UFRN/DIMAp/DIM0330 — Linguagens formais

http://www.consiste.dimap.ufrn.br/~david/ENSEIGNEMENT/2003.1/DIM0330

Gramáticas livres de contexto

David Déharbe

UFRN/DIMAp
Campus Universitário, Lagoa Nova, 59072-970 Natal, RN, Brasil
email: david@dimap.ufrn.br

Roteiro

- Definições sobre gramáticas.
- A(s) classe(s) das gramáticas regulares:
  - definições,
  - construção de autômato a partir de uma gramática regular.
- A classe das gramáticas livres de contexto:
  - introdução,
  - definições,
  - ambigüidade,
  - simplificação de gramáticas,
  - formas normais de Chomski e de Greibach.

Gramáticas livres de contexto

- As linguagens livres de contexto tem uma grande importância para definir a sintaxe de linguagens de programação;
- as gramáticas livres de contexto são as gramáticas associadas às linguagens livres de contexto;
- estruturas de bloco (com `begin` e `end` associados) ou parênteses aninhadas bem balanceadas não podem ser escritas com linguagens regulares, e sim com linguagens livres de contexto;
- os autômatos a pilha são as máquinas que permitem reconhecer as linguagens livres de contexto.

Histórico

- GLC foram inicialmente desenvolvidas por linguistas para representar linguagens naturais:

$$
\begin{array}{rcl}
\langle\text{frase}\rangle & \longrightarrow & \langle\text{grupo nominal}\rangle\ \langle\text{grupo verbal}\rangle \\
\langle\text{grupo nominal}\rangle & \longrightarrow & \langle\text{artigo}\rangle\ \langle\text{adjetivo}\rangle\ \langle\text{nome}\rangle \\
\langle\text{artigo}\rangle & \longrightarrow & \text{``o''} \\
\langle\text{adjetivo}\rangle & \longrightarrow & \text{``pequeno''} \\
\langle\text{nome}\rangle & \longrightarrow & \text{``príncipe''}
\end{array}
$$

- Descobriu-se que GLC não são adequadas para linguagens como o inglês.
- Na mesma época, cientistas da computação desenvolviam a notação de Backus e Naur (BNF), para descrever linguagens de programação, equivalente às GLC, com algumas abreviações.
- Essa notação simplifica bastante a definição de linguagens de programação e a construção de compiladores.

Roteiro

- Definições sobre gramáticas.
- A(s) classe(s) das gramáticas regulares:
  - definições,
  - construção de autômato a partir de uma gramática regular.
- A classe das gramáticas livres de contexto:
  - introdução,
  - definições,
  - ambigüidade,
  - simplificação de gramáticas,
  - formas normais de Chomski e de Greibach.

Definição: gramática livre de contexto

Definição 1 (Gramática livre de contexto) *Uma gramática* $G = (V, T, P, S)$ *é livre de contexto se toda produção tem a forma* $A \longrightarrow \alpha$*, onde* $A \in V$ *e* α *é uma palavra sobre* $V \cup T$*.*

Usamos a abreviação GLC para gramática livre de contexto. ▮

Definição 2 (Linguagem livre de contexto) *Uma linguagem* L *é livre de contexto se e somente se exite uma GLC* G *tal que* $\mathcal{L}(G) = L$*.*

Usamos a abreviação LLC para linguagem livre de contexto.

Exemplos

- $G_1 = (\{E\}, \{+, *, (,), \mathbf{id}\}, P, E)$ onde P é composto por

$$E \longrightarrow E + E \mid E * E \mid (E) \mid \mathbf{id}$$

- $G_2 = (\{S\}, \{a, b\}, P, S)$ onde P é composto por

$$S \longrightarrow aSb \mid ab$$

- $G_3 = (\{S, A, B\}, \{a, b\}, P, S)$ onde P é composto por

$$\begin{aligned} S &\longrightarrow aB \mid bA \\ A &\longrightarrow a \mid aS \mid bAA \\ B &\longrightarrow b \mid bS \mid aBB \end{aligned}$$

Pergunta: Quais são $\mathcal{L}(G_2)$ e $\mathcal{L}(G_3)$?

Exercício

Dar gramaticas livres de contexto para as seguintes linguagens

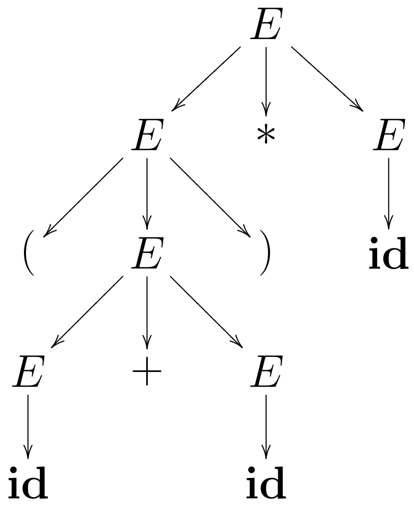
- palíndromas sobre o alfabeto $\{a, b\}$,
- o conjunto das palavras formadas por parênteses balanceadas, por exemplo $(()), ()(), (()()), \ldots$
- as palavras sobre $\{a, b, \emptyset, \varepsilon, ., (,), *\}$ das expressões regulares sobre $\{a, b\}$.
- $\{a^i b^j c^k \mid i \neq j \vee j \neq k\}$.

