

RELATÓRIO 4 DE TEORIA DOS COMPILADORES



Felippe Rocha Lôbo de Abreu - 201765185AC

Nathan Toschi Reis - 201865064C

Pendências)

Finalmente conseguimos implementar a sintaxe e semântica do “Data” e “Funções”, além de menores pendências como o uso do quebra-linha e alguns chars específicos. Também conseguimos instanciar as funções, datas e criar vetores.

Decisões)

Adicionamos Hashs para armazenamento das declarações dos Datas e Funções.

Para conseguir priorizar a execução da main, armazenamos todas as funções e após isso chamamos a main, e as funções são processadas conforme forem chamadas pela main. Usamos uma variável booleana para sinalizar que a main já foi visitada.

Criamos uma classe chamada Tupla, para representar as declarações dos Datas e Parâmetros que nos auxiliou no armazenamento das variáveis e parâmetros passados, nela armazenamos informações que julgamos serem importantes permanecerem juntas.

Como por exemplo, a declaração dos parâmetros de uma função e seus tipos e as declarações de Datas e seus atributos.

Para auxiliar no retorno na chamada de função por expressão também criamos uma lista que armazena os return de uma função e no final a usamos para retornar o return requisitado, fizemos algumas alterações e conseguimos mantê-la mesmo quando em recursão.

Ainda sobre return tivemos dificuldade para finalizar o processamento quando já havia alcançado um return da função, para isso tratamos com a criação de um atributo booleano que cancela o processamento de comandos quando é encontrado o primeiro return.

Como rodar)

- 1) Descompactar o arquivo.zip e abrir o terminal dentro do mesmo da “src”
- 2) Digitar o comando no terminal: “java -jar jflex-full-1.8.2.jar parsers\lang.jflex”
- 3) Digitar o comando no terminal: “java -jar parsers\beaver-cc-0.9.11.jar -T parsers\lang.grammar”
- 4) Retornar para a pasta anterior, usando o comando “cd ..”
- 5) Remover a linha 3 do arquivo “MiniLangParser.java” dentro da pasta “parsers”
- 6) Digitar o comando no terminal: “javac -d src/bin src/beaver/*.java src/Teste.java”
- 7) Retornar para a pasta “src” usando o comando “cd src”
- 8) Digitar o comando no terminal: “java -cp bin src.Teste testes/semantica/certo/testeX.lan ” com “X” sendo o número do teste desejado.

Problemas enfrentados)

Demoramos muito tempo para conseguir fazer funcionar a chamada de funções e data, por conta dos environments internos do nosso programa. Tivemos problemas com

recursividade e ao lidar com os escopos globais e locais. Pela falta de tempo de refatorar o código, enfrentamos atualmente três principais problemas, que comprometem os testes 1, 3, 5, 6 e 7: a atribuição de valor para um array de datas, o uso de matriz e o uso de uma função como parâmetro.