Compiladores, 2019/2020
Aula prática 13

Fernando Lobo

Exercício 1

Na tutoria electrónica está um ficheiro zip de nome `p13-tri-programs.zip` com 4 programas em Triangle. Compile cada um dos programas e faça o disassembler do código objecto respectivo. Estude o código assembler gerado e certifique-se que compreende o modo como a *Triangle Abstract Machine* (TAM) executa o programa. Para a compreensão do código assembler, é útil a consulta do apêndice C do livro *Programming Language Processors in Java*, o qual contém uma descrição da máquina abstracta TAM.

Passos para compilar, executar, e fazer o disassembler.

1. Compilar prog1.tri:
   - `java Triangle.Compiler prog1.tri`

   O resultado da compilação vai ser um ficheiro de nome `obj.tam`.
2. Executar o programa. Para tal devem invocar o interpretador de TAM, escrevendo:
   - `java TAM.Interpreter obj.tam`
3. Para ver o código TAM em assembler, devem invocar o disassembler;
   - `java TAM.Disassembler obj.tam`

Exercício 2

Mostre o estado do stack a cada instante da execução do programa `prog5.tri` (contido em `p13-tri-programs.zip`), indicando os frames e os links dinâmicos.

Exercício 3

Mostre o estado do stack a cada instante da execução do programa `prog6.tri` (contido em `p13-tri-programs.zip`), indicando os frames, os links dinâmicos e os links estáticos.

Exercício 4

Mostre o estado do stack a cada instante da execução do programa `factorial.tri` (contido em `p13-tri-programs.zip`), indicando os frames, os links dinâmicos e os links estáticos. Assuma que o input introduzido pelo utilizador é 3.