

EasyRegex

Uma engine simplificada de expressões
regulares

Linguagens Formais e Automatos

Período Acadêmico 2013.2

Centro de Informática

Universidade Federal da Paraíba

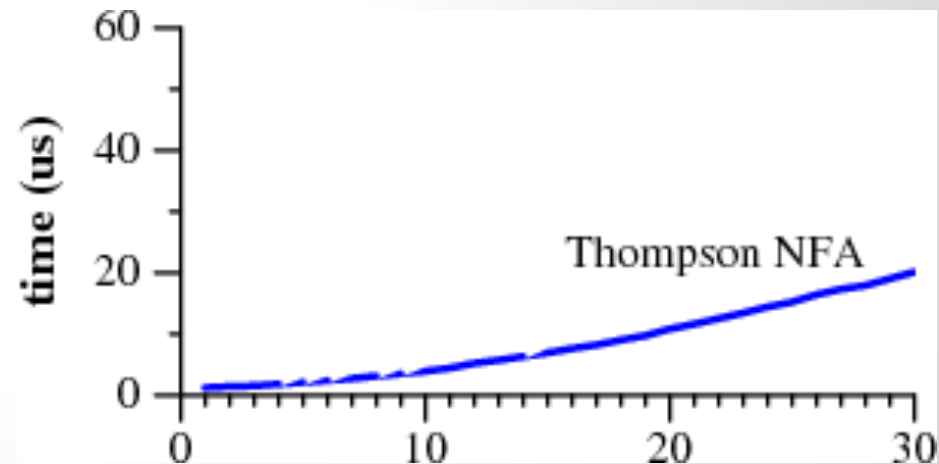
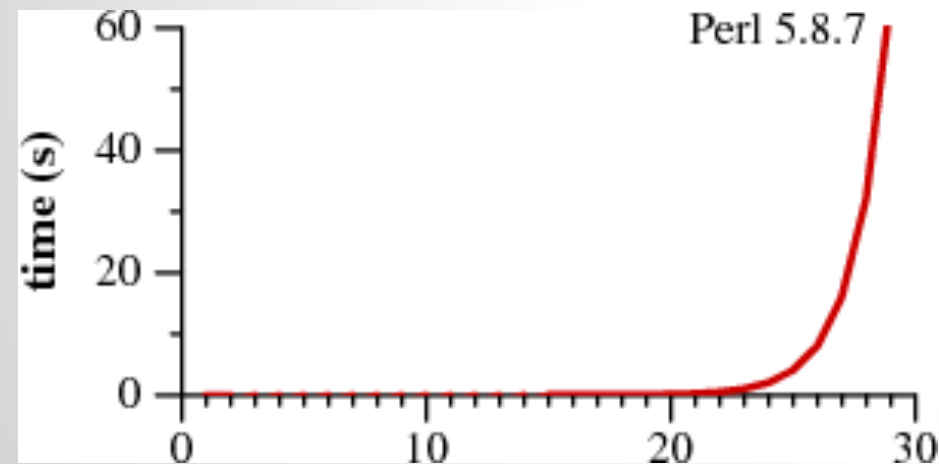
- David Leite Guilherme
- Jailson Lima
- Fernando Brito

Motivação e Objetivos

- O que são Expressões Regulares?
- Expressões regulares estão em todos os lugares;

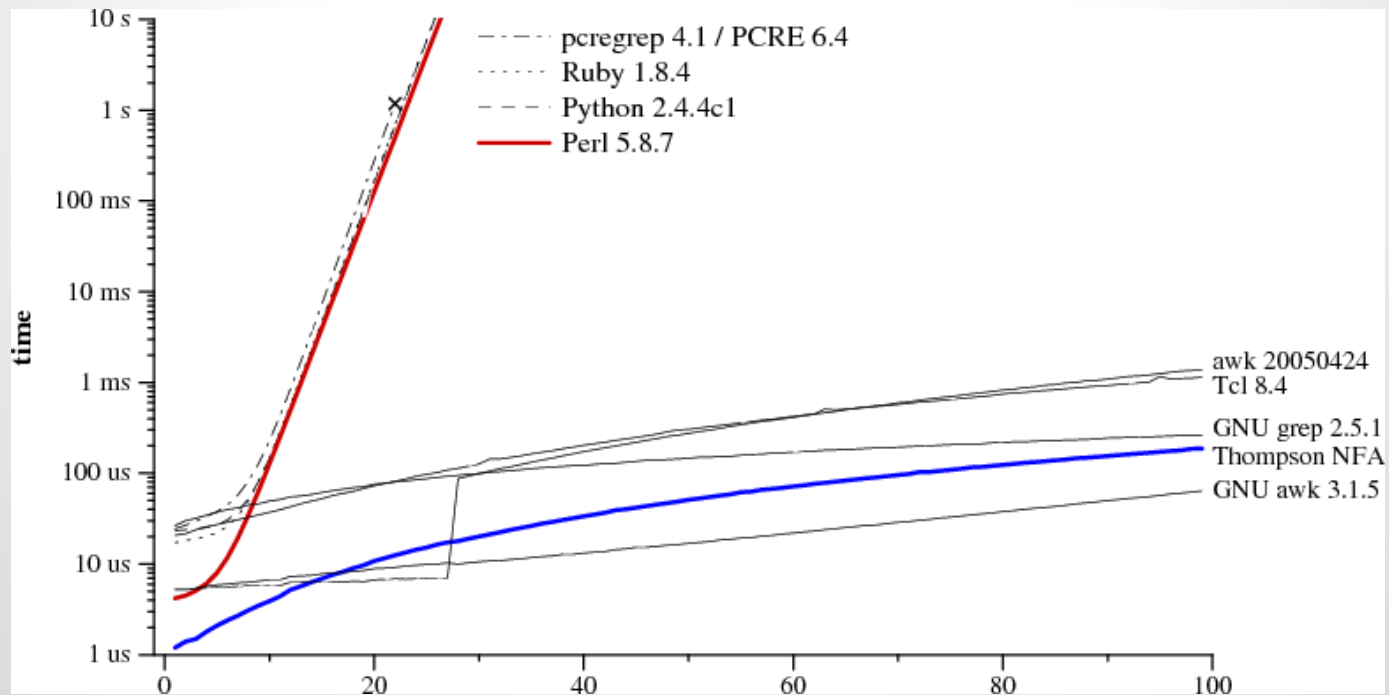
Motivação e Objetivos

- Comparação das implementações mais populares e a implementação Thompson NFA.



