

Compiladores, 2017/2018

Trabalho prático, parte 4

– Geração de Código –

Fernando Lobo

1 Introdução

Neste quarto e último trabalho iremos completar o compilador com a alteração do gerador de código. Como ponto de partida, deverá fazer download do código completo do compilador de Triangle para código TAM, disponível em:

- <http://www.dcs.gla.ac.uk/~daw/books/PLPJ/>

Visite o endereço web mencionado acima. Clique em 'Software' na barra do lado esquerdo e faça download do ficheiro `Triangle-tools-2.1.zip`.

O código completo do compilador está na pasta `tools-2.1`. Deverá incluir neste código as alterações que fez para as partes 1, 2 e 3, dos trabalhos práticos anteriores. Os ficheiros novos relativamente à parte 3 são os ficheiros que estão na pasta `TAM`, na pasta `Triangle/CodeGenerator` e também o ficheiro `Triangle/Compiler.java`. A classe `Compiler` implementada em `Triangle/Compiler.java` contém a função `main()`.

O trabalho consiste em modificar o gerador de código de forma a contemplar as extensões à linguagem introduzidas nas partes anteriores do trabalho com o comando `for` e o comando `case`.

Este trabalho não será submetido ao Mooshak. Deverá testar o programa criando programas de teste em Triangle, compilando-os, correndo o disassembler para inspecionar o código gerado, e correndo o respectivo código no interpretador de TAM para verificar se o programa faz aquilo que é suposto fazer.

2 Leitura e estudo prévio

Para a realização deste trabalho é muito importante a leitura do capítulo 6 e 7 do livro *Programming Language Processors in Java*, bem como o apêndice C do mesmo livro. (Também é muito útil estudarem o código do interpretador TAM que está em `tools-2.1/TAM`.)

3 Recomendação

Sugere-se que faça o seguinte:

1. Pegue no código fonte dos autores do livro que está em `tools-2.1/Triangle`
2. Copie o conteúdo das pastas que alterou nos trabalhos 1, 2, 3 (`AbstractSyntaxTrees`, `ContextualAnalyzer`, `SyntaticAnalyzer`) para dentro de `tools-2.1/Triangle`, mas mantenha a versão original dos ficheiros,
 - `SyntaticAnalyzer/SourceFile.java`
 - `AbstractSyntaxTrees/AST.java`
3. Para fazer o trabalho propriamente dito, terá de acrescentar alguns métodos na classe `Encoder` que se encontra em `CodeGeneration/Encoder.java`, nomeadamente aqueles que vão implementar a geração de código dos comandos `for` e `case`.

Implemente primeiro a geração de código para o comando `for`, que é bem mais simples. Após certificar-se que está correctamente implementado, dedique-se à geração de código para o comando `case`, que é mais complexo.

4 Prazo e entrega do trabalho

Prazo: 27/Mai/2018

Entrega: O trabalho deve ser entregue por email para `fernando.lobo@gmail.com` num único ficheiro ZIP que contenha a pasta `Triangle` e todas as suas subpastas. O assunto do email deve ser: `Compiladores TP4 Gdd`.
(Exemplo: `Compiladores TP4 G03`).

Nota importante: No dia 30/Mai (no horário da aula T e P) serão efectuadas as discussões individuais sobre os trabalhos. As discussões serão feitas no gabinete do professor, sala 1.64 do Edifício I. Oportunamente será colocado um horário para a discussão de cada aluno(a).