Analisador Semântico para a Linguagem lang

Dupla

- Eduardo Vieira Marques Pereira do Valle - 201665554C
- Matheus Brinati Altomar - 201665564C

Arquivos Fonte

- Foram criados novas classes para armazenar tipos, com a intenção de serem utilizadas no TypeVisitor, para identificação dos diferentes tipos disponíveis na linguagem e armazenamento dos tipos presentes no programa que está sendo lido:
    - STipo.java
    - STyArray.java
    - STyBool.java
    - STyFloat.java
    - STyFunc.java
    - STyInt.java
    - STyData.java
    - STyNull.java
    - STyVar.java
    - STyChar.java
- TypeVisitor.java para visitar os nós da arvore AST e realizar a análise semantica.
- TestType.java para realizar a bateria de testes do analisador semantico.

Estratégias Utilizadas

Para implementar o analizador semântico da linguagem lang consideramos:

- Todo comando de chamada de função que associa o retorno a variáveis deve associar cada retorno da função a uma variável;
- Para utilizar um tipo data deve ter sido definido anteriormente;
- O comando read só é utilizado para ler valores inteiros.
  Para verificar se uma função tenha retorno, caso tenha sido definido em sua declaração, percorremos seus comandos é verificamos se teria o comando return, caso não tenha verifica nos comandos if-else se teria o comando return em ambos resultados da expressão condicional.

Estruturas Utilizadas

- “datas” seria um hashmap de strings e STyData a fim de armazenar os tipos datas do programa;
- “localEnv” seria um hashmap de strings e STipo a fim de armazenar os tipos ligados a determinadas variáveis do escopo atual;
- “funcs” seria um hashmap de strings e ArrayList<STyFunc> a fim de armazenar as funções do programa, sendo necessário um ArrayList<STyFunc> para guardar funções sobrecarregadas;
- “tempFunc” um Objeto do tipo STyFunc para armazenar o tipo da função atual;
- “tyint, tyfloat, tybool, tychar, tynull, tyvar” para representarem instâncias de cada um dos tipos básicos do sistema de tipos.
- A variável stk, uma pilha de STipo, foi criada como a pilha para manter todos os tipos que estão sendo analisado durante a execução do programa;

Compilação

Todos os comandos a seguir devem ser executados dentro do terminal dentro da pasta lang. O primeiro comando para compilar todos os arquivos .java encontrados na pasta do programa seria este:

- javac -cp .:antlr-4.8-complete.jar ast/*.java parser/*.java visitors/*.java tipos/*.javaLangCompiler.java -d .

Por fim, basta usar o comando abaixo para executar os arquivos gerados pelo comando anterior, a fim de testar os arquivos encontrados na pasta certo:

- java -classpath .:antlr-4.8-complete.jar lang.LangCompiler -byt