Árvore de derivação

- Uma árvore de derivação é uma representação gráfica estruturada de uma derivação.
- Os vértices da árvore de derivação são rotulados com símbolos terminais ou não terminais ou com ε.
- Os vértices interiores da árvore são rotulados por símbolos não terminais.
- Se um vértice interior é rotulado por um não terminal X, seus filhos são rotulados por símbolos $X_1, X_2, \ldots X_n$, então necessariamente $X \longrightarrow X_1X_2 \ldots X_n$ é uma produção.

Exemplo de árvore de derivação

Gramática $E \longrightarrow E{+}E \mid E{*}E \mid (E) \mid \mathbf{id}$, palavra $(\mathbf{id} + \mathbf{id}) * \mathbf{id}$



Exercício. Verificar que cada nó interno corresponde a uma produção e cada folha a um símbolo terminal.

Derivações de uma árvore de derivação

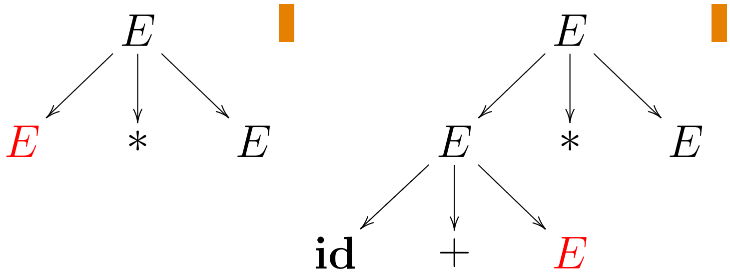
A partir da mesma gramática, uma árvore de derivação pode ser construida de diversas formas.

Por exemplo, a árvore de $(\mathbf{id} + \mathbf{id}) * \mathbf{id}$ pode ser derivada

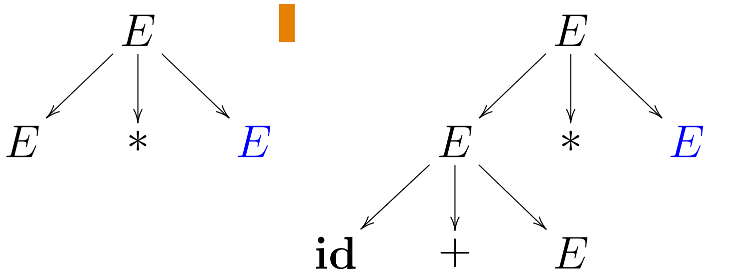
1. $E \Longrightarrow E * E \Longrightarrow (E) * E \Longrightarrow (E + E) * E \Longrightarrow (\mathbf{id} + E) * E \Longrightarrow (\mathbf{id} + \mathbf{id}) * E \Longrightarrow (\mathbf{id} + \mathbf{id}) * \mathbf{id}$
2. $E \Longrightarrow E * E \Longrightarrow E * \mathbf{id} \Longrightarrow (E) * \mathbf{id} \Longrightarrow (E + E) * \mathbf{id} \Longrightarrow (E + \mathbf{id}) * \mathbf{id} \Longrightarrow (\mathbf{id} + \mathbf{id}) * \mathbf{id}$
3. etc.

Posição relativa de não terminais

Não terminal mais a esquerda:



Não terminal mais a direita:



Derivação mais a esquerda, mais a direita

Definição 3 *A* derivação mais a esquerda *de uma árvore de derivação é a seqüência de aplicações de derivação direta substituindo sempre a variável localizada mais a esquerda.*

$$E \Longrightarrow E * E \Longrightarrow (E) * E \Longrightarrow (E + E) * E \Longrightarrow (\mathbf{id} + E) * E \Longrightarrow (\mathbf{id} + \mathbf{id}) * E \Longrightarrow (\mathbf{id} + \mathbf{id}) * \mathbf{id}$$

Definição 4 *A* derivação mais a direita *de uma árvore de derivação é a seqüência de aplicações de derivação direta substituindo sempre a variável localizada mais a direita.*

$$E \Longrightarrow E * E \Longrightarrow E * \mathbf{id} \Longrightarrow (E) * \mathbf{id} \Longrightarrow (E + E) * \mathbf{id} \Longrightarrow (E + \mathbf{id}) * \mathbf{id} \Longrightarrow (\mathbf{id} + \mathbf{id}) * \mathbf{id}$$

Exercício

Seja G a seguinte gramática:

$$
\begin{array}{rcl}
S & \longrightarrow & aB \mid bA \\
A & \longrightarrow & a \mid aS \mid bAA \\
B & \longrightarrow & b \mid bS \mid aBB
\end{array}
$$

Para a palavra $aaabbabbba$, achar a derivação mais a esquerda, a derivação mais a direita, a árvore de derivação.

