

Processamento Estatístico da Linguagem Natural

Aula 4

Professora Bianca
(Sala 302 – Bloco E)
bianca@ic.uff.br

<http://www.ic.uff.br/~bianca/peln/>

Aula 4 - 09/09/2008

1

Aula de Hoje

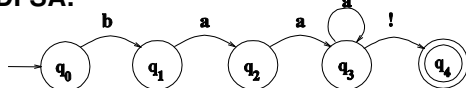
- Terminar Capítulo 2 – Expressões Regulares
- Começar Capítulo 3 – Palavras e Transducers

Aula 4 - 09/09/2008

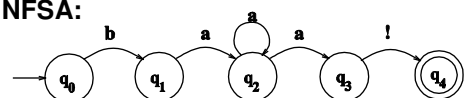
2

Não-Determinismo

DFSA:



NFSA:



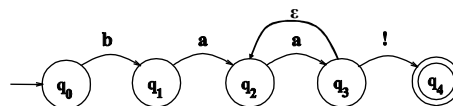
Duas transições com o mesmo símbolo a partir do estado q_2 !

Aula 4 - 09/09/2008

3

Não-determinismo (cont.)

- Outra técnica
 - Transições epsilon
 - Não examinam ou avançam a fita durante o reconhecimento.



Não sabemos se seguimos a transição ϵ ou a transição com "!" ao ler "!" em q_3 !

Aula 4 - 09/09/2008

4

Equivalência

- NFSAs podem ser convertidos para DFSAs através de um algoritmo simples.
 - Isso significa que eles são equivalentes em poder de expressão: as mesmas linguagem que podem ser representadas por um NFSA também podem ser representadas por um DFSA.
 - Isso também significa que podemos fazer reconhecimento em NFSAs usando o algoritmo de DFSAs.
 - Mas a transformação pode gerar um número exponencial de estados.

Aula 4 - 09/09/2008

5

Reconhecimento Não-Determinístico

- Em um NFSA existe pelo menos um caminho do estado inicial ao estado final para uma string que esteja na linguagem definida pelo autômato.
 - Mas nem todos os caminhos correspondentes a string levam a um estado final.
- Nenhum caminho leva uma string que não está na linguagem a um estado final.

Aula 4 - 09/09/2008

6

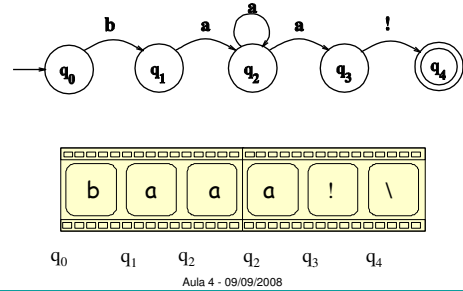
Reconhecimento Não-Determinístico

- Logo o sucesso no reconhecimento não-determinístico ocorre quando um caminho é encontrado que leva ao estado final.
- O fracasso ocorre quando nenhum dos caminhos possíveis para a string leva ao estado final.

