Autómatas y Lenguajes Formales 2016-1

Maestría en Ciencia e Ingeniería de la Computación UNAM

Tema 9: Propiedades de cerradura y normalización de gramáticas libres de contexto

Dr. Favio Ezequiel Miranda Perea

favio@ciencias.unam.mx

Facultad de Ciencias UNAM

29 de enero de 2020

Propiedades de cerradura

Lenguaje Libres del Contexto

La clase de los lenguajes libres del contexto es cerrada bajo las siguientes operaciones:

- Unión: si L_1, L_2 son lenguajes libres del contexto entonces $L_1 \cup L_2$ es un lenguaje libre del contexto.

Propiedades de cerradura

Lenguaje Libres del Contexto

La clase de los lenguajes libres del contexto es cerrada bajo las siguientes operaciones:

- Unión: si L_1, L_2 son lenguajes libres del contexto entonces $L_1 \cup L_2$ es un lenguaje libre del contexto.

- Concatenación: si L_1, L_2 son lenguajes libres del contexto entonces $L_1 L_2$ es un lenguaje libre del contexto.

Propiedades de cerradura

Lenguaje Libres del Contexto

La clase de los lenguajes libres del contexto es cerrada bajo las siguientes operaciones:

- Unión: si L_1, L_2 son lenguajes libres del contexto entonces $L_1 \cup L_2$ es un lenguaje libre del contexto.
- Concatenación: si L_1, L_2 son lenguajes libres del contexto entonces $L_1 L_2$ es un lenguaje libre del contexto.
- Estrella de Kleene:si L_1 es un lenguaje libres del contexto entonces $L_1^\star$ es un lenguaje libre del contexto.

Cerradura bajo la unión

Lenguajes Libres del Contexto

Si $G_1 = \langle V_1, T, S_1, P_1 \rangle$, $G_2 = \langle V_2, T, S_2, P_2 \rangle$ son GLC con $L_1 = L(G_1)$, $L_2 = L(G_2)$ entonces $L_1 \cup L_2 = L(G)$ donde G es la gramática

$$G = \langle V_1 \cup V_2 \cup \{S\}, T, S, P \rangle$$

y P está dado por $P_1 \cup P_2$ mas las producciones:

$$S \to S_1, \quad S \to S_2$$

Cerradura bajo la concatenación

Lenguajes Libres del Contexto

Si $G_1 = \langle V_1, T, S_1, P_1 \rangle$, $G_2 = \langle V_2, T, S_2, P_2 \rangle$ son GLC con $L_1 = L(G_1)$, $L_2 = L(G_2)$ entonces $L_1 L_2 = L(G)$ donde G es la gramática

$$G = \langle V_1 \cup V_2 \cup \{S\}, T, S, P \rangle$$

y P está dado por $P_1 \cup P_2$ mas la producción:

$$S \to S_1 S_2$$

Cerradura bajo la estrella de Kleene

Lenguajes Libres del Contexto

Si $G_1 = \langle V_1, T, S_1, P_1 \rangle$ es una GLC con $L_1 = L(G_1)$ entonces $L_1^\star = L(G)$ donde G es la gramática

$$G = \langle V_1 \cup \{S\}, T, S, P \rangle$$

y P está dado por P_1 mas la producciones:

$$S \to S_1 S_1 \quad S \to \varepsilon$$

Propiedades de cerradura inválidas

Lenguajes libres del contexto

En general las siguientes propiedades no son válidas para lenguajes libres del contexto.

- Cerradura bajo la intersección.

Propiedades de cerradura inválidas

Lenguajes libres del contexto

En general las siguientes propiedades no son válidas para lenguajes libres del contexto.

- Cerradura bajo la intersección.
- Cerradura bajo el complemento.

Propiedades de cerradura inválidas

Lenguajes libres del contexto

En general las siguientes propiedades no son válidas para lenguajes libres del contexto.

- Cerradura bajo la intersección.
- Cerradura bajo el complemento.
- Cerradura bajo la diferencia.

Propiedades de cerradura inválidas

Intersección

- La intersección de dos LIC puede ser un lenguaje que no es libre del contexto.