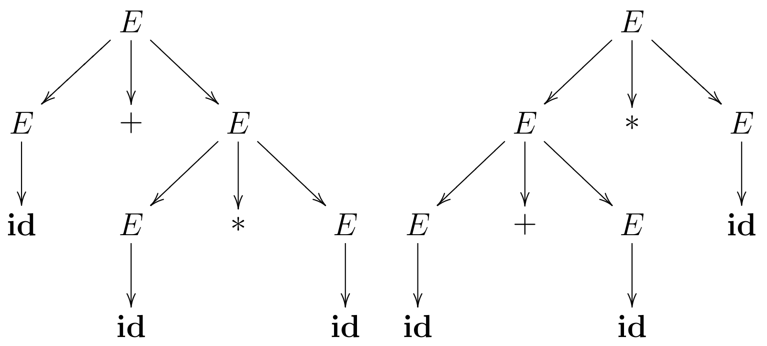
Roteiro

- Definições sobre gramáticas.
- A(s) classe(s) das gramáticas regulares:
  - definições,
  - construção de autômato a partir de uma gramática regular.
- A classe das gramáticas livres de contexto:
  - introdução,
  - definições,
  - ambigüidade,
  - simplificação de gramáticas,
  - formas normais de Chomski e de Greibach.

Existência de múltiplas árvores de derivação

Com a mesma gramática, uma palavra pode ter árvores de derivação distintas.

A palavra $\mathbf{id} + \mathbf{id} * \mathbf{id}$ tem duas árvores de derivação.



Ambigüidade

Definição 5 *Uma gramática livre de contexto G é ambígüa se existe uma palavra que tem árvores de derivação distintas.*

- Equivalentemente, G é ambígüa se existe uma palavra que tem várias derivações mais a esquerda (direita).
- $E \longrightarrow E + E \mid E * E \mid (E) \mid \mathbf{id}$ é ambígüa.
- Existem linguagens livres de contexto que são inherentemente ambígüas (qualquer GLC a gerando é ambígüa).

Exercício

Seja G a seguinte gramática:

$$
\begin{aligned}
S &\longrightarrow aB \mid bA \\
A &\longrightarrow a \mid aS \mid bAA \\
B &\longrightarrow b \mid bS \mid aBB
\end{aligned}
$$

G é ambígüa?

Roteiro

- Definições sobre gramáticas.
- A(s) classe(s) das gramáticas regulares:
  - definições,
  - construção de autômato a partir de uma gramática regular.
- A classe das gramáticas livres de contexto:
  - introdução,
  - definições,
  - ambigüidade,
  - simplificação de gramáticas,
  - formas normais de Chomski e de Greibach.

O que e para que simplificar ?

- Simplificar um gramática G consiste em calcular uma gramática equivalente que tem menor número de produções e/ou produções mais simples.

- Objetivo da simplificação:

  – evitar regras recursivas a esquerda;

  – todos os símbolos são usados para derivar alguma palavra;

  – não haja produções da forma $A \longrightarrow B$

  – se a linguagem não contem ε, não há necessidade de produções da forma $A \longrightarrow \varepsilon$;

- Quanto mais simples a gramáticas, mais simples fica o autômato correspondente.

Teorema de substituição

Teorema 1 *Seja $G = (V, T, P, S)$ uma GLC. Se P contem produções da forma*

$$A \longrightarrow \alpha B \beta \tag{1}$$

tal que A e B são não terminais distintos. Supondo que o conjunto das produções a partir de B seja:

$$B \longrightarrow \gamma_1 \mid \gamma_2 \mid \ldots \mid \gamma_n.$$

Seja $G' = (V, T, P', S)$ a gramática construida tirando de P a produção 1 e adicionando as produções

$$A \longrightarrow \alpha\gamma_1\beta \mid \alpha\gamma_2\beta \mid \ldots \mid \alpha\gamma_n\beta.$$

Então G e G' são gramáticas equivalentes.

