: por contradicción. Supongamos que $\mathscr{L}$ es de contexto libre, por lo que cumple el lema de bombeo. Sea N la constante del lema y $\sigma = a^N b^N c^N$, con $|\sigma| = 3N \geq N$. El lema nos dice que podemos dividir σ en $uvwxy$, con $|vwx| \leq N$ y $vx \neq \varepsilon$.

Lema de Bombeo - Ejemplo clásico

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

Demostración: por contradicción. Supongamos que $\mathscr{L}$ es de contexto libre, por lo que cumple el lema de bombeo. Sea N la constante del lema y $\sigma = a^N b^N c^N$, con $|\sigma| = 3N \geq N$. El lema nos dice que podemos dividir σ en $uvwxy$, con $|vwx| \leq N$ y $vx \neq \varepsilon$. Dado que la palabra escogida es $a^N b^N c^N$, el *substring* vwx abarcará a lo más 2 tipos de símbolos.

a	a	a	$\cdots$	a	a	a	b	b	b	$\cdots$	b	b	b	c	c	c	$\cdots$	c	c	c
1	2	3	$\cdots$	N-2	N-1	N	N+1	N+2	N+3	$\cdots$	2N-2	2N-1	2N	2N+1	2N+2	2N+3	$\cdots$	3N-2	3N-1	3N

Lema de Bombeo - Ejemplo clásico

Analizamos ***todos*** los casos posibles:

- Si aumento a's y b's, no puedo aumentar las c's.
- Si aumento b's y c's, no puedo aumentar las a's.
- Si aumento las a's, no puedo aumentar las b's ni las c's.
- Si aumento las b's, no puedo aumentar las a's ni las c's.
- Si aumento las c's, no puedo aumentar las b's ni las a's.

Con cualquiera de las opciones, no se cumplirá la condición de pertenencia a $\mathscr{L}$. Luego, no cumple el lema de bombeo, por lo que el lenguaje no es CFL ni regular.

$\therefore \mathscr{L}$ no es CFL.

$\square$

Propiedades de Clausura para CFL

Al igual que en los lenguajes regulares, hay ciertas operaciones que tienen clausura sobre el conjunto de los lenguajes de contexto libre. A continuación haremos un rápido *review*

Sustitución

La idea es exactamente la misma que para los lenguajes regulares; en este caso, reemplazamos cada símbolo terminal por el símbolo inicial de la gramática asociada a dicho símbolo.
Por ejemplo, sea $\mathscr{L}(G)$, con G descrita por las producciones:

$$S \rightarrow aSb \mid ab$$

y sea la sustitución σ dada por las siguientes producciones:

$$a \mapsto \mathscr{L}_a$$
$$b \mapsto \mathscr{L}_b$$

Sustitución

Donde $\mathscr{L}_a$ es generado por la gramática $S_a \to 1S_a1 \mid 00$ y $\mathscr{L}_b$ por la gramática:

$$
\begin{aligned}
S_b &\to 0B \mid 1 \\
B &\to 1A1 \mid 0 \\
A &\to 0B \mid 1
\end{aligned}
$$

Entonces, $\sigma(\mathscr{L}(G))$ es

$$
\begin{aligned}
S &\to S_aSS_b \mid S_aS_b \\
S_a &\to 1S_a1 \mid 00 \\
S_b &\to 0B \mid 1 \\
A &\to 0B \mid 1 \\
B &\to 1A1 \mid 0
\end{aligned}
$$

Propiedades de clausura de los CFL

Los lenguajes de contexto libre son cerrados respecto a:

- Sustitución
- Homomorfismo
- Homomorfismo inverso

Los CFL también son cerrados respecto a:

- Estrella de Kleene
- Unión
- Concatenación
- Operaciones regulares
- Intersección con lenguajes regulares

Propiedades de no-clausura

- Los CFL **no son cerrados** respecto al complemento.
  Note que esto NO significa que en casos particulares el complemento de un CFL no sea un CFL.
- Los CFL **no son cerrados** respecto a la intersección.