Propiedades de cerradura inválidas

Intersección

- La intersección de dos LIC puede ser un lenguaje que no es libre del contexto.
- $L_1 = \{a^i b^i c^j \mid i, j \geq 1\}$ es libre del contexto:

$$S \to AB,\ A \to aAb \mid ab,\ B \to cC \mid c$$

Propiedades de cerradura inválidas

Intersección

- La intersección de dos LIC puede ser un lenguaje que no es libre del contexto.
- $L_1 = \{a^i b^i c^j \mid i, j \geq 1\}$ es libre del contexto:

$$S \to AB,\ A \to aAb \mid ab,\ B \to cC \mid c$$

- $L_2 = \{a^i b^j c^j \mid i, j \geq 1\}$ es libre del contexto:

$$S \to AB,\ A \to aA \mid a,\ B \to bBc \mid bc$$

Propiedades de cerradura inválidas

Intersección

- La intersección de dos LIC puede ser un lenguaje que no es libre del contexto.
- $L_1 = \{a^i b^i c^j \mid i, j \geq 1\}$ es libre del contexto:

$$S \to AB,\ A \to aAb \mid ab,\ B \to cC \mid c$$

- $L_2 = \{a^i b^j c^j \mid i, j \geq 1\}$ es libre del contexto:

$$S \to AB,\ A \to aA \mid a,\ B \to bBc \mid bc$$

- $L_1 \cap L_2 = \{a^i b^i c^i \mid i \geq 1\}$ no es independiente del contexto.

Propiedades de cerradura inválidas

Complemento y diferencia

- Si el complemento de un LLC L, $\bar{L}$ fuera también libre del contexto entonces la intersección también lo sería pues

$$L_1 \cap L_2 = \overline{\overline{L_1} \cup \overline{L_2}}$$

Propiedades de cerradura inválidas

Complemento y diferencia

- Si el complemento de un LLC L, $\bar{L}$ fuera también libre del contexto entonces la intersección también lo sería pues

$$L_1 \cap L_2 = \overline{\overline{L_1} \cup \overline{L_2}}$$

- Si la diferencia fuera un LLC, entonces también lo sería el complemento pues

$$\bar{L} = \Sigma^\star - L$$

Introducción

Normalización de gramáticas

- La normalización consiste en transformar todas las producciones de una gramática de manera que tengan cierta forma sintáctica en particular.

Introducción

Normalización de gramáticas

- La normalización consiste en transformar todas las producciones de una gramática de manera que tengan cierta forma sintáctica en particular.
- La normalización de gramáticas libres de contexto es útil para homogeneizar la forma de las producciones así como para optimizar los procesos de derivación de cadenas.

Introducción

Normalización de gramáticas

- La normalización consiste en transformar todas las producciones de una gramática de manera que tengan cierta forma sintáctica en particular.
- La normalización de gramáticas libres de contexto es útil para homogeneizar la forma de las producciones así como para optimizar los procesos de derivación de cadenas.
- Con las formas normales se facilita la solución al problema de la pertenencia.

Problema de la pertenencia

Normalización de gramáticas

- Dada una gramática G y una palabra u, ¿Se cumple $u \in L(G)$? Es decir, pertenece u al lenguaje generado por G.

Problema de la pertenencia

Normalización de gramáticas

- Dada una gramática G y una palabra u, ¿Se cumple $u \in L(G)$? Es decir, pertenece u al lenguaje generado por G.
- Si la palabra u es generada por G la construcción de un árbol de derivación terminará eventualmente.

Problema de la pertenencia

Normalización de gramáticas

- Dada una gramática G y una palabra u, ¿Se cumple $u \in L(G)$? Es decir, pertenece u al lenguaje generado por G.
- Si la palabra u es generada por G la construcción de un árbol de derivación terminará eventualmente.
- En caso contrario no podemos saber cuando parar en la construcción del árbol.

Variables Inútiles

Definición

- Una variable A es accesible o alcanzable si existen $u, v \in (V \cup T)^\star$ tales que $S \rightarrow^\star uAv$. Obsérvese que según esta definición S es alcanzable.

Variables Inútiles

Definición

- Una variable A es accesible o alcanzable si existen $u, v \in (V \cup T)^*$ tales que $S \to^* uAv$. Obsérvese que según esta definición S es alcanzable.

- Una variable A es productiva o terminable si existe $w \in T^*$ tal que $A \to^* w$. En particular si $A \to \varepsilon$ es una producción entonces A es productiva.

Variables Inútiles

Definición

- Una variable A es accesible o alcanzable si existen $u, v \in (V \cup T)^*$ tales que $S \to^* uAv$. Obsérvese que según esta definición S es alcanzable.
- Una variable A es productiva o terminable si existe $w \in T^*$ tal que $A \to^* w$. En particular si $A \to \varepsilon$ es una producción entonces A es productiva.
- Una variable A es inútil si no es alcanzable o no es productiva.

