

Processamento Estatístico da Linguagem Natural

Aula 3

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

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

Aula 3 - 04/09/2008

1

Expressões Regulares

- Considere as seguintes tarefas:
 - Procurar por todos os preços em um documento.
 - Procurar a palavra “sapato” com um “s” opcional no final.
- Expressões regulares permitem especificar buscas desse tipo através de uma linguagem simples.

Aula 3 - 04/09/2008

2

Expressões Regulares

- Formalmente, uma expressão regular é uma notação que caracteriza um conjunto de **strings**.
 - Uma string é uma seqüência de caracteres alfanuméricos (letras, números, espaços, tabs e pontuação).
- A busca usando expressões regulares requer:
 - O **padrão** que queremos encontrar.
 - Um **corpus** de textos que será examinado.

Aula 3 - 04/09/2008

3

Expressões Regulares Básicas

	Pattern	Matches
Character Concat	went	went
Alternatives	(go went)	go went
	[aeiou]	a o u
disjunc. negation	[^aeiou]	b c d f g
wildcard char	.	a z &
Loops & skips	a*	e a aa aaa ...
one or more	a+	a aa aaa
zero or one	colou?r	color colour

Aula 3 - 04/09/2008

4

Mais Expressões Regulares

- Special characters
 - **\t** tab **\v** vertical tab
 - **\n** newline **\r** carriage return
- Aliases (shorthand)
 - **\d** digits **[0-9]**
 - **\D** non-digits **[^0-9]**
 - **\w** alphabetic **[a-zA-Z]**
 - **\W** non-alphabetic **[^a-zA-Z]**
 - **\s** whitespace **[\t\n\r\f\v]**
 - **\w** alphabetic **[a-zA-Z]**
- Examples
 - **\d+ dollars** 3 dollars, 50 dollars, 982 dollars
 - **\w*oo\w*** food, boo, oodles
- Escape character
 - **** is the general escape character; e.g. **\.** is not a wildcard, but matches a period **.**
 - if you want to use **** in a string it has to be escaped ****

Aula 3 - 04/09/2008

5

Mais Expressões Regulares

- **Anchors**. AKA, “zero width characters”.
- They match positions in the text.
 - **^** beginning of line
 - **\$** end of line
 - **\b** word boundary, i.e. location with **\w** on one side but not on the other.
 - **\B** negated word boundary, i.e. any location that would not match **\b**

Aula 3 - 04/09/2008

6

Exemplo 1

- Expressão regular para encontrar todas as instâncias da palavra “dia” em um texto.
 - **dia**
Não encontra “Dia”
 - **[dD]ia**
Retorna “sadia” or “diafragma”
 - **\b[dD]ia\b**

Aula 3 - 04/09/2008

7

Exemplo 2

- Expressões regulares para preços:
 - **[\$[0-9]+**
 - Não inclui centavos
 - **[\$[0-9]+\.[0-9][0-9]**
 - Obriga o uso de centavos
 - **[\$[0-9]+(\.[0-9][0-9])?**

Aula 3 - 04/09/2008

8

Precedência e Uso de Parênteses

- Como especificar “lição” ou “lições”?
 - A expressão **lição|ões** não funciona porque encontrará “lição” e “ões” já que as strings tem menor precedência que o operador “|”
 - Temos que escrever **liç(ão|ões)**, que faz o operador “|” operar somente sobre o que está em parênteses.
- Precedência em expressões regulares
 - Parênteses
 - Contadores (*, +, ?, {}, {})
 - Sequências e âncoras (**the ^my end\$**)
 - Disjunção |

Aula 3 - 04/09/2008

9

Memória

- Podemos usar números para nos referir a elementos que apareceram anteriormente em parênteses.
 - Isso permite uma “memória”
- Exemplos:
 - **b(w+)h\1**
 - Gera as strings boohoo e baha, mas não booha.
 - **The (.*)er they (.*) the \1er we \2**

Formalmente expressões regulares não tem memória, essa é uma característica “estendida”.

Aula 3 - 04/09/2008

10

Exemplo ELIZA

```
User: Men are all alike.
ELIZA: IN WHAT WAY?
User: They're always bugging us about something or other.
ELIZA: CAN YOU THINK OF A SPECIFIC EXAMPLE?
User: Well, my boyfriend made me come here.
ELIZA: YOUR BOYFRIEND MADE YOU COME HERE.
User: He say's I'm depressed much of the time.
ELIZA: I'M SORRY TO HEAR THAT YOU ARE DEPRESSED.
```

Implemented with regular expression substitution!

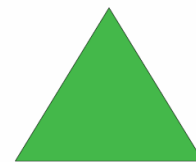
```
s/. "I'm (depressed)sad) ./I AM SORRY TO HEAR THAT YOU ARE \1/
s/. "always ./CAN YOU THINK OF A SPECIFIC EXAMPLE?/
```

Aula 3 - 04/09/2008

11

Três Visões

Regular Languages
the accepted strings



Finite-state Automata
machinery for accepting

Regular Expressions
a way to type the automata

Aula 3 - 04/09/2008

12

Autômatos Finitos (FSAs)

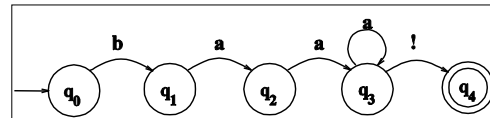
- Terminologia: Máquinas de estado finito, FSA (*Finite State Automata*), Autômatos finitos.
- Expressões regulares são uma maneira de especificar a estrutura de autômatos finitos.
- FSAs e seus “parentes próximos” formam o cerne da maioria dos algoritmos de processamento de linguagem natural.