Aula 4 - 09/09/2008

7

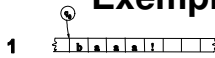
Exemplo



Aula 4 - 09/09/2008

8

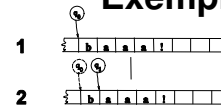
Exemplo



Aula 4 - 09/09/2008

9

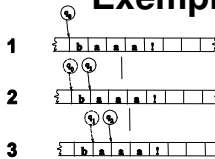
Exemplo



Aula 4 - 09/09/2008

10

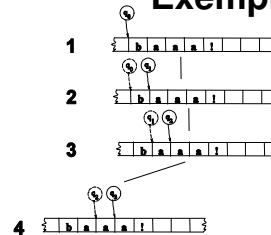
Exemplo



Aula 4 - 09/09/2008

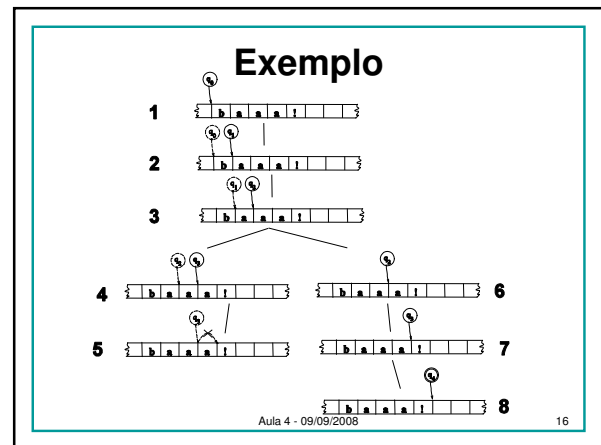
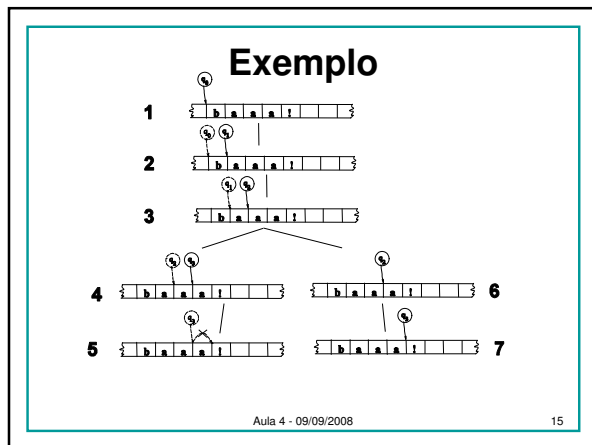
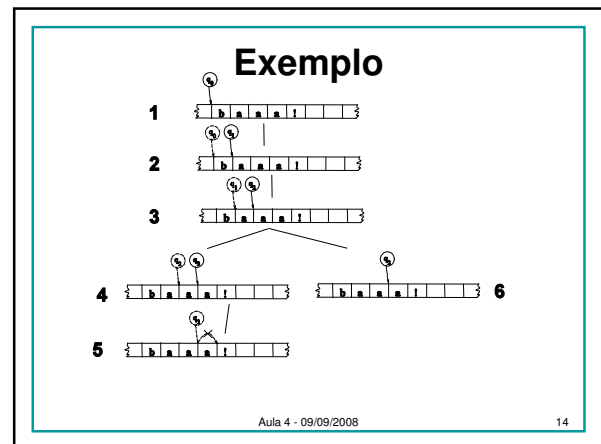
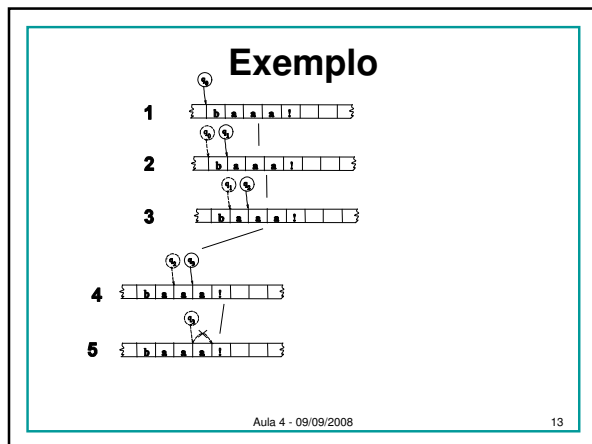
11

Exemplo



Aula 4 - 09/09/2008

12



Reconhecimento = Busca

- Estados do espaço de busca são formados por pares que contêm a posição da fita e o estado do autômato.
- Guardando a informação de que estados ainda não foram explorados, o reconhecedor pode explorar sistematicamente todos os caminhos possíveis para um autômato com uma determinada entrada.

Aula 4 - 09/09/2008 17

ND-Recognize Code

```

function ND-RECOGNIZE(tape, machine) returns accept or reject
  agenda  $\Leftarrow$  [(Initial state of machine, beginning of tape)]
  current-search-state  $\Leftarrow$  NEXT(agenda)
  loop
    if ACCEPT-STATE?(current-search-state) returns true then
      return accept
    else
      agenda  $\Leftarrow$  agenda  $\cup$  GENERATE-NEW-STATES(current-search-state)
    if agenda is empty then
      return reject
    else
      current-search-state  $\Leftarrow$  NEXT(agenda)
  end
  
```

Dependendo de como Next for implementada temos uma busca em extensão ou uma busca em profundidade.

Aula 4 - 09/09/2008 18

Busca infinita

- Buscas deste tipo podem entrar em loop infinito se não formos cuidadosos.
- Como?