Algoritmo para hallar variables productivas

Transformaciones en GLC

$$Prod := \{A \in V \mid A \to w \ \in P,\ w \in T^{\star}\}$$

Algoritmo para hallar variables productivas

Transformaciones en GLC

$$Prod := \{A \in V \mid A \to w \ \in P,\ w \in T^{\star}\}$$

repetir

$$Prod := Prod \cup \{A \in V \mid A \to w,\ w \in (T \cup Prod)^{\star}\}$$

Algoritmo para hallar variables productivas

Transformaciones en GLC

$Prod := \{A \in V \mid A \to w \ \in P,\ w \in T^\star\}$

repetir

$$Prod := Prod \cup \{A \in V \mid A \to w,\ w \in (T \cup Prod)^\star\}$$

hasta que no se añaden nuevas variables a $Prod$.

Ejemplo

Variables productivas

$$
\begin{aligned}
S &\to ACD \mid bBd \mid ab \\
A &\to aB \mid aA \mid C \\
B &\to aDS \mid aB \\
C &\to aCS \mid CB \mid CC \\
D &\to bD \mid ba \\
E &\to AB \mid aDb
\end{aligned}
$$

Ejemplo

Variables productivas

$$
\begin{aligned}
S &\to ACD \mid bBd \mid ab \\
A &\to aB \mid aA \mid C \\
B &\to aDS \mid aB \\
C &\to aCS \mid CB \mid CC \\
D &\to bD \mid ba \\
E &\to AB \mid aDb
\end{aligned}
$$

$$Prod = \{S, D, B, E, A\}$$

Ejemplo

Eliminación de variables improductivas

C es la única variable improductiva, se elimina junto con todas las reglas donde figure:

$$\begin{aligned} S &\to bBd \mid ab \\ A &\to aB \mid aA \\ B &\to aDS \mid aB \\ D &\to bD \mid ba \\ E &\to AB \mid aDb \end{aligned}$$

Algoritmo para hallar variables accesibles

Transformaciones en GLC

$Acc := \{S\}$

Algoritmo para hallar variables accesibles

Transformaciones en GLC

$Acc := \{S\}$

repetir

$$Acc := Acc \cup \{A \in V \mid \exists\ B \to uAv \in P,\ B \in Acc,\ u, v \in (V \cup T)^*\}$$

Algoritmo para hallar variables accesibles

Transformaciones en GLC

$Acc := \{S\}$

repetir

$$Acc := Acc \cup \{A \in V \mid \exists\ B \to uAv \in P,\ B \in Acc,\ u, v \in (V \cup T)^*\}$$

hasta que no se añaden nuevas variables a Acc.

Ejemplo

Variables accesibles

$$
\begin{aligned}
S &\to aS \mid AaB \mid ACS \\
A &\to aS \mid AaB \mid AC \\
B &\to bB \mid DB \mid BB \\
C &\to aDa \mid ABD \mid ab \\
D &\to aD \mid DD \mid ab \\
E &\to FF \mid aa \\
F &\to aE \mid EF
\end{aligned}
$$

Ejemplo

Variables accesibles

$$
\begin{aligned}
S &\to aS \mid AaB \mid ACS \\
A &\to aS \mid AaB \mid AC \\
B &\to bB \mid DB \mid BB \\
C &\to aDa \mid ABD \mid ab \\
D &\to aD \mid DD \mid ab \\
E &\to FF \mid aa \\
F &\to aE \mid EF
\end{aligned}
$$

$$Acc = \{S, A, B, C, D\}$$

Ejemplo

Eliminación de variables inaccesibles

E, F son variables inaccesibles, se eliminan junto con todas las reglas donde figuren:

$$\begin{aligned} S &\to aS \mid AaB \mid ACS \\ A &\to aS \mid AaB \mid AC \\ B &\to bB \mid DB \mid BB \\ C &\to aDa \mid ABD \mid ab \\ D &\to aD \mid DD \mid ab \end{aligned}$$

Eliminación de variables inútiles

Transformaciones en GLC

Para eliminar variables inútiles se aplican los dos algoritmos anteriores en el siguiente orden:

Eliminación de variables inútiles

Transformaciones en GLC

Para eliminar variables inútiles se aplican los dos algoritmos anteriores en el siguiente orden:

- Eliminar variables no productivas.

Eliminación de variables inútiles

Transformaciones en GLC

Para eliminar variables inútiles se aplican los dos algoritmos anteriores en el siguiente orden:

- Eliminar variables no productivas.
- Eliminar variables no accesibles.

Eliminación de variables inútiles

Importancia del orden de los algoritmos

Si se aplican los algoritmos en orden inverso el resultado puede ser una gramática que aún contenga variable inútiles.

