LFA - PARTE 1

Linguagens Regulares e Autômatos de Estados Finitos

Um Modelo Fraco de Computação

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

1

Linguagens Formais

- alfabeto $X = \{a, b\}$
- *palavra sobre X* (exemplos a, ab, ba, $bbaa$)
- *comprimento* $|w|$ da palavra w
- *palavra vazia* ou *palavra nula* λ: $|\lambda| = 0$
- a^3b^2 para $aaabb$
- $n_a(aba) = 2$ e $n_b(aba) = 1$

2

Linguagens Formais (cont.)

- X^* para o conjunto de todas as palavras sobre X
- Linguagens = subconjuntos de X^*
- $L = \{w \in X^*: |w| = 3\} = \{aaa, aab, aba, abb, baa, bab, bba, bbb\}$
- $L' = \{w \in X^*: |w| \text{ é par}\} = \{\lambda, aa, ab, ba, bb, aaaa, aaab, ...\}$

3

Casos Especiais

- Linguagem vazia $\emptyset$
- Linguagem unidade $L = \{aaba\}$

4

Expressões Regulares Denotam Linguagens

- a^* denota a linguagem $\{a^n | n \geq 0\}$
- $(a).(b)$ ou apenas ab denota a linguagem unidade $\{ab\}$
- a^*b^* denota $\{a^n b^m | n, m \geq 0\}$
- $a^2 b^3$ denota $\{aabbb\}$
- **a^n não é expressão regular**
- $a^+ bb$ denota $\{a^n bb | n \geq 1\}$

5

Expressões Regulares (cont.)

- $a|b$ denota $\{a, b\}$
- Definição indutiva
- $\emptyset$ é expressão regular (sobre X) e denota a linguagem $\emptyset$
- λ é expressão regular (sobre X) e denota a linguagem $\{\lambda\}$.

6

Linguagens Formais (revisão)

- X^* para o conjunto de todas as palavras sobre X
- Linguagens = subconjuntos de X^*
- Operações de formação de linguagens
  - $L_1 \cup L_2$
  - $L_1 . L_2$
  - L_1^* (fechamento de Kleene)

7

Linguagens Regulares

- Seja L uma linguagem sobre o alfabeto X, i.e., $L \subseteq X^*$. Então L é *linguagem regular* se L é denotada por alguma expressão regular sobre X
- X é o alfabeto finito e L_1 e L_2 são linguagens regulares sobre X. Então $L_1 \cup L_2$, $L_1 . L_2$, e L_1^* são também regulares

8

Lembre-se

- X é alfabeto finito e w é qualquer palavra sobre X. Então a linguagem unidade $\{w\}$ é regular.
- Qualquer linguagem finita sobre X é regular.

9

Autômatos de Estados Finitos Determinístico

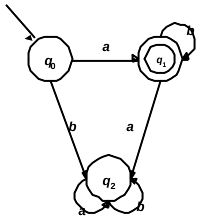


Figure 9.2.1(a)

Outro Exemplo

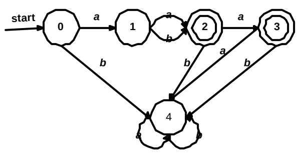


Figure 9.2.1(b)

Determinismo

- Determinismo significa que, dentro de qualquer diagrama de estados para o AEF, o caminho rotulado pela palavra dada w é único: para a palavra $w \in X^*$, há exatamente um caminho começando em q_0 rotulado por w

Funções de Transição

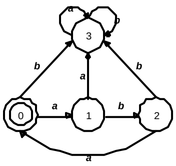


Figure 9.2.1(c)

13

Definição Formal

- AEF é uma quíntupla $\langle X, Q, q_0, F, d_M \rangle$
- X é o *alfabeto de entrada*
- Q é o conjunto finito de *estados*, não vazio
- $q_0 \in Q$ é o *estado inicial*
- $F \subseteq Q$ é um (possivelmente vazio) conjunto de *estados de aceitação* ou *terminais*
- d_M: $Q \times X \rightarrow Q$ é *função de transição* (total)

14

Aceitação da Palavra

- O *autômato de estados finitos determinístico M aceita a palavra* $w \in X^*$ se um único caminho começando em q_0 e rotulado por w leva a algum elemento de F, i.e., a algum estado de aceitação de M.

15

Aceitação da Linguagem

- A *linguagem aceita por M* é o conjunto de todas e apenas as palavras sobre *X* que são aceitas por *M*.
- *L*(*M*) é a linguagem aceita por *M*.
- AEFs são aceitadores de linguagens

16

Uma Máquina Não Determinística

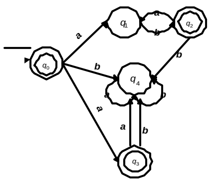


Figure 9.3.3

17

Não Determinismo

- $d_M\colon\ Q \times X \to Q$ é um *mapeamento* de transição
- Assumida como total ("completamente definida") mas permitida ser multivalorada

18

Aceitação da Palavra

- A *palavra* $w \in X^*$ *aceita por M* provê que existe pelo menos um caminho, rotulado por *w*, no diagrama de estados de *M* levando de q_0 para um estado terminal
- Compare com o caso determinístico

19

Aceitação da Linguagem

- A *linguagem aceita por um autômato de estados finitos não determinístico* AND é o conjunto de palavras aceitas por *M*.