Exemplo

Seja a gramática: $\begin{array}{lcl} A & \longrightarrow & a \mid aaA \mid abBc, \\ B & \longrightarrow & abbA \mid b. \end{array}$ Uma derivação da palavra $aaabbc$ é:

$$A \Longrightarrow aaA \Longrightarrow aaabBc \Longrightarrow aaabbc.$$

O resultado da substituição de B é a gramática:

$$\begin{array}{lcl} A & \longrightarrow & a \mid aaA \mid ababbAc \mid abbc, \\ B & \longrightarrow & abbA \mid b. \end{array}$$

A derivação da palavra $aaabbc$ na nova gramática é:

$$A \Longrightarrow aaA \Longrightarrow aaabbc$$

Eliminação de regras recursivas a esquerda

Definição 6 *Uma regra* $A \longrightarrow A\alpha$ *é* recursiva a esquerda*.*

Teorema 2 *Seja* $G = (V, T, P, S)$ *uma GLC, e* $A \in V$*.* P^r_A *é o conjunta das produções recursivas a esquerda cujo símbolo a esquerda é* A*:* $A \longrightarrow A\alpha_1 \mid \ldots \mid A\alpha_n$*.*$P^A$ *é o conjunta das produções não recursivas a esquerda cujo símbolo a esquerda é* A*:* $A \longrightarrow \beta_1 \mid \ldots \mid \beta_m$*.*

A gramatica $G' = (V \cup \{Z\}, T, P', S)$ *tal que* $Z \notin V$*, e* P' *é obtido de* P*, substituindo as regras de* P^r_A *e* P_A *pelas seguintes regras*

$$
\begin{aligned}
A &\longrightarrow \beta_i \mid \beta_i Z \text{ para todo } i = 1, \ldots m, \text{ e} \\
Z &\longrightarrow \alpha_i \mid \alpha_i Z \text{ para todo } i = 1, \ldots n.
\end{aligned}
$$

Então G *e* G' *são gramáticas equivalentes.*

Prova do teorema

Considere um trecho de derivação da gramática G envolvendo A. Esse trecho é prefixado por uma sucessão de produções de P_A^r e termina com uma produção de P_A:

$$A \Longrightarrow A\alpha_i \Longrightarrow A\alpha_j\alpha_i \Longrightarrow^* A\alpha_k \ldots \alpha_j\alpha_i \Longrightarrow \beta_h\alpha_k \ldots \alpha_j\alpha_i.$$

Existe um trecho equivalente em G':

$$A \Longrightarrow \beta_h Z \Longrightarrow \beta_h\alpha_k Z \Longrightarrow^* \beta_h\alpha_i \ldots \alpha_j Z \Longrightarrow \beta_h\alpha_k \ldots \alpha_j\alpha_i.$$

Qualquer palavra de $\mathcal{L}(G)$ pode ser derivada também por G'.

De forma parecida podemos provar que qualquer palavra gerada por G' também pode ser gerada por G.

Exemplo

Seja a gramática $\begin{array}{rcl} A & \longrightarrow & Aa \mid aBc \mid \varepsilon, \\ B & \longrightarrow & Bb \mid ba. \end{array}$

Aplicamos o teorema 2 para A; $\alpha_1 = a$, $\beta_1 = aBc$, e $\beta_2 = \varepsilon$:

$$\begin{array}{rcl} A & \longrightarrow & aBc \mid aBcZ \mid \varepsilon \mid Z, \\ Z & \longrightarrow & a \mid aZ, \\ B & \longrightarrow & Bb \mid ba. \end{array}$$

Aplicamos o teorema 2 para B; $\alpha_1 = b$, $\beta_1 = ba$:

$$\begin{array}{rcl} A & \longrightarrow & aBc \mid aBcZ \mid \varepsilon \mid Z, \\ Z & \longrightarrow & a \mid aZ, \\ B & \longrightarrow & baY \mid ba, \\ Y & \longrightarrow & b \mid bY. \end{array}$$

Exercícios

- Construir GLC equivalentes sem produções recursivas a esquerda:

$$
\begin{array}{llcl}
a) & S & \longrightarrow & abS \mid SS \mid \varepsilon. \\
b) & S & \longrightarrow & Sb \mid A, \\
 & A & \longrightarrow & Aa \mid a. \\
c) & S & \longrightarrow & aZ \mid SbS, \\
 & Z & \longrightarrow & aZb \mid \varepsilon.
\end{array}
$$

- Construir uma GLC sem produções recursivas a esquerda que produz a linguagem das palavras sobre $\{a, b\}$ tais que o número de a é superior ao número de b.

Eliminação de símbolos inúteis

Definição 7 *Seja $G = (V, T, P, S)$ uma GLC. Um símbolo $X \in V \cup T$* é útil *se existir alguma derivação $S \Longrightarrow^* \alpha X \beta \Longrightarrow^* w$ envolvendo X. Caso contrário X* é inútil.