Eliminación de variables inútiles

Importancia del orden de los algoritmos

Si se aplican los algoritmos en orden inverso el resultado puede ser una gramática que aún contenga variable inútiles.

$$S \to a \mid AB, \quad A \to aA \mid \varepsilon$$

Eliminación de variables inútiles

Importancia del orden de los algoritmos

Si se aplican los algoritmos en orden inverso el resultado puede ser una gramática que aún contenga variable inútiles.

$$S \to a \mid AB, \quad A \to aA \mid \varepsilon$$

Al eliminar primero las variables no accesibles se obtiene la misma gramática, al ser todas las variables accesibles.

Eliminación de variables inútiles

Importancia del orden de los algoritmos

Si se aplican los algoritmos en orden inverso el resultado puede ser una gramática que aún contenga variable inútiles.

$$S \to a \mid AB, \quad A \to aA \mid \varepsilon$$

Al eliminar primero las variables no accesibles se obtiene la misma gramática, al ser todas las variables accesibles.Posteriormente al eliminar variables improductivas resulta

$$S \to a, \; A \to aA \mid \varepsilon$$

Eliminación de variables inútiles

Importancia del orden de los algoritmos

Si se aplican los algoritmos en orden inverso el resultado puede ser una gramática que aún contenga variable inútiles.

$$S \to a \mid AB, \quad A \to aA \mid \varepsilon$$

Al eliminar primero las variables no accesibles se obtiene la misma gramática, al ser todas las variables accesibles.Posteriormente al eliminar variables improductivas resulta

$$S \to a, \; A \to aA \mid \varepsilon$$

y claramente A es inútil por ser inaccesible.

Eliminación de ε-producciones

Transformaciones en GLC

- Una producción de la forma $A \to \varepsilon$ es una ε-producción

Eliminación de ε-producciones

Transformaciones en GLC

- Una producción de la forma $A \to \varepsilon$ es una ε-producción
- Una variable A se llama anulable si $A \to^{\star} \varepsilon$.

Algoritmo para hallar variables anulables

Transformaciones en GLC

$$Anul := \{A \in V \mid A \to \varepsilon \ \in P\}$$

Algoritmo para hallar variables anulables

Transformaciones en GLC

$$Anul := \{A \in V \mid A \to \varepsilon \in P\}$$

repetir

$$Anul := Anul \cup \{A \in V \mid \exists\ A \to w \in P,\ \ w \in Anul^{\star}\}$$

Algoritmo para hallar variables anulables

Transformaciones en GLC

$$Anul := \{A \in V \mid A \to \varepsilon \in P\}$$

repetir

$$Anul := Anul \cup \{A \in V \mid \exists\ A \to w \in P,\ \ w \in Anul^{\star}\}$$

hasta que no se añaden nuevas variables a *Anul*.

Eliminación de ε-producciones

Transformaciones en GLC

- Eliminar todas las ε-producciones.

Eliminación de ε-producciones

Transformaciones en GLC

- Eliminar todas las ε-producciones.
- Para cada producción de la forma $A \to w_1 \ldots w_n$ agregar las producciones $A \to v_1 \ldots v_n$ donde:

Eliminación de ε-producciones

Transformaciones en GLC

- Eliminar todas las ε-producciones.
- Para cada producción de la forma $A \to w_1 \ldots w_n$ agregar las producciones $A \to v_1 \ldots v_n$ donde:
  - $v_i = w_i$ si $w_i \notin \mathit{Anul}$.

Eliminación de ε-producciones

Transformaciones en GLC

- Eliminar todas las ε-producciones.
- Para cada producción de la forma $A \to w_1 \ldots w_n$ agregar las producciones $A \to v_1 \ldots v_n$ donde:
  - $v_i = w_i$ si $w_i \notin Anul$.
  - $v_i = w_i$ ó $v_i = \varepsilon$ si $w_i \in Anul$.

Eliminación de ε-producciones

Transformaciones en GLC

- Eliminar todas las ε-producciones.
- Para cada producción de la forma $A \to w_1 \ldots w_n$ agregar las producciones $A \to v_1 \ldots v_n$ donde:
  - $v_i = w_i$ si $w_i \notin Anul$.
  - $v_i = w_i$ ó $v_i = \varepsilon$ si $w_i \in Anul$.
  - Verificando que no se anulen todos los v_i al mismo tiempo.

