

Objective-00

Análise Semântica

Cezar Augustus Signori

João Paulo Pizani

José João Junior



Ferramentas utilizadas

- ANTLR
 - Plugin "ANTLR Eclipse"
 - Utilização de uma "Tree Grammar" para ligar verificações a sub-árvores da AST.
- Código gerado: Java
- Junit para testes de unidade



Verificações implementadas

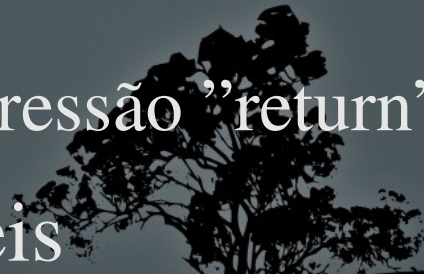
- Checagem de tipos *dentro* de expressões
 - Agora resume-se à igualdade dos tipos
 - Expandir para *compatibilidade*
 - Parâmetros formais vs. Passados
 - Checagem de atribuições
 - Tipo do lado esquerdo = Tipo do lado direito
 - Tipo de retorno de uma função
 - Retorno declarado vs. Tipo da expressão "return"
 - Checagem de declaração de variáveis
- 

Tabela de símbolos

- `HashMap<String, SymbolTableEntry>`
 - Associa nomes a registros
- Um registro `SymbolTableEntry` tem a forma:
 - `<type = PrimitiveType, params = List<PrimitiveType>>`
- No caso de uma variável, a lista `params` é vazia



Testes

- tests/TreeTests.java
- tests/WalkerTests.java
- Também pode ser feita a análise semântica dos programas exemplo em testPrograms/



Falta implementar

- Escopos locais
 - Atualmente todos os identificadores pertencem ao mesmo escopo global
- Reconhecimento dos parâmetros formais
 - Os nomes dos parâmetros devem ser reconhecidos como declarados