Lema 1 *Seja $G = (V, T, P, S)$ uma GLC, tal que $\mathcal{L}(G) \neq \emptyset$. É possível encontrar $G' = (V', T', P', S)$ equivalente tal que para cada $A \in V$, $A \Longrightarrow^* w$, para alguma palavra $w \in T^*$.*

Lema 2 *Seja $G = (V, T, P, S)$ uma GLC. É possível calcular $G' = (V', T', P', S)$ equivalente tal que para cada $X \in V' \cup T'$, existe uma derivação $S \Longrightarrow^* \alpha X \beta$ (onde $\alpha, \beta \in (V \cup T)^*$).*

Teorema 3 *Qualquer LLC é gerada por uma GLC sem símbolo inútil.*

Prova do lema 1

Seja $G = (V, T, P, S)$ uma GLC, tal que $\mathcal{L}(G) \neq \emptyset$. É possível encontrar $G' = (V', T', P', S)$ tal que para cada $A \in V$, $A \Longrightarrow^* w$, para alguma palavra $w \in T^*$.

Seja $A \in V$ um não terminal.

- Se $A \longrightarrow w$ é uma produção, onde $w \in T^*$, pertence a P, então A pertence a V'.
- Se $A \longrightarrow X_1X_2 \ldots X_n$ é uma produção, onde $X_i \in V' \cup T$, pertence a P, então A pertence a V'.

Primeiro algoritmo de eliminação

oldV $:= \emptyset$

newV $:= \{A \mid \exists w \in T^*, A \longrightarrow w \in P\}$

while oldV $\neq$ newV **do**

oldV $:=$ newV

newV $:=$ oldV $\cup \{A \mid \exists \alpha \in (T \cup oldV)^*, A \longrightarrow \alpha \in P\}$

od

Exemplo: $S \longrightarrow AB \mid a, A \longrightarrow a$.

B não produz nenhum terminal, logo pode ser eliminado.

A regra de produção $S \longrightarrow AB$ também pode ser eliminado.

O resultado é $S \longrightarrow a, A \longrightarrow a$.

Prova do lema 2

Seja $G = (V, T, P, S)$ uma GLC. É possível calcular u$G' = (V', T', P', S)$ equivalente tal que para cada $X \in V' \cup T'$, existe uma derivação $S \Longrightarrow^{*} \alpha X \beta$ (onde $\alpha, \beta \in (V' \cup T')^{*}$).

newV $:= \{S\}$, newT $:= \{\}$, oldV $:= \{\}$, oldT $:= \{\}$
while oldV $\neq$ newV abd oldT $\neq$ newT **do**
  oldV $:=$ newV
  newV $:=$ oldV $\cup$ $\{\alpha_i \mid \exists A \in newV \bullet \alpha_i \in V \wedge A \longrightarrow \alpha_1 \ldots \alpha_n\}$
  newT $:=$ oldT $\cup$ $\{\alpha_i \mid \exists A \in newV \bullet \alpha_i \in T \wedge A \longrightarrow \alpha_1 \ldots \alpha_n\}$
od

Exemplo: $S \longrightarrow A, A \longrightarrow a$. O não terminal A não pode ser alcançado a partir de S. A e $A \longrightarrow a$ são eliminados. O resultado é $S \longrightarrow a$.

Prova do lema 3

Qualquer LLC é gerada por uma GLC sem símbolo inútil.

Seja $G = (V, T, P, S)$ uma GLC gerando a LLC dada.

- Seja $G_1 = (V_1, T_1, P_1, S)$ a GLC resultando da primeira eliminação sobre G;
- Seja $G_2 = (V_2, T_2, P_2, S)$ a GLC resultando da segunda eliminação sobre G_1;
- Supondo que $X \in V_2 \cup T_2$ é inútil.
  - Pelo lema 2, existe uma derivação $S \Longrightarrow^*_{G_2} \alpha X \beta$, onde todos os símbolos de α e β pertencem a $V_2 \cup T_2 \subseteq V_1 \cup T_1$.
  - X também é um símbolo de G_1. Pelo lema 1, $S \Longrightarrow^*_{G_1} \alpha X \beta \Longrightarrow^*_{G_1} w$. Logo X é útil.

Exercícios

Calcular uma GLC sem símbolo inútil equivalente a seguinte gramática:

$$
\begin{array}{lllllll}
S & \longrightarrow & AB \mid CA & \quad & B & \longrightarrow & BC \mid AB \\
A & \longrightarrow & a & & C & \longrightarrow & aB|b
\end{array}
$$

Eliminar os símbolos e produções inúteis da seguinte gramática:

$$
\begin{array}{lll}
S & \longrightarrow & a \mid aA \mid B \mid C, \\
A & \longrightarrow & aB \mid \varepsilon, \\
B & \longrightarrow & Aa, \\
C & \longrightarrow & cCD, \\
D & \longrightarrow & dd \mid ddD.
\end{array}
$$

Eliminação de produções ε

Chamamos de produção ε as produções cuja parte esquerda é ε.