Ejemplo

Eliminación de ε-producciones

$$
\begin{aligned}
S &\to AB \mid ACA \mid ab \\
A &\to aAa \mid B \mid CD \\
B &\to bB \mid bA \\
C &\to cC \mid \varepsilon \\
D &\to aDc \mid CC \mid ABb
\end{aligned}
$$

Ejemplo

Eliminación de ε-producciones

$$
\begin{aligned}
S &\to AB \mid ACA \mid ab \\
A &\to aAa \mid B \mid CD \\
B &\to bB \mid bA \\
C &\to cC \mid \varepsilon \\
D &\to aDc \mid CC \mid ABb
\end{aligned}
$$

$$Anul = \{C, D, A, S\}$$

Ejemplo

Eliminación de ε-producciones

Al eliminar la producción $C \to \varepsilon$ se obtiene la siguiente gramática:

$$
\begin{aligned}
S &\to AB \mid ACA \mid ab \mid B \mid CA \mid AA \mid AC \mid A \mid C \mid \varepsilon \\
A &\to aAa \mid B \mid CD \mid aa \mid C \mid D \\
B &\to bB \mid bA \mid b \\
C &\to cC \mid c \\
D &\to aDc \mid CC \mid ABb \mid ac \mid C \mid Bb
\end{aligned}
$$

Acerca de la palabra vacía

Transformaciones en GLC

- Si originalmente se tenia $\varepsilon \in L(G)$ la eliminación de ε producciones genera una gramática que no genera a ε.

Acerca de la palabra vacía

Transformaciones en GLC

- Si originalmente se tenia $\varepsilon \in L(G)$ la eliminación de ε producciones genera una gramática que no genera a ε.
- Es posible saber si se pierde la palabra vacía al eliminar ε-producciones verificando si $S \in \mathit{Anul}$.

Acerca de la palabra vacía

Transformaciones en GLC

- Si originalmente se tenia $\varepsilon \in L(G)$ la eliminación de ε producciones genera una gramática que no genera a ε.
- Es posible saber si se pierde la palabra vacía al eliminar ε-producciones verificando si $S \in \mathit{Anul}$.
- Si se quiere recuperar a ε debe agregarse un nuevo símbolo inicial S' y las producciones $S' \to S$ y $S' \to \varepsilon$.

Acerca de la palabra vacía

Transformaciones en GLC

- Si originalmente se tenia $\varepsilon \in L(G)$ la eliminación de ε producciones genera una gramática que no genera a ε.
- Es posible saber si se pierde la palabra vacía al eliminar ε-producciones verificando si $S \in Anul$.
- Si se quiere recuperar a ε debe agregarse un nuevo símbolo inicial S' y las producciones $S' \to S$ y $S' \to \varepsilon$.
- $S' \to \varepsilon$ es la única ε-producción permitida.

Eliminación de producciones unitarias

Transformaciones en GLC

- Una producción de la forma $A \to B$ donde A y B son ambas variables se llama producción unitaria.

Eliminación de producciones unitarias

Transformaciones en GLC

- Una producción de la forma $A \to B$ donde A y B son ambas variables se llama producción unitaria.
- El conjunto unitario de A se define como sigue:

$$Unit(A) = \{B \in V \mid A \to^{\star} B \text{ usando sólo prod. unitarias } \}$$

Eliminación de producciones unitarias

Transformaciones en GLC

- Una producción de la forma $A \to B$ donde A y B son ambas variables se llama producción unitaria.
- El conjunto unitario de A se define como sigue:

$$Unit(A) = \{B \in V \mid A \to^{\star} B \text{ usando sólo prod. unitarias } \}$$

- Por definición se tiene $A \in Unit(A)$.

Algoritmo para hallar el conjunto $Unit(A)$

Transformaciones en GLC

$$Unit(A) := \{A\}$$

Algoritmo para hallar el conjunto $Unit(A)$

Transformaciones en GLC

$Unit(A) := \{A\}$

repetir

$$Unit(A) := Unit(A) \cup \{B \in V \mid \exists \ \ C \to B, \ \ C \in Unit(A) \ \}$$

Algoritmo para hallar el conjunto $Unit(A)$

Transformaciones en GLC

$Unit(A) := \{A\}$

repetir

$$Unit(A) := Unit(A) \cup \{B \in V \mid \exists \ \ C \to B, \ \ C \in Unit(A)\ \}$$

hasta que no se añaden nuevas variables a $Unit(A)$.

