



METRICS--

Jorge Ferreira, a64293
Pedro Cunha, a67677
MIEI:MFES:ATS
2015/16

A blue vertical sidebar on the left side of the slide, featuring a repeating pattern of white line-art icons. These icons include a clock, a checkmark, a puzzle piece, a tag, a speech bubble, a target with a fork, a document, a pie chart, a lightbulb, a magnifying glass, and a smartphone.

Agenda

- ▶ Introdução/Contextualização
- ▶ Funcionalidades
- ▶ Implementação
- ▶ App Demo
- ▶ Conclusões/Trabalhos Futuros

Métricas → Compreender questões pertinentes sobre o código desenvolvido.

Smells → Dificultam a compreensão

Refactoring → Resolve problemas de compreensibilidade

Recurso a ANTLR (Árvore sintática) e TOM (para travessia da mesma)

Funcionalidades

- ▶ Medição de métricas sintáticas (Linhas de código,...)
- ▶ Medição de métricas de complexidade (Halstead e McCabe)
- ▶ Detecção de *smells* (Negação de Condições, Excesso de Linhas...)
- ▶ *Refactoring* (Troca de Instruções, Eliminação de variáveis)
- ▶ GUI baseada em *Swing*

Implementação

- ▶ Extração de métricas segue uma estratégia global que preenche as estruturas de dados(Programa e Função)
- ▶ *Smells* partem da comparação dos valores obtidos com valores de referência
- ▶ Valores de referência : Valoração pesada (Repositório e Comunidade)
- ▶ *Refactoring* segue uma estratégia por medida implementada.

Imp. Extração Métricas

```
While(_,_,_,_,_,_,_,_) -> {  
    auxFunc.incLines(1);  
    auxFunc.incWhiles();  
    auxFunc.adicionaOperador("While");  
    auxFunc.adicionaOperador(" ");  
    auxFunc.adicionaOperador("(");  
}  
  
For(_,_,_,_,_,_,_,_,_,_,_,_) -> {  
    auxFunc.incLines(1);  
    auxFunc.incFors();  
    auxFunc.adicionaOperador("For");  
    auxFunc.adicionaOperador(" ");  
    auxFunc.adicionaOperador("(");  
}
```

```
private static int contaArgumentos(Argumentos args){  
    %match(args){  
        ListaArgumentos(arg1, argsTail*) -> {  
            return `contaArgumentos(arg1) + `contaArgumentos(argsTail*);  
        }  
  
        Argumento(_,_,_,id,_) -> {  
  
            auxFunc.adicionaOperando(`id);  
            auxFunc.incLocalVars(`id);  
  
            return 1;  
        }  
    }  
    return 0;  
}
```

Imp. Refactoring

```
%strategy refactorCondNeg() extends Identity(){
  visit Instrucao {

    If(c1,c2,c3,Nao(e),c4,c5,then,els) -> {
      if(`els != `Exp(Empty())){
        return `If(c1,c2,c3,e,c4,c5,els,then);}
      else
        return `If(c1,c2,c3,Nao(e),c4,c5,then,els);
    }
  }
}

public static Declaracoes testeRemoveVars(List<String> unusedVars, Declaracoes decls) {
  %match(decls) {

    ListaDecl(decl, declss*) -> {

      %match(decl){
        d@Decl(id,_,_,_,_) -> {
          if (unusedVars.contains(`id)){
            return testeRemoveVars(unusedVars, `declss);
          }
          else{
            return `ListaDecl(d, testeRemoveVars(unusedVars, declss));
          }
        }
      }
    }

  }
  return decls;
}
```

Utilização de Tom permitiu a diminuição de esforço programático, em oposição a escrita explícita na gramática ANTLR

GUI permite uma fácil e intuitiva visualização dos dados

Aumentar o catálogo de métricas, smells e refactoring

Rever a modularização do programa (Estratégias)



App Demo



METRICS--

Jorge Ferreira, a64293
Pedro Cunha, a67677
MIEI:MFES:ATS
2015/16