Aula 4 - 09/09/2008

19

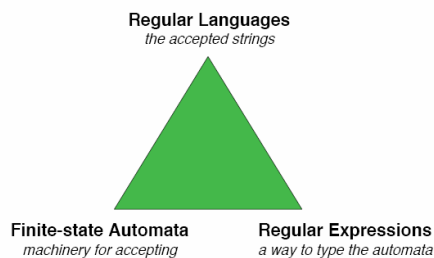
Por que usar não-determinismo?

- Não-determinismo não dá mais poder formalmente e causa “dor-de-cabeça” então porque utilizá-lo?
 - Soluções mais simples e naturais
 - Autômatos baseados no algoritmo de conversão são muito grandes.

Aula 4 - 09/09/2008

20

Três Visões



Aula 4 - 09/09/2008

21

Linguagens Regulares

- É a classe de linguagens que pode ser descrita por expressões regulares.
- Dado um alfabeto Σ , as linguagens regulares sobre Σ são:
 - O conjunto vazio $\emptyset$ é uma linguagem regular.
 - $\forall a \in \Sigma \cup \epsilon$, $\{a\}$ é uma linguagem regular.
 - Se L_1 e L_2 são linguagens regulares, então:
 - $L_1 \cdot L_2 = \{xy | x \in L_1, y \in L_2\}$, **concatenação** de L_1 e L_2
 - $L_1 \cup L_2$, a **união** de L_1 e L_2
 - L_1^* , o **fechamento de Kleene** de L_1

Aula 4 - 09/09/2008

22

De expressões regulares para FSAs

- Todas as linguagens regulares obedecem as propriedades acima e são também as linguagens caracterizáveis por expressões regulares.
- Todos os operadores de expressão regular podem ser implementados através de combinações de união, disjunção, fechamento
 - Contadores (*,+) são repetição mais fechamento
 - Âncoras são símbolos individuais.
 - [], () e . são tipos de disjunção.

Aula 4 - 09/09/2008

23

De expressões regulares para FSAs

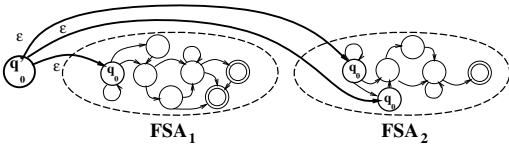
- Se pudéssemos mostrar como transformar fechamento/união/concatenação de expressões regulares para um FSA, teríamos uma idéia de como a conversão para FSA funciona.
- A prova de que linguagens regulares = FSAs tem 2 pares
 - Um FSA pode ser construído pra qualquer linguagem regular.
 - Uma linguagem regular pode ser construída pra cada autômato.
 - Pegue qualquer linguagem regular e construa um autômato.
 - Intuição: indução
 - Caso base: construa um autômato para um símbolo individual (por exemplo 'a')
 - Passo indutivo: Mostre como fazer as 3 operações de expressões regulares com autômatos.

Aula 4 - 09/09/2008

24

União

- Aceita uma string em qualquer das duas linguagens.

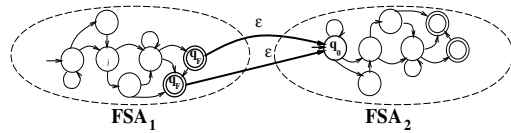


Aula 4 - 09/09/2008

25

Concatenação

- Aceita uma string consistindo de uma string da linguagem L1 seguida de uma string da linguagem L2.



Aula 4 - 09/09/2008

26

Aula de Hoje

- Terminar Capítulo 2 – Expressões Regulares
- **Começar Capítulo 3 – Palavras e Transducers**

Aula 4 - 09/09/2008

27

FSAs e Morfologia Computacional

- Um uso importante dos FSA é para morfologia, o estudo de como as palavras se subdividem em partes menores.

Aula 4 - 09/09/2008

28

Morfologia da Língua Inglesa

- Morfologia é o estudo das maneiras com que palavras são construídas a partir de unidades menores de significado chamadas de morfemas.
- Podemos normalmente dividir morfemas em duas classes
 - **Raízes** (stems): As unidades principais de significado.
 - **Afixos**: aderem às raízes para mudar seu significado e função gramatical.

Aula 4 - 09/09/2008

29

Substantivos e Verbos (em Inglês)

- Substantivos são simples
 - Marcadores para plural e possessivo.
- Verbos são apenas um pouco mais complexos
 - Marcadores apropriados ao tempo do verbo.