Eliminación de producciones unitarias

Transformaciones en GLC

- Para cada $B \in Unit(A)$ y cada producción $A \to w$ agregar la producción

$$B \to w$$

Eliminación de producciones unitarias

Transformaciones en GLC

- Para cada $B \in Unit(A)$ y cada producción $A \to w$ agregar la producción

$$B \to w$$

- Eliminar todas las producciones unitarias

Ejemplo

Eliminación de producciones unitarias

$$
\begin{aligned}
S &\to AS \mid AA \mid BA \mid \varepsilon \\
A &\to aA \mid a \\
B &\to bB \mid bC \mid C \\
C &\to aA \mid bA \mid B \mid ab
\end{aligned}
$$

Ejemplo

Eliminación de producciones unitarias

$$\begin{aligned} S &\to AS \mid AA \mid BA \mid \varepsilon \\ A &\to aA \mid a \\ B &\to bB \mid bC \mid C \\ C &\to aA \mid bA \mid B \mid ab \end{aligned}$$

Los conjuntos unitarios para cada variable son:

$$Unit(S) = \{S\} \quad Unit(A) = \{A\}$$

$$Unit(B) = \{B, C\} \quad Unit(C) = \{C, B\}$$

Ejemplo

Eliminación de producciones unitarias

La gramática obtenida al eliminar producciones unitarias es:

$$
\begin{aligned}
S &\to AS \mid AA \mid BA \mid \varepsilon \\
A &\to aA \mid a \\
B &\to bB \mid bC \mid aA \mid bA \mid ab \\
C &\to aA \mid bA \mid ab \mid bB \mid bC
\end{aligned}
$$

Problema de la pertenencia

Normalización de gramáticas

- Dada una gramática G sin ε-producciones ni producciones unitarias y una palabra u de longitud n, ¿Se cumple $u \in L(G)$? Es decir, pertenece u al lenguaje generado por G.

Problema de la pertenencia

Normalización de gramáticas

- Dada una gramática G sin ε-producciones ni producciones unitarias y una palabra u de longitud n, ¿Se cumple $u \in L(G)$? Es decir, pertenece u al lenguaje generado por G.
- Si la palabra u es generada por G la construcción de un árbol de derivación terminará eventualmente.

Problema de la pertenencia

Normalización de gramáticas

- Dada una gramática G sin ε-producciones ni producciones unitarias y una palabra u de longitud n, ¿Se cumple $u \in L(G)$? Es decir, pertenece u al lenguaje generado por G.
- Si la palabra u es generada por G la construcción de un árbol de derivación terminará eventualmente.
- En caso contrario basta con construir el árbol hasta el nivel $2n - 1$ para concluir que $u \notin L(G)$.

Problema de la pertenencia

Normalización de gramáticas

- Dada una gramática G sin ε-producciones ni producciones unitarias y una palabra u de longitud n, ¿Se cumple $u \in L(G)$? Es decir, pertenece u al lenguaje generado por G.
- Si la palabra u es generada por G la construcción de un árbol de derivación terminará eventualmente.
- En caso contrario basta con construir el árbol hasta el nivel $2n - 1$ para concluir que $u \notin L(G)$.
- En cada paso se obtiene un nuevo terminal (a lo mas n pasos) o se aumenta la longitud de la palabra en 1 (a lo mas $n - 1$ pasos)

Forma Normal de Chomsky

Definición

Una GLC G está en forma normal de Chomsky (FNC) si:

- G no contiene variables inútiles.

Forma Normal de Chomsky

Definición

Una GLC G está en forma normal de Chomsky (FNC) si:

- G no contiene variables inútiles.
- G no contiene producciones unitarias ni ε-producciones (salvo $S \to \varepsilon$)

Forma Normal de Chomsky

Definición

Una GLC G está en forma normal de Chomsky (FNC) si:

- G no contiene variables inútiles.
- G no contiene producciones unitarias ni ε-producciones (salvo $S \to \varepsilon$)
- Todas las producciones son de la forma:

$$A \to BC \text{ ó } A \to a$$

donde $B, C \in V$, $a \in T$

Transformación a forma normal de Chomsky

Transformaciones en GLC

Cualquier GLC es equivalente a una gramática en FNC, lo cual se logra como sigue:

- Eliminar las variables inútiles.

Transformación a forma normal de Chomsky

Transformaciones en GLC

Cualquier GLC es equivalente a una gramática en FNC, lo cual se logra como sigue:

- Eliminar las variables inútiles.
- Eliminar las ε-producciones (salvo $S \to \varepsilon$)

Transformación a forma normal de Chomsky

Transformaciones en GLC

Cualquier GLC es equivalente a una gramática en FNC, lo cual se logra como sigue:

- Eliminar las variables inútiles.
- Eliminar las ε-producciones (salvo $S \to \varepsilon$)
- Eliminar producciones unitarias.