Teorema 4 *Seja $G = (V, T, P, S)$ uma GLC, e $L = \mathcal{L}(G)$ a linguagem gerada por G. Então $L - \{\varepsilon\}$ é a LLC gerada por uma GLC sem símbolos inúteis e sem produções ε.*

Parafraseando, se uma LLC não contem ε, existe uma GLC a gerando que não contem símbolos inúteis, nem produções ε.

Iremos ver agora um procedimento efetivo que, dado uma GLC que deriva uma linguagem L, calcule uma GLC que gera $L - \{\varepsilon\}$ e que não possui produções ε.

Procedimento de eliminação de prod. ε

$G = (V, T, P, S)$ é uma GLC com produções ε.

- $A \in V$ é chamado de *cancelável* se $A \longrightarrow \varepsilon$ é uma produção.
- $A \in V$ também é cancelável se existe uma produção $A \longrightarrow X_1 \ldots X_n$, onde todos os X_i são canceláveis.
- Construimos o conjunto de produções P' da forma seguinte. Se $A \longrightarrow X_1 \ldots X_n \in P$, incluir a P' todas as produções $A \longrightarrow \alpha_1 \ldots \alpha_n$, onde:

  1. se X_i não é cancelável: então $\alpha_i = X_i$;
  2. se X_i é cancelável: então α_i é X_i ou ε;
  3. pelo menos um dos α_i não é ε.

Exemplo de eliminação de produções ε

Seja a gramática $S \longrightarrow AB, A \longrightarrow a, B \longrightarrow b \mid \varepsilon$.

$P = \{S \longrightarrow AB, A \longrightarrow a, B \longrightarrow b, B \longrightarrow \varepsilon\}$.

B é cancelável. Aplicando as regras de construção de P', temos:

regra	P	P'
2	$S \longrightarrow AB$	$S \longrightarrow AB, S \longrightarrow A$
1	$A \longrightarrow a$	$A \longrightarrow a$
1	$B \longrightarrow b$	$B \longrightarrow b$
3	$B \longrightarrow \varepsilon$	

Então $P' = \{S \longrightarrow AB, S \longrightarrow A, A \longrightarrow a, B \longrightarrow b\}$.

Exercícios

Eliminar as produções ε da seguinte gramática:

$$\begin{aligned} S &\longrightarrow AaB \mid aaB, \\ A &\longrightarrow aBa \mid \varepsilon \mid B, \\ B &\longrightarrow bbA \mid \varepsilon. \end{aligned}$$

Eliminação de produções unitárias

Chamamos de produção unitárias aquelas produções $A \longrightarrow B$, onde A, B são não terminais.

Teorema 5 *Cada LLC que não contem ε pode ser definido através de uma GLC sem símbolos inúteis, sem produções ε e sem produções unitárias.*

Construção: Seja $G = (V, T, P, S)$ sem símbolos inúteis e sem produções ε. Construir P' tal que

- inicialmente P' contem todas as produções não unitárias de P.;
- se $A \Rightarrow^*_G B$, onde $A, B \in V$, P' contem todas as produções da forma $A \rightarrow \alpha$ onde $B \longrightarrow \alpha$ é uma produção não unitária de P.

$G' = (V, T, P', S)$ atende as condições do teorema e é equivalente a G.

Exercícios

Eliminar as produções unitárias da gramática:

$$
\begin{aligned}
S &\longrightarrow AaB \mid aaB,\\
A &\longrightarrow aBa \mid \varepsilon \mid B,\\
B &\longrightarrow bbA \mid \varepsilon.
\end{aligned}
$$

Eliminar as produções unitárias da gramática:

$$
\begin{aligned}
S &\longrightarrow a \mid aA \mid B \mid C,\\
A &\longrightarrow aB \mid \varepsilon,\\
B &\longrightarrow Aa,\\
C &\longrightarrow cCD,\\
D &\longrightarrow dd \mid ddD.
\end{aligned}
$$

Exercício recapitulativo de simplificação

Seja a gramática composta pelas seguintes produções:

$$
\begin{array}{rcl}
S & \longrightarrow & XYZ \\
X & \longrightarrow & AXA \mid BXB \mid Z \mid \varepsilon \\
Y & \longrightarrow & AYB \mid BYA \mid Z \mid \varepsilon \\
A & \longrightarrow & a \\
B & \longrightarrow & b \\
Z & \longrightarrow & Zu \mid Zv \mid \varepsilon
\end{array}
$$