PDA - Pushdown Automata

Un PDA (o autómata de pila $M = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$ consta de:

Q : Conjunto finito de estados

Σ : Alfabeto de entrada

Γ : Alfabeto de *stack*

δ : Función de transición : $Q \times (\Sigma \cup \{\varepsilon\}) \times \Gamma \rightarrow \underbrace{2^{Q \times \Gamma^*}}_{\text{subconjuntos finitos}}$

q_0 : Estado inicial, $q_0 \in Q$.

Z_0 : *Stack* inicial, $Z_0 \in \Gamma$

F : Estados finales, $F \subseteq Q$.

PDA: notación de transiciones

Los arcos corresponden a las transiciones del PDA, bajo la notación:

$$x, A / \gamma$$

donde x es un símbolo de entrada, A es un símbolo de *stack*, y γ es una palabra de *stack*.
La expresión anterior se lee como: *si leo x y en el tope del stack hay una A, realizo una transición al estado indicado (puede ser el mismo) reemplazando el tope del stack A por γ.*

Aceptación

¿Cómo aceptamos una palabra en un PDA? Hay básicamente 2 opciones:

- Stack vacío (por convención, se define $F = \varnothing$)
- Estado final

Observación: en general, los lenguajes aceptados por estado final y por *stack vacío* son diferentes.

¿Qué tipo de autómata es un PDA?

Claramente, un PDA es de naturaleza **no determinista** (como el NFA).

¿Puede haber un PDA determinista? Sí, solo se deben considerar las siguientes situaciones:

- Dada una entrada, un tope de *stack* y un estado, el PDA puede tener a lo más una movida posible.
- El PDA puede hacer movimientos consumiendo ε o $a \in \Sigma$. Entonces, debemos limitar a que pueda moverse sólo con uno de los símbolos de entrada, no con ambos.

Ejemplo

Construya un PDA que reconozca bifurcaciones if-else. Para ello, asuma que cada **if** tendrá un **else** que lo cerrará, y que no existen sentencias **else if/elif**.
También puede simplificar **if** y **else** con símbolos que los representen.
Ejemplo: nuestro PDA reconoce *if if else if else else if else* pero no *else if*

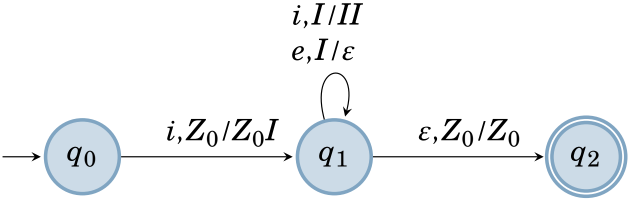
Ejemplo: Solución

Sea $P = (Q, \Sigma, \Gamma, \delta, q_0, z, F)$ nuestro PDA buscado. En este caso será un PDA por estado final

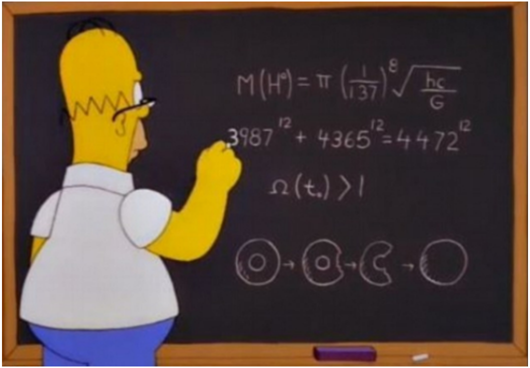
- $Q = \{q_0, q_1, q_2\}$
- $\Sigma = \{\varepsilon, i, e\}$ donde ***i*** hace referencia a **if** y ***e*** a **else**
- $\Gamma = \{Z_0, I, \varepsilon\}$
- $F = \{q_2\}$

Ejemplo: Solución

La representación visual de nuestro PDA viene dada por el siguiente autómata:



Ejercicios


$M(H^0) = \pi \left(\frac{1}{137}\right)^8 \sqrt{\frac{hc}{G}}$
$3987^{12} + 4365^{12} = 4472^{12}$
$\Omega(t_0) > 1$

