

Perfil de EL - Engenharia de Linguagens (1º ano do MEI)

Trabalho Prático 3 (TP3) de EG – Engenharia Gramatical

Ano Letivo 2022/23

1 Grafos na análise e interpretação de código fonte

Este último Trabalho Prático (TP3) surge na sequência do TP2 em que se pediu para desenvolver, usando o módulo Lark do Python, um Analisador de Código para uma evolução da Linguagem de Programação Imperativa (LPI), a qual deveria permitir declarar variáveis atômicas e estruturadas (incluindo as estruturas: *conjunto*, *lista*, *tuplo*, *dicionário*), instruções condicionais e pelo menos 3 variantes de ciclos.

Pretende-se agora enriquecer o já poderoso Analisador Estático criado, estudando também o *Comportamento* dos programas-fonte com base na construção dos vários DAG (Directed Acyclic Graph) que se usam para estudar o fluxo da execução (controlo) e dos dados (em função das dependências entre as variáveis).

Em suma, deve escrever em Python, usando o Parser e os Visitors do módulo para geração de processadores de linguagens Lark.Interpreter, uma ferramenta que analise programas escritos na sua linguagem LPI e gere em formato dot ¹ os grafos standard de análise que abaixo se pedem.

1.1 Construção de grafos para análise de código

1. **CFG (Control Flow Graph)** - Crie e represente o *CFG* para as seguintes instruções, suportadas pela sua linguagem:
 - (a) Para as estruturas cíclicas (caso não tenha tempo ou esteja com grande dificuldade considere **apenas um** dos **ciclos** - Exemplo : for, while , do-while, etc...);
 - (b) Para a estrutura **condicional** *if-else*;
 - (c) Para as instruções de **declaração**, **atribuição** e **input/output**.
2. Crie um **SDG (System Dependency Graph)** “lite”, que apenas tem em consideração o controlo de fluxo (ignorando o fluxo dos dados).

1.2 Análise de código utilizando grafos

Com o auxílio do **SDG** gerado na **alínea 2, questão 1.1**, identifique e represente as seguintes informações num formato à sua escolha:

1. A **complexidade de McCabe’s** para um determinado excerto de código.(**item obrigatório**);
2. Zonas de **código inalcançável** (grafos de ilha) (**item opcional**).

¹<https://www.graphviz.org/doc/info/lang.html>

1.3 Output

1.3.1 Sugestões

1. Adicione o novo output, em formato *html*, à ferramenta de análise de código desenvolvida no trabalho prático 2. Por exemplo gerando imagens a partir do dot.
2. Crie um documento de texto onde escreva as informações pedidas.

Como é habitual, o TP será entregue na forma de um relatório desenvolvido em \LaTeX , utilizando para isso o template de relatório que se encontra no Material de Apoio à disciplina da Blackboard.

A Exemplo de CFG para instrução if

```
digraph G {
    inicio -> "if x"
    "if x" -> "z=2"
    "z=2" -> "z=z+1"
    "if x" -> "z=z+1"
    "z=z+1" -> "fim"
    "if x" [shape=diamond];
}
```

A título de exemplo coloca-se aqui um link para o respetivo Visualizador online.