1. Qual a linguagem gerada ?
2. Simplificar a gramática usando os diferentes tipos de eliminação expostos.

Exercício recapitulativo de simplificação

Eliminar produções unitárias, produções ε, produções e símbolos inúteis, e produções recursivas a esquerda das seguintes gramáticas:

$$
\begin{array}{llll}
a) & S & \longrightarrow & Sa \mid aAB, \\
 & A & \longrightarrow & B \mid aa \mid \varepsilon, \\
 & B & \longrightarrow & AbB \mid \varepsilon \mid C, \\
 & C & \longrightarrow & aCCa \mid ABC. \\
b) & S & \longrightarrow & aA \mid aB \mid SaA, \\
 & A & \longrightarrow & B \mid aa \mid \varepsilon, \\
 & B & \longrightarrow & bB \mid \varepsilon \mid C, \\
 & C & \longrightarrow & aCa \mid ABC.
\end{array}
$$

Roteiro

- Definições sobre gramáticas.
- A(s) classe(s) das gramáticas regulares:
  - definições,
  - construção de autômato a partir de uma gramática regular.
- A classe das gramáticas livres de contexto:
  - introdução,
  - definições,
  - ambigüidade,
  - simplificação de gramáticas,
  - formas normais de Chomski e de Greibach.

Formal normal de Chomski

Formas normais impõem restrições sobre as produções possíveis.

Definição 8 *Uma GLC é dita na* forma normal de Chomski *quando todas suas produções são da forma* $A \longrightarrow BC$ *ou* $A \longrightarrow a$*, onde* $A, B, C \in V$ e $a \in T$.

A parte direita das regras possuem exatamente dois não terminais ou um terminal.

Teorema 6 *Qualquer LLC* L *não contendo a palavra* ε *pode ser gerada por uma gramática na forma normal de Chomski.*

Prova do teorema

Dada uma GLC G não produzindo ε, construimos uma GLC G' na forma normal de Chomski. Seja $G_1 = (V, T, P, S)$ equivalente a G, sem produções unitárias nem produções ε.

- Para qualquer produção $A \longrightarrow X$ de G_1 com único símbolo X a direita, $X \in T$, e a produção é na forma normal de Chomski.
- Seja uma produção $A \longrightarrow X_1, \ldots X_m$, com $m \geq 2$.
  - Para todo X_i terminal, introduzir um novo não terminal C e a produção $C \rightarrow X_i$. Substituir X_i por C. Seja $A \longrightarrow Y_1, \ldots Y_m$ a produção resultante.
  - Substituir $A \longrightarrow Y_1 \ldots Y_m$ pelas produções $A \longrightarrow Y_1 A_1$, $A_1 \longrightarrow A_2 Y_2$, ... $A_{m-2} \longrightarrow Y_{m-1} Y_m$.

Exemplo

$$
\begin{array}{rcl}
S & \longrightarrow & bA \mid aB \\
A & \longrightarrow & bAA \mid aA \mid a \\
B & \longrightarrow & aBB \mid bS \mid b
\end{array}
$$

$A \longrightarrow a$ e $B \longrightarrow b$ já tem a forma desejada.
Introduzimos $C_a, C_b, A_2, eB_2, C_a \longrightarrow a$ e $C_b \longrightarrow b$:

$$
\begin{array}{rcl}
S & \longrightarrow & C_bA \mid C_aB \\
A & \longrightarrow & C_bA_2 \mid C_aA \mid a \\
B & \longrightarrow & C_aBB \mid C_bS \mid b \\
A_2 & \longrightarrow & AA, B_2 \longrightarrow BB \\
C_a & \longrightarrow & a, C_b \longrightarrow b
\end{array}
$$

Exercício

Calcular a forma normal de Chomski da gramática do cálculo proposicional:

$$S \longrightarrow \,\tilde{}\,S \mid [S \supset S] \mid \mathbf{p} \mid \mathbf{q}$$

Calcular a forma normal de Chomski da gramática das expressões aritméticas:

$$E \longrightarrow E + E \mid E * E \mid (E) \mid \mathbf{id}$$

Formal normal de Greibach

Definição 9 *Uma GLC* (V, T, P, S) *é dita na* forma normal de Greibach *quando todas suas produções são da forma* $A \longrightarrow a\alpha$, *onde* $a \in T$ *e* $\alpha \in V^*$.

As produções de uma gramática na forma normal de Greibach não podem ser recursivas a esquerda.

Teorema 7 *Qualquer GLC* G, *tal que* $\varepsilon \notin \mathcal{L}(G)$, *possui uma GLC* G' *equivalente na forma normal de Greibach.*

Prova do teorema (1/4)

Dada uma GLC G não produzindo ε, construimos uma GLC G' na forma normal de Greibach.