Transformación a forma normal de Chomsky

Transformaciones en GLC

Cualquier GLC es equivalente a una gramática en FNC, lo cual se logra como sigue:

- Eliminar las variables inútiles.
- Eliminar las ε-producciones (salvo $S \to \varepsilon$)
- Eliminar producciones unitarias.
- Las producciones restantes son todas de la forma $A \to a$ con $a \in T$ ó $A \to w$ con $|w| \geq 2$

Eliminación de producciones $A \to w, \quad |w| \geq 2$

Forma Normal de Chomsky

- Hacer lo siguiente para cada producción P de la forma $A \to \alpha_1 \alpha_2 \ldots \alpha_n$ con $\alpha_i \in V \cup T, n \geq 2$ (si $n = 2$, al menos uno de α_1, α_2 debe ser terminal, pues si no la producción ya es válida para FNC)

Eliminación de producciones $A \to w, \quad |w| \geq 2$

Forma Normal de Chomsky

- Hacer lo siguiente para cada producción P de la forma $A \to \alpha_1 \alpha_2 \ldots \alpha_n$ con $\alpha_i \in V \cup T, n \geq 2$ (si $n = 2$, al menos uno de α_1, α_2 debe ser terminal, pues si no la producción ya es válida para FNC)
- Si $\alpha_i \in T$, digamos $\alpha_i = a$, entonces

Eliminación de producciones $A \to w, \quad |w| \geq 2$

Forma Normal de Chomsky

- Hacer lo siguiente para cada producción P de la forma $A \to \alpha_1 \alpha_2 \ldots \alpha_n$ con $\alpha_i \in V \cup T, n \geq 2$ (si $n = 2$, al menos uno de α_1, α_2 debe ser terminal, pues si no la producción ya es válida para FNC)
- Si $\alpha_i \in T$, digamos $\alpha_i = a$, entonces
  - Agregar la producción $T_a \to a$, donde T_a es una nueva variable.

Eliminación de producciones $A \to w, \quad |w| \geq 2$

Forma Normal de Chomsky

- Hacer lo siguiente para cada producción P de la forma $A \to \alpha_1 \alpha_2 \ldots \alpha_n$ con $\alpha_i \in V \cup T, n \geq 2$ (si $n = 2$, al menos uno de α_1, α_2 debe ser terminal, pues si no la producción ya es válida para FNC)
- Si $\alpha_i \in T$, digamos $\alpha_i = a$, entonces
  - Agregar la producción $T_a \to a$, donde T_a es una nueva variable.
  - Cambiar α_i por T_a en la producción P

Eliminación de producciones $A \to w, \quad |w| \geq 2$

Forma Normal de Chomsky

- Hacer lo siguiente para cada producción P de la forma $A \to \alpha_1 \alpha_2 \ldots \alpha_n$ con $\alpha_i \in V \cup T, n \geq 2$ (si $n = 2$, al menos uno de α_1, α_2 debe ser terminal, pues si no la producción ya es válida para FNC)
- Si $\alpha_i \in T$, digamos $\alpha_i = a$, entonces
  - Agregar la producción $T_a \to a$, donde T_a es una nueva variable.
  - Cambiar α_i por T_a en la producción P
- Para cada producción P de la forma $A \to B_1 B_2 \ldots B_m$ con $B_i \in V, \ m \geq 3$

Eliminación de producciones $A \to w, \quad |w| \geq 2$

Forma Normal de Chomsky

- Hacer lo siguiente para cada producción P de la forma $A \to \alpha_1 \alpha_2 \ldots \alpha_n$ con $\alpha_i \in V \cup T, n \geq 2$ (si $n = 2$, al menos uno de α_1, α_2 debe ser terminal, pues si no la producción ya es válida para FNC)
- Si $\alpha_i \in T$, digamos $\alpha_i = a$, entonces
  - Agregar la producción $T_a \to a$, donde T_a es una nueva variable.
  - Cambiar α_i por T_a en la producción P
- Para cada producción P de la forma $A \to B_1 B_2 \ldots B_m$ con $B_i \in V, \ m \geq 3$
  - Agregar $(m-2)$ nuevas variables $D_1, \ldots, D_{m-2}$ y reemplazar a P con las siguientes producciones:

$$A \to B_1 D_1, \ D_1 \to B_2 D_2, \ldots, D_{m-2} \to B_{m-1} B_m$$

Ejemplo

Simulación de producciones $A \to w_1 w_2 \dots w_n,\ n \geq 2$

La producción $A \to abBaC$ se simula con producciones simples y binarias como sigue:

- Agregamos las nuevas variables T_a, T_b y las producciones

$$A \to T_a T_b B T_a C, \quad T_a \to a, \quad T_b \to b$$

Ejemplo

Simulación de producciones $A \to w_1 w_2 \dots w_n,\ n \geq 2$

La producción $A \to abBaC$ se simula con producciones simples y binarias como sigue:

