UNIVERSIDADE DO MINHO

MESTRADO INTEGRADO EM ENGENHARIA INFORMÁTICA

MÉTODOS FORMAIS EM ENGENHARIA DE SOFTWARE

VERIFICAÇÃO FORMAL

DEPARTAMENTO DE INFORMÁTICA
ESCOLA DE ENGENHARIA

David Moreira

2016/2017

ESTRUTURA DA APRESENTAÇÃO

- Introdução
- Linguagem Alvo
- Infraestrutura de Desenvolvimento
- Linguagem de Especificação
- Árvore de Sintaxe
- Condições de Verificação
- Validação
- Conclusões e Trabalho Futuro

INTRODUÇÃO

- Verificar programas
- Parser para uma *simple language (sl)* imperativa
- Contruir árvore de sintaxe
- Gerar condições de verificação
- Validar condições de verificação

LINGUAGEM ALVO

- Variáveis do tipo inteiro
- Expressões do tipo inteiro e *boolean*
- Atribuições
- Sequências de instruções
- Estruturas condicionais (*if then else*)
- Estruturas cíclicas (*while do*)

```
begin:
    while x < 1000 do:
        x = x + 1;
    end
end
```

INFRAESTRUTURA DE DESENVOLVIMENTO

- *ANTLR*, para desenvolver a gramática e gerar o *parser* da *simple language*
- *GOM*, para ajudar a gerar a árvore de sintaxe
- *TOM*, em ambiente *Java*, que juntamente com o *parser* gerado, produzir valores conforme o *match*
- *Z3*, para tentar validar as condições de verificação geradas

LINGUAGEM DE ESPECIFICAÇÃO

- Pré-condição: "*pre*"
- Invariante: "{" ... "}"
- Pós-condição normal: "*postn*"
- Pós-condição excecional: "*poste*"

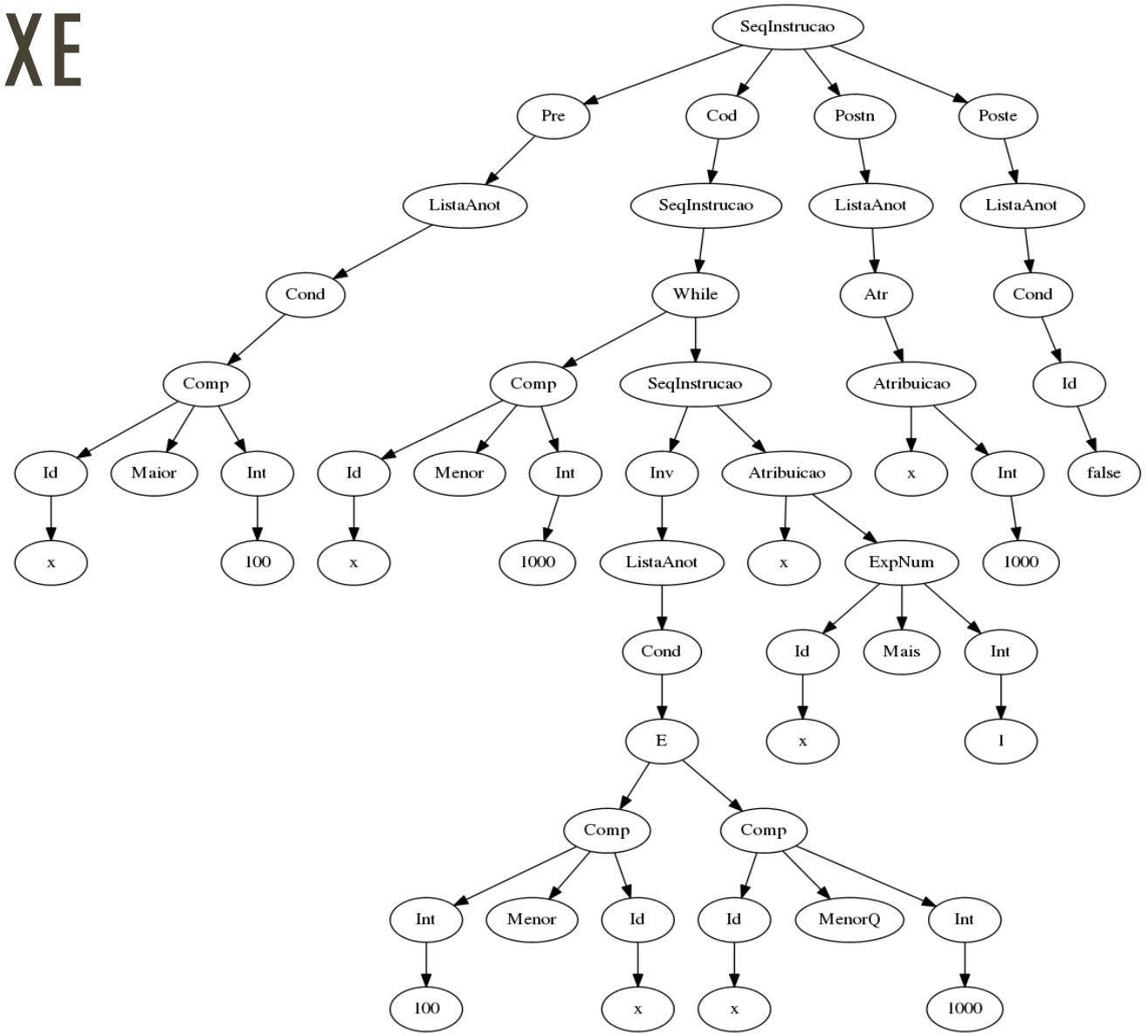
```
pre x > 100

begin:
    while x < 1000 do:
        {100 < x && x <= 1000}
        x = x + 1;
    end
end

postn x = 1000
poste false
```

ÁRVORE DE SINTAXE

- Árvore de sintaxe (*AST*) do exemplo anterior
  - Gerada através do ficheiro *.dot* produzido



CONDIÇÕES DE VERIFICAÇÃO

- Condições de verificação geradas para o exemplo anterior:

VC1: `x>100 => 100<x and x<=1000`

VC2: `100<x and x<=1000 and x<1000 => 100<x+1 and x+1<=1000`

VC3: `100<x and x<=1000 and not(x<1000) => x=1000`

VALIDAÇÃO

- Tentativa de validação das condições de verificação geradas, para o exemplo dado
  - Após traduzir para formato *SMT-LIB*

```
$> z3 ../output_files/while.sl.smt2
$> sat
```

CONCLUSÕES E TRABALHO FUTURO

- Etapas principais alcançadas
- Validar aplicação para um maior número de testes
- Concluir implementação do tratamento de exceções

UNIVERSIDADE DO MINHO

MESTRADO INTEGRADO EM ENGENHARIA INFORMÁTICA

MÉTODOS FORMAIS EM ENGENHARIA DE SOFTWARE

VERIFICAÇÃO FORMAL

DEPARTAMENTO DE INFORMÁTICA
ESCOLA DE ENGENHARIA

David Moreira

2016/2017