- Calculamos uma GLC equivalente na forma normal de Chomski;
- Renomeamos os não terminais $A_1, \ldots A_n$, com $S = A_1$ (determinamos uma ordem sobre os não terminais);

Por exemplo,

$$\begin{aligned} A_1 &\longrightarrow A_2A_2 \mid a, \\ A_2 &\longrightarrow A_1A_2 \mid b. \end{aligned}$$

Prova do teorema (2/4)

Usando substituições e eliminação de produções recursivas a esquerda, calculamos uma gramática equivalente tal que todas as produções tem uma das seguintes formas:

$$\begin{aligned} A_i &\longrightarrow A_j x_i \text{ com } j > i \text{ e } x_i \in V^*, \\ Z_i &\longrightarrow A_j x_j \text{ com } j \leq n \text{ e } x_j \in V^*, \\ A_i &\longrightarrow a x_i \text{ com } a \in T \text{ e } x_i \in V^*. \end{aligned}$$

$A_1 \longrightarrow A_2A_2 \mid a$, $A_2 \longrightarrow b$ satisfazem os requisitos.
$A_2 \longrightarrow A_1A_2 \mid b$ não satisfaz os requisitos.
A_1 é substituído por A_2A_2 e a: $A_2 \longrightarrow A_2A_2A_2$, $A_2 \longrightarrow aA_2$.
Eliminação da recursividade a esquerda:

$$\begin{aligned} A_2 &\longrightarrow aA_2 \mid aA_2Z \mid b \mid bZ \\ Z &\longrightarrow A_2A_2 \mid A_2A_2Z \end{aligned}$$

Prova do teorema (3/4)

- As produções de A_n tem a forma $A_n \longrightarrow a\alpha_n$ (forma desejada).
- Essas produções são substituídas à esquerda das produções de A_{n-1} e Z_{n-1}, que tornam a ter a forma $A_{n-1} \longrightarrow a\alpha_{n-1}$.
- A operação é repetida até A_1 e todas a gramática resultante é na formal normal de Greibach.

Prova do teorema (4/4)

Temos a gramática:

$$\begin{aligned} A_1 &\longrightarrow A_2A_2 \mid a, \\ A_2 &\longrightarrow aA_2 \mid aA_2Z \mid b \mid bZ \\ Z &\longrightarrow A_2A_2 \mid A_2A_2Z \end{aligned}$$

Substituimos A_2 pelas partes direitas das produções associadas a A_2 na parte direita das produções de A_1 e Z.

$$\begin{aligned} A_1 &\longrightarrow aA_2A_2 \mid aA_2ZA_2 \mid bA_2 \mid bZA_2 \mid a, \\ A_2 &\longrightarrow aA_2 \mid aA_2Z \mid b \mid bZ \\ Z &\longrightarrow aA_2A_2 \mid aA_2ZA_2 \mid bA_2 \mid bZA_2 \mid \\ &\quad\ aA_2A_2Z \mid aA_2ZA_2Z \mid bA_2A_2Z \mid bZA_2Z \mid \end{aligned}$$

Exercício

- Calcular a forma normal de Greibach da gramática seguinte:

$$S \longrightarrow aaS \mid SS \mid aa.$$

- Calcular a forma normal de Greibach da gramática seguinte:

$$S \longrightarrow aSb \mid ab.$$

Conclusões recapitulativas

- Gramáticas são conjuntos de regras de produção;
- Gramáticas regulares são as gramáticas lineares a esquerda e as gramáticas lineares a direita; As linguagens regulares formam a classe de linguagens associadas com as gramáticas regulares;
- Vimos procedimentos efetivos para construir o autômato finito de uma gramática regular, e vice-versa.
- GLC são as gramáticas tais que as produções tem um não terminal a esquerda. LLC formam a classe de linguagens associadas com as GLC.
- Uma GLC é ambígua se existir mais de uma derivação mais a esquerda para uma mesma palavra.
- Apresentamos procedimentos efetivos para calcular GLC equivalentes e formas normais (Chomski, Greibach).

Fontes

A principal fonte para elaborar essa apresentação é:

- *Introdução à Teoria das Linguagens Formais, dos Autômatos e da Computabilidade*. Benedito Melo Aciòly, Benjamín René Callejas Bedregal, e Aarão Lyra. Edições UnP, 2002.

Também foram usados:

- *Linguagens Formais e Autômatos*. Paulo Blauth Menezes. Instituto de Informática da UFRGS, Série livros didáticos, número 3. Editora Sagra Luzzatto, 1998.
- *Introdução to Automata Theory, Languages and Computation*. John E. Hopcroft e Jeffrey D. Ullman. Addison-Wesley Publishing Company. 1979.

A apresentação foi preparada usando os programas `pdfLaTeX`, `ppower4`, `GNU Emacs`, e `linux`.