- Agregamos las nuevas variables T_a, T_b y las producciones

$$A \to T_a T_b B T_a C, \quad T_a \to a, \quad T_b \to b$$

- Para simular la producción $A \to T_a T_b B T_a C$ agregamos nuevas variables D_1, D_2, D_3 y las producciones binarias necesarias obteniendo finalmente:

$$A \to T_a D_1, \quad D_1 \to T_b D_2, \quad D_2 \to B D_3, \quad D_3 \to T_a C$$

$$T_a \to a, \quad T_b \to b$$

Forma normal de Chomsky

Ejemplo

Transformar la siguiente gramática a FNC:

$$S \to AB \mid aBC \mid SBS$$

$$A \to aA \mid C$$

$$B \to bbB \mid b$$

$$C \to cC \mid \varepsilon$$

Forma normal de Chomsky

Ejemplo

La gramática equivalente en FNC es:

$$
\begin{aligned}
S &\to AB \mid T_aD_1 \mid SD_2 \mid T_aB \mid T_bD_3 \mid b \\
A &\to T_aA \mid T_cC \mid a \mid c \\
B &\to T_bD_3 \mid b \\
C &\to T_cC \mid c
\end{aligned}
$$

$$
\begin{aligned}
D_1 &\to BC \\
D_2 &\to BS \\
D_3 &\to T_bB
\end{aligned}
$$

$$
\begin{aligned}
T_a &\to a \\
T_b &\to b \\
T_c &\to c
\end{aligned}
$$

Forma Normal de Greibach (FNG)

Definición

Una GLC G está en forma normal de Greibach (FNG) si:

- La variable inicial S no es recursiva, es decir, no figura en el lado derecho de las producciones.

Forma Normal de Greibach (FNG)

Definición

Una GLC G está en forma normal de Greibach (FNG) si:

- La variable inicial S no es recursiva, es decir, no figura en el lado derecho de las producciones.
- G no tiene variables inútiles ni ε-producciones.

Forma Normal de Greibach (FNG)

Definición

Una GLC G está en forma normal de Greibach (FNG) si:

- La variable inicial S no es recursiva, es decir, no figura en el lado derecho de las producciones.
- G no tiene variables inútiles ni ε-producciones.
- Todas las producciones son de la forma $A \to a\alpha$, con $\alpha \in V^\star$.

Transformación a forma normal de Greibach

Transformaciones en GLC

- Cualquier GLC es equivalente a una gramática en FNG.

Transformación a forma normal de Greibach

Transformaciones en GLC

- Cualquier GLC es equivalente a una gramática en FNG.
- El proceso requiere que la gramática esté en Forma Normal de Chomsky.

Transformación a forma normal de Greibach

Transformaciones en GLC

- Cualquier GLC es equivalente a una gramática en FNG.
- El proceso requiere que la gramática esté en Forma Normal de Chomsky.
- Con una gramática en forma normal de Greibach se simplifica el problema de la pertenencia.

Problema de la pertenencia

Normalización de gramáticas

- Dada una gramática G en forma normal de Greibach y una palabra u de longitud n, ¿Se cumple $u \in L(G)$? Es decir, pertenece u al lenguaje generado por G.

Problema de la pertenencia

Normalización de gramáticas

- Dada una gramática G en forma normal de Greibach y una palabra u de longitud n, ¿Se cumple $u \in L(G)$? Es decir, pertenece u al lenguaje generado por G.
- Si u es generada por G la construcción de un árbol de derivación terminará eventualmente.

Problema de la pertenencia

Normalización de gramáticas

- Dada una gramática G en forma normal de Greibach y una palabra u de longitud n, ¿Se cumple $u \in L(G)$? Es decir, pertenece u al lenguaje generado por G.
- Si u es generada por G la construcción de un árbol de derivación terminará eventualmente.
- En caso contrario basta con construir el árbol hasta el nivel n para concluir que $u \notin L(G)$.

Problema de la pertenencia

Normalización de gramáticas

- Dada una gramática G en forma normal de Greibach y una palabra u de longitud n, ¿Se cumple $u \in L(G)$? Es decir, pertenece u al lenguaje generado por G.
- Si u es generada por G la construcción de un árbol de derivación terminará eventualmente.
- En caso contrario basta con construir el árbol hasta el nivel n para concluir que $u \notin L(G)$.
- En cada paso se obtiene un nuevo terminal (a lo mas n pasos) y por lo general hay una menor ramificación.