Ejercicio 1

Escriba la siguiente gramática en CFN (*Chomsky Normal Form*):

$$
\begin{aligned}
S &\rightarrow XY \\
X &\rightarrow abb|aXb|\varepsilon \\
Y &\rightarrow c|cY
\end{aligned}
$$

Solución Ejercicio 1

Notemos que la gramática dada no es de contexto libre, ya que contiene producciones ε. Luego, el primer paso es eliminarlas para obtener una CFG:

$$S \Rightarrow XY|Y$$
$$X \Rightarrow abb|aXb|ab$$
$$Y \Rightarrow c|cY$$

El siguiente paso es eliminar las producciones unitarias $A \Rightarrow B$, obteniendo:

$$S \Rightarrow XY|c|cY$$
$$X \Rightarrow abb|aXb|ab$$
$$Y \Rightarrow c|cY$$

Solución Ejercicio 1

Ahora debemos transformar las producciones a la forma $A \Rightarrow a$ o $A \Rightarrow BC$.

- Definimos $X_1 \Rightarrow bb$ y $X_2 \Rightarrow Xb$

$$\begin{aligned}
S &\Rightarrow XY|c|cY \\
X &\Rightarrow aX_1|aX_2|ab \\
Y &\Rightarrow c|cY \\
X_1 &\Rightarrow bb \\
X_2 &\Rightarrow Xb
\end{aligned}$$

Solución Ejercicio 1

- Definimos $B \Rightarrow b$:

$$
\begin{aligned}
S &\Rightarrow XY|c|cY \\
X &\Rightarrow aX_1|aX_2|aB \\
Y &\Rightarrow c|cY \\
X_1 &\Rightarrow BB \\
X_2 &\Rightarrow XB \\
B &\Rightarrow b
\end{aligned}
$$

Solución Ejercicio 1

- Definimos $A \Rightarrow a$:

$$
\begin{aligned}
S &\Rightarrow XY|c|cY \\
X &\Rightarrow AX_1|AX_2|AB \\
Y &\Rightarrow c|cY \\
X_1 &\Rightarrow BB \\
X_2 &\Rightarrow XB \\
B &\Rightarrow b \\
A &\Rightarrow a
\end{aligned}
$$

Solución Ejercicio 1

- Definimos $C \Rightarrow c$:

$$
\begin{aligned}
S &\Rightarrow XY|c|CY \\
X &\Rightarrow AX_1|AX_2|AB \\
Y &\Rightarrow c|CY \\
X_1 &\Rightarrow BB \\
X_2 &\Rightarrow XB \\
B &\Rightarrow b \\
A &\Rightarrow a \\
C &\Rightarrow c
\end{aligned}
$$

Y así obtuvimos la gramática en CFN.

Ejercicio 2

Dada la siguiente gramática G:

$$
\begin{aligned}
S &\rightarrow XT \\
X &\rightarrow 1X2Z|12L \\
Z2 &\rightarrow 2Z \\
L2 &\rightarrow 2L \\
ZT &\rightarrow T3 \\
LT &\rightarrow 3
\end{aligned}
$$

¿El lenguaje descrito es de contexto libre? ¿Es regular? Demuestre su afirmación utilizando lema de bombeo.

Solución Ejercicio 2

- La idea es ir desarrollando y observar el patrón que se va generando:

$$\begin{aligned} XT \\ 1X2ZT \\ 112L2ZT \\ 112L2T3 \\ 1122LT3 \\ 112233 \end{aligned}$$

Luego, es claro que el lenguaje es $\mathscr{L}(G) = \{1^n 2^n 3^n : n \geq 1\}$

Solución Ejercicio 2

- Suponemos que $\mathscr{L}$ es un CFL, por lo que cumple el lema de bombeo para CFL. Sea N la constante del lema. Elegimos $\sigma \in \mathscr{L}$ tal que:
  - $\sigma = 1^N 2^N 3^N$
  - $|\sigma| = 3N \geq N$

  Por el lema de bombeo para CFL, hay $u, v, w, x, y \in \Sigma^*$ tales que:
  - $|vwx| \leq N$
  - $vx \neq \varepsilon$

  de modo que, para todo $k \geq 0$, $uv^k wx^k y \in \mathscr{L}$.