Aula 3 - 04/09/2008

13

Intuição: FSAs como grafos

- Começando com a “língua das ovelhas” do livro
– **baa+!**

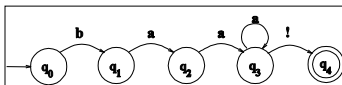


Aula 3 - 04/09/2008

14

FSA da “ovelha”

- Podemos dizer o seguinte sobre esse FSA
 - Tem 5 estados
 - Pelo menos b, a e ! estão no seu alfabeto
 - q0 é o estado inicial
 - q4 é o estado final
 - Tem 5 transições



Aula 3 - 04/09/2008

15

Mais formalmente: Definindo um FSA

- Podemos especificar um FSA enumerando
 - O conjunto de estados: Q
 - Um alfabeto finito: Σ
 - Um estado inicial q_0
 - Um conjunto F de estados finais $F \subseteq Q$
 - Uma função de transição $\delta(q, i)$ que mapeia de $Q \times \Sigma$ para Q

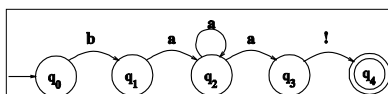
Aula 3 - 04/09/2008

16

Outra visão

- Tabela de estado-transição

State	Input		
	b	a	!
0	1	0	0
1	0	2	0
2	0	3	0
3	0	3	4
4:	0	0	0



Aula 3 - 04/09/2008

17

Reconhecimento

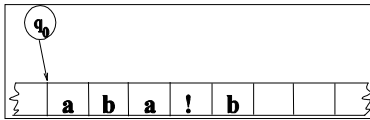
- Reconhecimento é o processo de determinar se uma string deve ser aceita por um autômato.
- Ou... é o processo de determinar se uma string está na linguagem que a máquina define.
- Ou... é o processo de determinar se uma expressão regular é compatível com a string.

Aula 3 - 04/09/2008

18

Reconhecimento

- Tradicionalmente, (idéia de Turing) o processo de reconhecimento utiliza uma fita.



Aula 3 - 04/09/2008

19

Reconhecimento

- Comece no estado inicial
- Examine a entrada atual
- Consulte a tabela de estado-transição
- Vá para o novo estado e atualize o ponteiro da tabela.
- Até que a fita acabe.

Aula 3 - 04/09/2008

20

Reconhecimento Determinístico

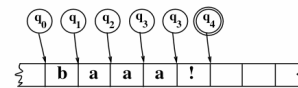
```
function D-RECOGNIZE(tape, machine) returns accept or reject
  index  $\Leftarrow$  Beginning of tape
  current-state  $\Leftarrow$  Initial state of machine
  loop
    if End of input has been reached then
      if current-state is an accept state then
        return accept
      else
        return reject
    elseif transition-table[current-state, tape[index]] is empty then
      return reject
    else
      current-state  $\Leftarrow$  transition-table[current-state, tape[index]]
      index  $\Leftarrow$  index + 1
  end
```

Aula 3 - 04/09/2008

21

Acompanhando a execução de D-Recognize

State	Input		
	b	a	!
0	1	0	0
1	0	2	0
2	0	3	0
3	0	3	4
4	0	0	0



Aula 3 - 04/09/2008

22

Pontos Principais

- Determinístico significa que a cada instante do processamento há uma única ação a ser feita.
- D-recognize é um interpretador simples baseado em tabela.
- O algoritmo é universal para todas as linguagens não-ambíguas.
 - Para mudar a máquina, mudamos a tabela.

Aula 3 - 04/09/2008

23

Pontos Principais

- Logo encontrar strings definidas por expressões regulares pode ser feito
 - traduzindo a expressão para um autômato (tabela) e
 - passando a tabela para um interpretador semelhante ao D-Recognize.

Aula 3 - 04/09/2008

24

Linguagens Formais

- Linguagens formais são conjuntos de strings compostas de símbolos pertencentes a um conjunto finito de símbolos.
- FSAs definem linguagens formais (sem ter que enumerar todas as strings da linguagem)
- O autômato pode ser executado para **gerar** todas as strings da linguagem.

Aula 3 - 04/09/2008

25

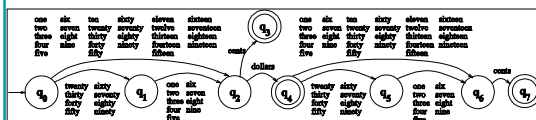
Linguagens Formais

- FSAs podem ser vistos a partir de duas perspectivas:
 - “Aceitadores” que dizem se uma string pertence ou não à linguagem.
 - Geradores que produzem todas e somente as strings que pertencem à linguagem.

Aula 3 - 04/09/2008

26

Dólares e Centavos



Aula 3 - 04/09/2008

27

Resumo

- **Expressões regulares** são representações compactas de FSAs
- **Reconhecimento** é o processo de determinar se uma string está na linguagem definida por uma autômato.
 - Reconhecimento é uma tarefa simples com autômatos determinísticos.
- Próxima aula: autômatos não-determinísticos (final do capítulo 2).

Aula 3 - 04/09/2008

28

Primeira Lista de Exercícios

- Entregar até o dia 16/09 na aula.
- Exercícios do Livro
 - 2.1
 - 2.3
 - 2.6
 - 2.8
 - 2.10

Aula 3 - 04/09/2008

29