Motivação e Objetivos

- Comparação das implementações mais populares e a implementação Thompson NFA.



Motivação e Objetivos

- Comparações de tempo;
- Expressões Regulares Patológicas;
- Importância de estudar teoria.

Motivação e Objetivos

- Implementações são difíceis de entender e estudar;
- Nossa implementação tenta ser simples.

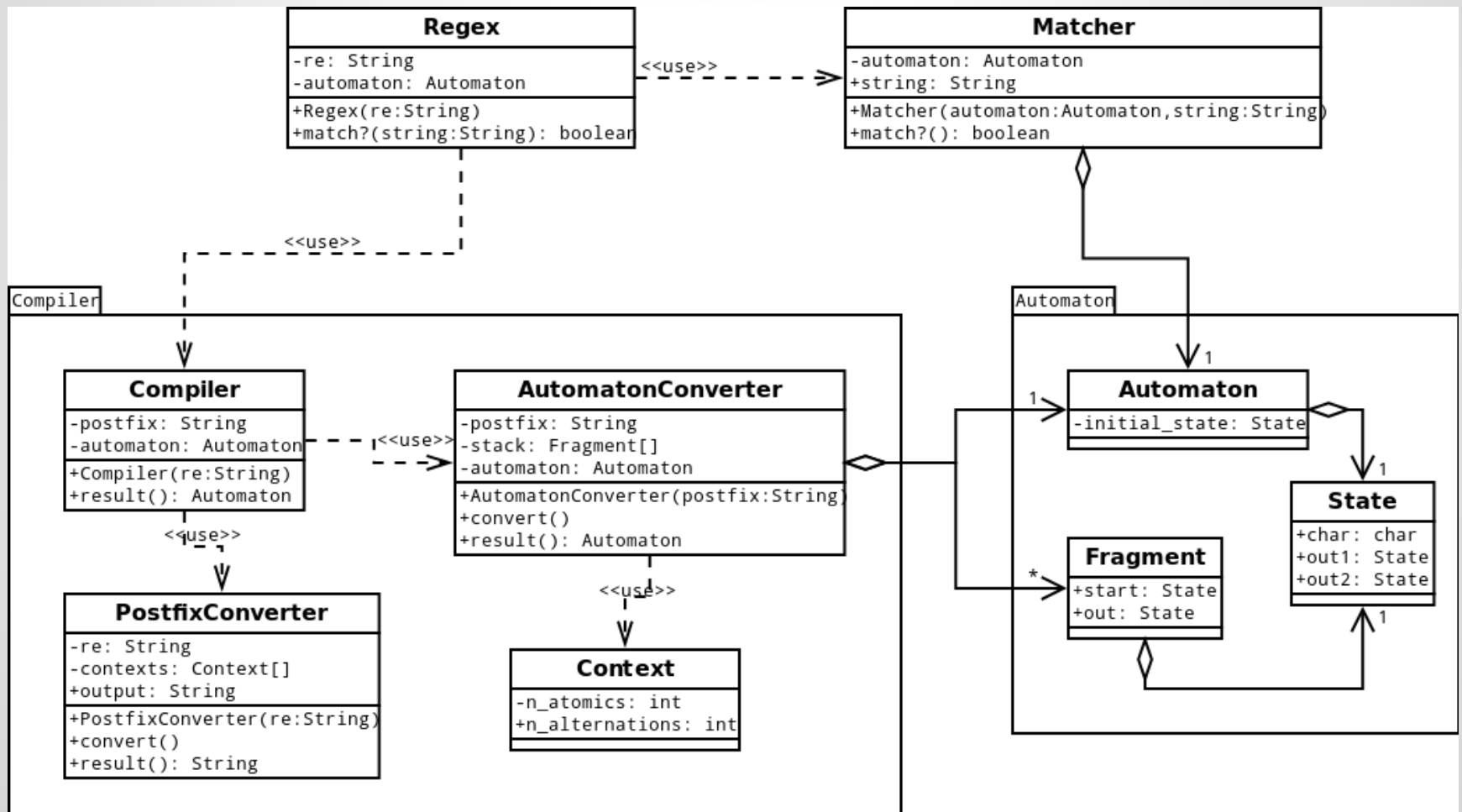
Implementação:

- Baseada em *Regular Expression Matching Can Be Simple And Fast*
- <http://swtch.com/~rsc/regexp/regexp1.html>

Implementação: passos

1. **Conversão:** converte expressão regular para forma pós-fixa. $(ab)|(ab) \Rightarrow ab.ab.|$
2. **Compilação:** cria um AFND para representar a expressão regular;
3. **Execução:** simula o AFND na string.

Implementação: arquitetura



Resultados

- Operadores básicos suportado;
- Log com os passos executados;
- Demo!

Limitações

- Pouco tempo: escopo reduzido;
- Falta de operadores avançados e *açúcares sintáticos*.

Fim

Dúvidas?

Código fonte, relatório e manuais:

http://github.com/fernandobrito/easy_regex