

Universidade Federal de São Carlos

Construção de Compiladores 2

LGraph: linguagem para criação e manipulação de grafos

Integrantes: Wilton Vicente Gonçalves da Cruz RA: 586889
Guilherme Lemos RA: 587010
Wellyson Freitas RA: 587125
Vitor Machado Cid RA: 564478

Ciência da Computação
2º semestre de 2016

Visão geral da linguagem

- A ideia da linguagem LGraph é auxiliar na **criação** de estruturas de **grafos**, **manipular** esses componentes e estudar algumas **propriedades** comuns a grafos. Desta forma, será possível ao usuário escrever um algoritmo de forma rápida para **gerar** e salvar, caso queira, o grafo gerado, **métricas** obtidas e **representações visuais** obtidas.

Gramática final

- Parser: ANTLR 4.
- Ver arquivo “.g4”.

```
'begin' comandos_for 'end'
;
comandos_for :
    ('if' '(' ctx_if+=expressao_if ')' cif+=corpo_if)*
    ;
corpo_if :
    'begin' coms+=comandos_if* 'end'| ('else' celse=corpo_else)?
    ;
comandos_if :
    imprimir | at=IDENT '=' atribuicao_for
    ;
atribuicao_for :
    NUM_INT | NUM_REAL | id=IDENT | STRING | nodes | edges | nodes_atributos_atribuicao | id1=IDENT '.' id2=IDENT
    ;
imprimir :
    'print' '(' (STRING | IDENT) ')'
    ;
corpo_else:
    'begin' imp+=imprimir* 'end'
    ;

expressao_if :
    exp_relacional | exp_igualdade
    ;
exp_relacional :
    (op1_int=NUM_INT | op1_real=NUM_REAL | (op1_v=IDENT '.' at1=IDENT) | op1_id=IDENT) op=op_relacional (op2_int=NUM_INT |
op2_real=NUM_REAL | op2_v=IDENT '.' at2=IDENT | op2_id=IDENT) (mais+=mais_exp)*
    ;
```

Análise semântica

- Feita usando recurso de “visitor” do ANTLR;
- Verificação de variáveis não declaradas, variáveis com identificadores já declaradas;
- Compatibilidade de tipos em atribuição, inclusive em nós com atributos;
- Verificação de tipo em comandos;
- Verificação de existência de atributos em grafo na estrutura de repetição;
- Verificação de compatibilidade de tipos em expressões relacionais e de igualdade;
- Outras....

Geração de código

- Feita usando recurso de “visitor” do ANTLR;
- Geração para código **PYTHON**;
- Uso de recursos do módulo “networkx”;
- Gerado caso nenhum erro sintático ou semântico ocorra em um arquivo “geradoPython.py”;
- Arquivo é interpretado pelo compilador e gera arquivos de acordo com os comandos gerados a partir do programa “LGraph”.
- Exemplo....

Exemplo 1

```
begin
```

```
  data grafo: graph,
```

```
  nos : nodes,
```

```
  arestas : edges
```

```
  nos = ["A", "B", 1, 2, 4, 5, "C", "D", 8.9, "O", "ex", "9"]
```

```
  arestas = [("A","B",0), ("9",5,1), ("C", 1, 0), ("ex",8.9,0),  
             (4,2,0)]
```

```
  create graph grafo (nodes = nos, edges = arestas)
```

```
  plot graph grafo
```

```
end
```

Exemplo 2

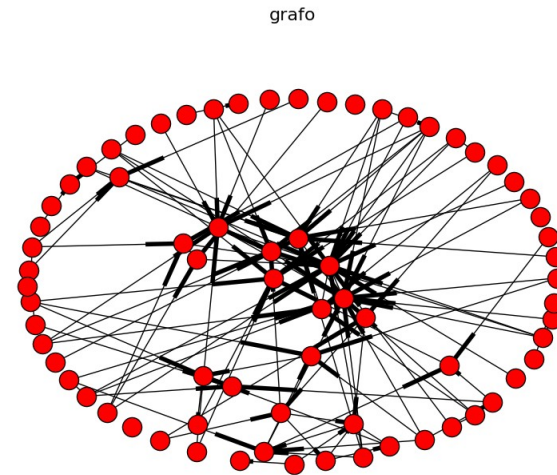
begin

data grafo: graph

read graph grafo from file "/home/wilton/Documents/futebol.paj"

plot graph grafo

end



Exemplo 3

- Exemplo completo;
- Envolve leitura de grafo a partir de arquivo, cálculo de métricas, criação manual de grafos com vértices com e sem atributos e laço repetição;
- No manual é exemplo 9.

Dificuldades

- Análise semântica;
- Geração de código;
- Problemas com tipos de grafos específicos;
- Limitar a complexidade da linguagem.