Solución Ejercicio 2

Como $|vwx| \leq N$, dicho *substring* puede estar formado a lo más por dos tipos de símbolos. Escogemos $k = 2$ y notamos que:

- Si aumento 1's y 2's, no puedo aumentar los 3's.
- Si aumento 2's y 3's, no puedo aumentar los 1's.
- Si aumento los 1's, no puedo aumentar los 2's ni los 3's.
- Si aumento los 2's, no puedo aumentar los 1's ni los 3's.
- Si aumento los 3's, no puedo aumentar los 2's ni los 1's.

Con cualquiera de las opciones, no se cumplirá la condición de pertenencia a $\mathscr{L}$. Luego, no cumple el lema de bombeo, por lo que el lenguaje no es CFL ni regular.

Ejercicio 3

Se define la operación Sufijo del lenguaje $\mathscr{L}$ sobre el alfabeto Σ mediante:

$$\textsc{Sufijo} = \{\beta : \exists \alpha \in \Sigma^*, \alpha\beta \in \mathscr{L}\}$$

Demuestre que los Lenguajes de Contexto Libre son cerrados respecto de Sufijo.

Solución Ejercicio 3

La forma más sencilla de demostrar lo solicitado es con propiedades de clausura. Sabemos que los CFL son cerrados respecto a intersección con lenguajes regulares, sustituciones y homomorfismos. Con esto en mente, el plan de ataque es el siguiente:

- Marcar todos los símbolos de la palabra con primas y segundas. Para ello, definimos la sustitución σ para $a \in \Sigma$:

  $$\sigma(a) = \{a', a''\}$$

  Al ser lenguajes finitos, son CFL.

Solución Ejercicio 3

- Separar las primas de las segundas. Para ello, intersectamos con un lenguaje regular representado por la siguiente RE:

$$\mathscr{L}_i = \mathscr{L}((a' \mid b' \mid \ldots)^* (a'' \mid b'' \mid \ldots)^*)$$

  Los lenguajes regulares son CFL.

- Ahora, solo queremos quedarnos con los símbolos con segundas y eliminar aquellos con primas. Para ello, definimos el homomorfismo h para $a \in \Sigma$:

$$h(a') = \varepsilon$$
$$h(a'') = a$$

  Los CFL son cerrados respecto a homomorfismos.

Solución Ejercicio 3

De esta forma, Sufijo se define como:

$$\textsc{Sufijo}(\mathscr{L}) = h(\sigma(\mathscr{L}) \cap \mathscr{L}_i)$$

Dado que solo utilizamos propiedades de clausura, los CFL son cerrados respecto de Sufijo.

$\square$

Ejercicio 4

Sea $\mathscr{L} = \{a^n b^n : n \geq 1 \wedge n \neq 100\}$ un lenguaje sobre $\Sigma = \{a, b\}$. Demuestre, usando propiedades de clausura, que el lenguaje $\mathscr{L}$ es de contexto libre.

Solución Ejercicio 4

Para demostrar que $\mathscr{L}$ es de contexto libre, primero definamos $\mathscr{L}_1 = \{a^{100}b^{100}\}$ y $\mathscr{L}_2 = \{a^n b^n : n \geq 1\}$. Como ya debe saber, el lenguaje $\mathscr{L}_1$ es regular y $\mathscr{L}_2$ es de contexto libre (las producciones son $S \rightarrow aSb|ab$).

Notar que $\mathscr{L} = \overline{\mathscr{L}_1} \cap \mathscr{L}_2$, donde por *propiedades de clausura* de lenguajes regulares $\overline{\mathscr{L}_1}$ es **regular**. Además, por *propiedades de clausura* de los lenguajes de contexto libre, la intersección entre un lenguaje de contexto libre y uno regular resultan en un lenguaje de contexto libre, por lo tanto, $\mathscr{L}$ es de **contexto libre**.

INF155 - Informática Teórica

Ayudantía #6

NP-Complete Warriors

Semestre 2022-2

UNIVERSIDAD TECNICA FEDERICO SANTA MARIA

DEPARTAMENTO DE INFORMÁTICA