Aula 4 - 09/09/2008

30

Regulares e Irregulares

- Morfologia inglesa é um pouco mais complicada porque algumas palavras são irregulares.
 - Mouse/mice, goose/geese, ox/oxen
 - Go/went, fly/flew
- Os termos regular e irregular serão usados para palavras que seguem a regra geral ou não, respectivamente.

Aula 4 - 09/09/2008

31

Substantivos e Verbos Regulares e Irregulares

- Regulares
 - Walk, walks, walking, walked, walked
 - Table, tables
- Irregulares
 - Eat, eats, eating, **ate**, **eaten**
 - Catch, catches, catching, **caught**, **caught**
 - Cut, cuts, cutting, **cut**, **cut**
 - Goose, **geese**

Aula 4 - 09/09/2008

32

Compute

- Muitos caminhos são possíveis...
- Comece com **compute**
 - Computer -> computerize -> computerization
 - Computation -> computational
 - Computer -> computerize -> computerizable
 - Compute -> computee

Aula 4 - 09/09/2008

33

Pra que serve a morfologia?

- 'Stemming' em recuperação de informação
 - Podemos procurar por "aardvark" e encontrar páginas com "aardvark" e "aardvarks"
- Morfologia em tradução automática
 - Precisamos saber que as palavras **quiero** e **quieres** são relacionadas a **querer** 'want'
- Morfologia em correção ortográfica
 - Precisamos saber que **misclam** e **antiundoggingly** are não são palavras apesar de serem feitas de partes de palavras.

Aula 4 - 09/09/2008

34

Não podemos listar todas as palavras

- Turco
- Uygarlastiramadiklarimizdanmissinizcasina
- '(behaving) as if you are among those whom we could not civilize'
- Uygar 'civilized' + las 'become' + tir 'cause' + ama 'not able' + dik 'past' + lar 'plural' + imiz 'p1pl' + dan 'abl' + mis 'past' + siniz '2pl' + casina 'as if'

Aula 4 - 09/09/2008

35

O que queremos

- Algo que faça automaticamente um mapeamento do tipo:
 - Cats **cat** +N +PL
 - Cat **cat** +N +SG
 - Cities **city** +N +PL
 - Merging **merge** +V +Present-participle
 - Caught **catch** +V +past-participle

Aula 4 - 09/09/2008

36

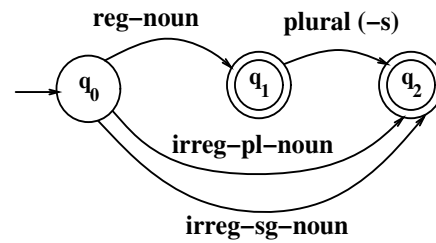
Parsing Morfológico

- Para construir um parser morfológico precisamos de:
 - Léxico:** lista de raízes e afixos, mais informação básica sobre eles (se uma raiz é substantivo ou verbo, por exemplo).
 - Morfotática:** o modelo de ordenação de morfemas que explica que classes de morfemas podem seguir outras classes de morfemas dentro da mesma palavra.
 - Por exemplo, o fato de o morfema de plural vir depois do morfema substantivo em inglês.
 - Regras ortográficas:** usadas para modelar as mudanças que ocorrem em uma palavra quando dois morfemas se combinam.

Aula 4 - 09/09/2008

37

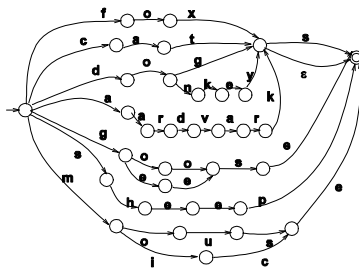
Regras Simples



Aula 4 - 09/09/2008

38

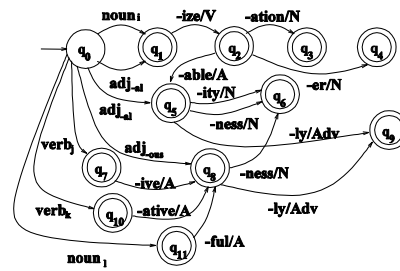
Adicionando palavras



Aula 4 - 09/09/2008

39

Regras Derivacionais



Aula 4 - 09/09/2008

40