LFA - PARTE 2

Linguagens Livres de Contexto e Autômatos a Pilha (*Push-Down*)

Um Modelo de Computação de Força Intermediária

João Luís Garcia Rosa
LFA-FEC-PUC-Campinas 2002

1

Gramática Livre de Contexto

- O lado esquerdo de cada produção consiste de um único não terminal
- Exemplo

$$S \to aSb$$
$$S \to \lambda$$

gera a linguagem $\{a^n b^n | n \geq 0\}$.

2

Outro Exemplo

$$S \to aaZcc$$
$$Z \to aZc$$
$$Z \to b$$

gera a linguagem $\{a^n b c^n | n \geq 2\}$

3

Terminologia

Compare a produção

$$Y \to abc$$

com $$ZYW \to ZabcW$$

- Livre de contexto/sensível ao contexto

4

Linguagens Livres de Contexto

- Gerada por gramáticas livres de contexto
  - $\{a^n b^n | n \geq 0\}$
  - $\{a^n bc^n | n \geq 2\}$

5

Gramáticas Lineares à Direita

- Toda produção da forma $Z \to xY$ onde x é um símbolo terminal de X
- Exemplo

$$S \to aS$$
$$S \to a$$

- Um tipo especial de gramática livre de contexto

6

Linguagens Regulares e Gramáticas Lineares à Direita

- A Linguagem *L* é regular se e apenas se for gerada por uma gramática linear à direita
- Conclusão: Toda linguagem regular é livre de contexto.
- Mas algumas linguagens livres de contexto não são regulares (Lema do Bombeamento para AEFs)

7

Gramáticas Livres de Contexto Positivas

- *G* é uma gramática livre de contexto com produções-λ tal que $L(G)$ não contém λ.
- Então existe uma gramática livre de contexto equivalente *G´* sem produções-λ.
- *G´* é dita *positiva.*

8

Produções Unidade

- Da forma $Z \rightarrow Y$
- Pode-se sempre eliminá-las.

9

Forma Normal de Chomsky (FNC)

- Gramática livre de contexto G
- Toda produção de G ou é da forma $A \rightarrow BC$ ou da forma $A \rightarrow a$, onde A, B são não terminais e a é terminal.
- Aplicações em projetos de compiladores (construção de parser automático).

10

Algoritmo de Conversão para FNC

- **Passo (1)** Elimine as produções-λ e produções unidade
- **Passo (2)** Para as produções remanescentes $a \rightarrow b$ não da forma $A \rightarrow BC$ nem da forma $A \rightarrow a$, substitua as ocorrências dos terminais a, b, c, … em b por novos não terminais C_a, C_b, C_c, … e então adicione as novas produções

$$C_a \rightarrow a \quad C_b \rightarrow b \quad C_c \rightarrow c \ \ldots$$

11

Algoritmo (cont.)

- **Passo (3)** Se o lado direito de qualquer produção contém três ou mais não terminais, então decomponha esta produção em uma série de produções cujos lados direitos consistam de exatamente dois não terminais

12

Disponibilidade

- **Teorema.** Suponha que G seja uma gramática livre de contexto positiva. Então existe uma gramática G' livre de contexto positiva na forma normal de Chomsky tal que $L(G) = L(G')$.

13

Forma Normal de Greibach (FNG)

- Toda produção da forma $Z \rightarrow a\alpha$ onde Z é não terminal, a é terminal, e α é uma cadeia (possivelmente vazia) de não terminais.

14

Algoritmo de Conversão

- **Passo (1)** Ache a gramática equivalente G' na FNC
- **Passo (2)** Ordene os não terminais de G' de V_1 a V_n.
- **Passo (3)** Trabalhe para cima através dos não terminais de G', fazendo as substituições de tal forma a assegurar que todos fiquem ascendentes

15

Algoritmo (cont.)

- **Passo (4)** Trabalhe para baixo através dos não terminais, fazendo as substituições para assegurar que todas as produções estejam na FNG, i.e., na forma $Z \rightarrow a\alpha$, onde Z é não terminal, a é um terminal, e α uma cadeia (possivelmente vazia) de não terminais.

16

Disponibilidade

- **Teorema.** Seja G uma gramática livre de contexto positiva. Então existe G' na FNG que é equivalente a G.

17

Autômato *Push-Down* (a Pilha)

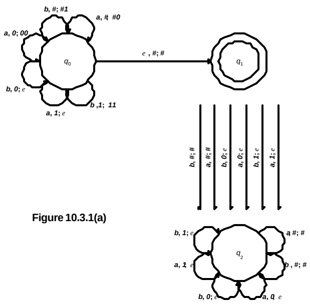

Figure 10.3.1(a)

18

APDs

Função de transição d_M

$$d_M(q_0, a, \#) = (q_0, \#0)$$

- Pilha e símbolo de iniciação de pilha
- Estados de aceitação (apenas aceitação de linguagem)
- Usualmente não determinístico

19

Aceitação de Palavra

- $M = \langle Q, X, \Gamma, d_M, q_0, Z_0, F \rangle$
- *A aceitação da palavra não vazia w* provê que existe alguma computação tal que, começando em q_0 com Z_0 na pilha, percorrendo o símbolo mais a esquerda de w, M pára em algum estado de aceitação depois de ter lido w completamente da esquerda para a direita.
- Caso especial da palavra vazia

20

Aceitação da Linguagem

- *O APD não determinístico M aceita a linguagem L* se *M* aceita todas e apenas as palavras de *L*.
- *L* é *APD-aceitável* se existe um APD não determinístico que aceita *L*.
- Exemplo: $\{a^n b^m \mid 0 \leq n \leq m \text{ e } m \geq 2\}$.
- Exemplo: $\{a^n b^m \mid n \geq m \geq 0\}$.

21

Nova Noção de Aceitação da Palavra

- **Definição.** *O autômato "push-down" M aceita a palavra w pela pilha vazia* desde que pare com a pilha vazia, tendo lido *w* (e não mais) um símbolo por vez da esquerda para a direita.

22

Um Resultado Equivalente

- Existe um APD *M* que aceita *L* pelo estado de aceitação se e somente se existir um APD *M´* que aceita *L* pela pilha vazia.

23

AEFs e APDs

- Toda linguagem AEF-aceitável é APD-aceitável.
- É possível converter um AEF em um APD ignorando a pilha.

24

Teoremas

- Se L é livre de contexto, então existe um APD M que aceita L pela pilha vazia. Em outras palavras, se L é livre de contexto, então L é APD-aceitável.
- Se L é APD-aceitável, então L é livre de contexto.

25

O Lema do Bombeamento para Linguagens Livres de Contexto

- Seja L uma linguagem livre de contexto infinita.
- Então L deve conter a palavra $z = uvwxy$ tal que

  (i) v e x não são ambos λ

  (ii) $uv^i wx^i y$ está em L para todo $i \geq 0$.

26

Aplicação do Lema do Bombeamento

- $L = \{a^n b^n c^n \mid n > 0\}$ não é livre de contexto.
- Então existem linguagens que não são livres de contexto.
- Os APDs têm força intermediária (aceitação de linguagem) entre os AEFs e as Máquinas de Turing.

27