20

O Projeto Não Determinístico é Freqüentemente Mais Fácil

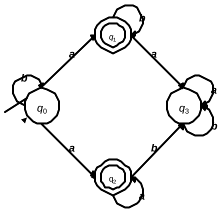


Figure 9.3.4

This nondeterministic finite-state automaton accepts the language denoted by regular expression $b^*((ab^*)|a^+)$

b*((ab*) | a+)

21

Meta

- Suponha um dado AND não determinístico *M* que aceita *L*. Apresenta-se um algoritmo para construir, baseado em *M*, um novo AEF *determinístico* *M´* que aceita *L*

22

Construção de Subconjuntos

- Estados do AND não determinístico *M´* corresponderão a *conjuntos* de estados não vazios do AEF determinístico *M*
- Como q_0 é o estado inicial de *M*, use $\{q_0\}$ como estado inicial de *M´*.
- Estados de aceitação de *M´* serão os conjuntos de estados que contêm pelo menos um estado de aceitação de *M*.

23

Algoritmo

- Para o conjunto inicial $\{q_0\}$, verifique quais transições a partir de q_0 há em *M* e crie um novo estado composto pelo conjunto de estados destinos das transições para cada símbolo terminal.
- Repita para cada novo estado criado, até que não haja mais estados novos.

24

Teorema de Kleene

- Seja M um autômato não determinístico (AND) que aceita L. Então existe um autômato de estados finitos determinístico (AEF) M' que também aceita L.

25

Teorema

- Qualquer linguagem finita é AEF-aceitável
- Exemplo $L = \{abba, abb, abab\}$

26

Lema do Bombeamento para Linguagens AEF-Aceitáveis

- Parece que $L = \{a^n b^n \mid n \geq 0\}$ não é aceita por nenhum AEF (argumento intuitivo)
- Mas isto pode ser demonstrado?
- Conseqüência do Lema do Bombeamento

27

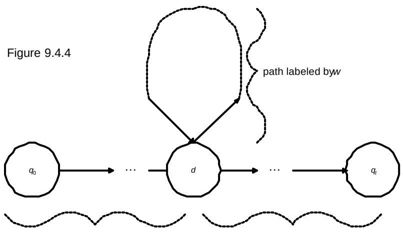
Lema do Bombeamento

- Suponha que L é AEF-aceitável e infinita. Então há palavras u, w e v com $w \neq \lambda$ tal que uw^iv está em L para todo $i \geq 0$
- Escreve-se w^i para o resultado da concatenação da palavra w consigo mesma i vezes (iteração)

28

Prova

Figure 9.4.4



29

Teorema

- $L = \{a^nb^n \mid n > 0\}$ não é aceita por nenhum autômato de estados finitos
- Aplicação do Lema do Bombeamento

30

Prova Indireta

- Suponha que *L* é AEF-aceitável por contradição.
- 3 possibilidades:
  - *w* consiste de apenas *a*s
    - ***aaaaaaaaa…aaa bbbbbbbbb…bbb***
    - O bombeamento de *w* leva à contradição (*a*s demais)

31

Prova (Apenas Outras Possibilidades)

- *w* consiste apenas de *b*s
  - Como o primeiro caso (Desta vez o bombeamento leva a *b*s demais)
- *w* consiste de *a*s seguido de *b*s
  - ***aaaa…aaa…a bbbb…bbb … b***

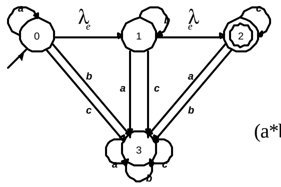
32

Rótulos Vazios

- A execução de arcos rotulados por λ não avançam a entrada
- Arcos-λ podem ou não introduzir não determinismo

33

Exemplo



(a*b*c*)

This FSA accepts the language $L(a^*b^*c^*)$.

Figure 9.7.1

34

Resultado de Equivalência

- Seja M um autômato de estados finitos com movimentos λ. Então existe um autômato de estados finitos M' com nenhum movimento λ tal que $L(M) = L(M')$

35

Gramáticas Gerativas

Exemplo com apenas 2 *produções*

(1) $S \rightarrow aSb$

(2) $S \rightarrow \lambda$

- Gera todas as palavras da forma $a^n b^n$ para $n \geq 0$

36

Definição

- *Produções vazias*
- terminais da gramática (minúsculas)
- alfabeto terminal X
- não terminais da gramática (maiúsculas)
- alfabeto não terminal V
- símbolo inicial S em V
- conjunto de produções P

37

Segundo Exemplo

(1) $S \to aaXcc$
(2) $X \to aXc$
(3) $X \to b$

- Gera todas as palavras da forma $a^n bc^n$ para $n \geq 2$

38

Terceiro Exemplo

(1) $S \to aS'bc$
(2) $S \to \lambda$
(3) $S' \to aS'bC$
(4) $S' \to \lambda$
(5) $Cb \to bC$
(6) $Cc \to cc$

- Gera a linguagem $\{a^n b^n c^n | n \geq 0\}$

39

Linguagens de Estrutura de Frase

- Seja $G = \langle X, V, S, P\rangle$ uma gramática gerativa. Então a linguagem $L(G)$ gerada por G consistirá de todas e apenas as palavras w sobre X tal que $S \Rightarrow^*_G w$
- L é uma *linguagem de estrutura de frase* desde que $L = L(G)$ para alguma gramática gerativa G

40

Equivalência

- Duas gramáticas gerativas G and G' são *equivalentes* se $L(G) = L(G')$.

41