

## Linguagens Formais

UNICAP

Eduardo Araújo Oliveira  
<http://sites.google.com/site/eaoufpe>



## Bibliotecas de Expressões

- Praticamente todas as linguagens de programação possuem bibliotecas de expressões regulares
- De modo geral, a idéia é comparar
  - uma string que representa a expressão
  - com outra string que precisa ser validada

## Bibliotecas de Expressões

- Criam um reconhecedor para a expressão regular dada
- Como funcionam
  - Convertem a expressão para um AFε, depois, opcionalmente, para um AFD (mais eficiente)
  - Para testar uma string, simulam o processo de reconhecimento no autômato

## Aplicações

- Usos de expressões regulares
  - Validar se uma string é um endereço de e-mail bem formado
  - Validar se uma string é uma data
  - Encontrar ocorrências de e-mails (ou de datas, etc.) em um texto dado
  - Find/Replace avançado, etc.

## Sintaxe Mais Comum

- A mais comum é a usada em Perl, que tem expressões regulares como parte importante da linguagem
- Existem outras sintaxes, mas são todas parecidas com a que Perl usa
  - Pequenas diferenças

## Operadores

- União: usa-se a barra "|"
  - É o operador "+" do livro...
- Exemplos
  - **a|b** → casa com "a" ou com "b"
  - **alfa|beta** → casa com "alfa" ou "beta"

## Operadores

- Agrupamento: expressão entre parênteses
- Exemplos
  - **((ab)c)** → abc → aceita "abc"
  - **(a|b)c** → aceita "ac" ou com "bc"
  - **(pre|ca)sa** → aceita "presa" ou com "casa"

## Operadores

- Classes de caracteres: lista de caracteres entre colchetes
  - Equivale ao "|" entre todos eles
  - Pode especificar faixas de caracteres
- Exemplos
  - **[abc]** → a | b | c → aceita "a", "b" ou "c"
  - **[a-z]** → aceita "a", "b", "c" ... ou "z"

## Operadores

- Zero ou mais ocorrências:  $*$
- Uma ou mais ocorrências :  $+$
- Exemplos
  - $\mathbf{ba^*}$  → aceita "b", "ba", "baa", "baaa", ...
  - $\mathbf{a+}$  → aceita "a", "aa", "aaa", ...

## Operadores

- De  $n$  a  $m$  ocorrências:  $\{n,m\}$
- Exatamente  $n$  ocorrências:  $\{n\}$
- No mínimo  $n$  ocorrências:  $\{n, \}$
- Exemplos
  - $\mathbf{a\{2,3\}}$  → aceita "aa" ou "aaa"
  - $\mathbf{a\{3\}}$  → aceita apenas "aaa"
  - $\mathbf{a\{4,\}}$  → aceita "aaaa", "aaaaa", ...

## Operadores

- Zero ou uma ocorrência: ?
  - Dá a idéia de opcional
- Exemplos
  - **a?** → aceita "" ou "a"
  - **ab?** → aceita "a" ou "ab"
  - **aa(b|c)?** → aceita "aab", "aac" ou "aa"

## Operadores

- Qualquer caractere: usa-se ponto "."
- Exemplos
  - **.+** → aceita qualquer palavra com um ou mais caracteres
  - **a.\*** → aceita qualquer palavra que começa com a

## Operadores

- Para casar com os caracteres usados como operadores, geralmente usa-se uma barra “\” antes
- Exemplos
  - `\(x\)` → casa com a string “(x)”
  - `\?\?` → casa com a string “?”

## Aviso

- Todos os operadores vistos aqui podem ser definidos a partir dos operadores vistos na aula anterior
- Na prova, não pode usar os operadores apresentados aqui

## Biblioteca Java

- Faz parte da linguagem desde a versão 1.4
  - Pacote "java.util.regex"

```
import java.util.regex.Pattern;
public class SimpleExample {
    public static void main(String[] args) {
        boolean stringCasou;
        String regExpression = "a(b|c)*d";
        String palavra = "abccbcdbd";
        System.out.println("EXPRESSAO REGULAR: " + regExpression);
        System.out.println("PALAVRA TESTADA: " + palavra + "\r\n");
        System.out.print("A palavra casa com a expressao? ");
        stringCasou = Pattern.matches(regExpression, palavra);
        if (stringCasou) {
            System.out.println("Sim.");
        } else {
            System.out.println("Nao.");
        }
    }
}
```



## Referências

- Sintaxe de Perl
  - <http://www.cs.tut.fi/~jkorpela/perl/regexp.html>
- Pacote regex de Java
  - <http://java.sun.com/docs/books/tutorial/essential/regex/index.html>
- Biblioteca Boost.Regex
  - <http://www.boost.org/libs/regex/doc/index.html>

## Linguagens Formais

UNICAP

Eduardo Araújo Oliveira  
<http://sites.google.com/site/eaoufpe>