

# Grupo de Pesquisa em Linguagens de Programação, Verificação e Engenharia de Sistemas

Elton Máximo  
Glauber Cabral  
Leonardo Reis  
Rodrigo Ribeiro

Departamento de Computação e Sistemas (DECSI)

18 de Junho, 2015

# Projetos

- 1 Rodrigo'work
- 2 Modularização e Extensibilidade de Linguagens
- 3 Elton'work
- 4 Glauber'work

# Espaço do Rodrigo

# Era da Produtividade



- Foco na eficiência do programador
- DSLs como uma alternativa para melhorar a eficiência do programador
- Linguagens extensíveis como mecanismo para implementar e usar DSLs



# O que são Linguagens Extensíveis?

- Linguagens extensíveis são linguagens que permitem estender a própria sintaxe concreta

```
1 package syntactic;
2 public sugar Pair {
3     context-free syntax
4     "(" JavaType "," JavaType ")"
5         → Java Type
6     "(" JavaExpr "," JavaExpr ")"
7         → Java Expr
8     ...
9 }
```

---

SugarJ defining syntax

```
1 import syntactic.Pair;
2 public class Test {
3     private (String, Integer) p
4         = ("12", 34);
5 }
```

---

Using Pair syntax

# O que são Linguagens Extensíveis?

- Linguagens extensíveis são linguagens que permitem estender a própria sintaxe concreta

```
1 package syntactic;
2 public sugar Pair {
3     context-free syntax
4     "(" JavaType "," JavaType ")"
5         → JavaType
6     "(" JavaExpr "," JavaExpr ")"
7         → JavaExpr
8     ...
9 }
```

---

SugarJ defining syntax

```
1 import syntactic.Pair;
2 public class Test {
3     private (String, Integer) p
4         = ("12", 34);
5 }
```

---

Using Pair syntax

# Como Essas Características Dinâmicas Afetam o Parsing?

- Necessidade de modificar o parser de forma dinâmica, durante a análise da entrada

```
1 package syntactic;
2 public sugar Pair {
3     context-free syntax
4     ("JavaType ", " JavaType") "
5         → JavaType
6     ("JavaExpr ", " JavaExpr") "
7         → JavaExpr
8     ...
9 }
```

---

SugarJ defining syntax

```
1 import syntactic.Pair;
2 public class Test {
3     private (String, Integer) p
4         = ("12", 34);
5 }
```

---

Using Pair syntax

# As Teorias de Parsing Suportam Modificação Dinâmica?

- Principais avanços recentes na área não tratam de modificações dinâmicas
  - ▶ PEG, LL(\*), Adaptive LL(\*), SGLR, YAKKER
- Trabalhos que lidam com modificação dinâmica das regras têm eficiência questionável ou não apresentam algoritmos de parsing
  - ▶ Adaptable Grammar de Christiansen; RAG; Parsing Reflective Grammars;
  - ▶ AMG; Dynamic Grammars; Evolving Grammars

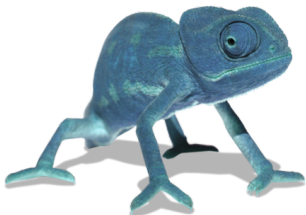




# Adaptable Parsing Expression Grammars



- Extensão de Parsing Expression Grammar;
- Modelo que permite modificações no conjunto de regras dinamicamente.



# A Pesquisa

- Desenvolvimento de um gerador automático de analisador sintático baseado em APEG:
  - ▶ Implementação eficiente;
  - ▶ Tratamento de erros;
  - ▶ Construção automática de AST e metaprogramação;
  - ▶ provas de propriedades;
- Análise (métricas) de uso de DSLs em sistema reais;
- Estudo de formalismos e mecanismo para especificação modular de linguagens
  - ▶ especificação de sintaxe e semântica;
  - ▶ implementação de DSLs como bibliotecas.

# Espaço do Elton

# Espaço do Glauber