

Cmb

Bruno Azevedo*and Miguel Costa†

*Análise e Transformação de Software,
UCE30 Engenharia de Linguagens,
Mestrado em Engenharia Informatica,
Universidade do Minho*

30 de Junho de 2012

Resumo

Este documento apresenta as resolução do Trabalho Prático de Análise e Transformação de Software em que se usa as técnicas de e construção de ASTs e suas travessias tendo em conta atributos. O exercício está relacionado com a implementação da linguagem Cmb, dada nas aulas, e na criação de um Control Flow Graph (CFG), um Program Dependency Graph (PDG), System Dependency Graph (SDG) e ainda de um Directed Acyclic Graph (DAG). O resultado é um WebSite para mostrar os vários grafos criados.

*Email: azevedo.252@gmail.com

†Email: miguelpintodacosta@gmail.com

Conteúdo

1	Ambiente de Trabalho	3
2	Descrição do problema	3
3	Resolução do Problema	3
3.1	Linguagem para descrever os Mapas de Conceitos	3
3.1.1	Exemplo da linguagem	3
3.2	Gramática	4
3.3	Modelo de Dados	4
3.4	WebSite	4
4	Conclusões	6

1 Ambiente de Trabalho

Foi necessário usar um Gerador de Compiladores para gerar o nosso próprio compilador, por isso usamos o ANTLR que é também usado nas aulas. Para facilitar o processo de debugging durante a resolução do problema dado, usamos a ferramenta ANTLRWorks, que tem uma interface bastante agradável e simpática para ajudar a resolver problemas desta natureza.

A linguagem de programação adoptada foi o JAVA. De forma a tornar a nossa solução mais legível e estruturada. Para guardar a informação usamos uma Base de Dados relacional em MySQL e para o WebSite dinâmico usamos a linguagem PHP com o auxílio de JavaScript e JQuery.

2 Descrição do problema

O pretendido para este exercício era gerar um pequeno WebSite, para navegar num Mapa de Conceitos, a partir de uma linguagem simples para descrever esses mesmos mapas.

A linguagem criada por nós tem de ser validada para depois ser apresentada a informação visualmente, terá ainda de haver uma forma de mostrar as ocorrências de cada conceito.

3 Resolução do Problema

3.1 Linguagem para descrever os Mapas de Conceitos

A linguagem criada por nós baseia-se muito na linguagem CMC apresentada nas aulas, usada para descrever problemas da mesma natureza mas que não permite especificar algumas necessidades.

Com a nossa linguagem é possível definir conceitos, associações que podem haver entre eles e propriedades para os classificarem. As associações entre conceitos são criadas através de um **mapa**. Cada conceito pode ter várias instâncias e estas podem estar caracterizadas pela por propriedades.

3.1.1 Exemplo da linguagem

Listing 1: Linguagem criada

```
1 conceito("pessoa");
2 conceito("emigrante");
3 conceito("evento");
4 conceito("nascimento");
5 conceito("morte");
6 conceito("local");
7
8 associacao("e um");
9 associacao("nasceu em");
10
11 propriedade("tem nome");
12 propriedade("tem idade");
13
14 mapaConceitos(m1, "pessoa", "e um", "emigrante");
15 mapaConceitos(m2, "evento", "e um", "nascimento");
16 mapaConceitos(m3, "evento", "e um", "morte");
17 mapaConceitos(m4, "pessoa", "nasceu em", "local");
18
19 mapaConceitoProp(m5, "pessoa", "tem nome");
20 mapaConceitoProp(m6, "pessoa", "tem idade");
21
22 instancia(pAntonio, "pessoa");
```

```

23 instancia(lFafe, "local");
24 instancia(pEmigranteBrasil, "emigrante");
25
26 mapaInstancias(pAntonio, m4, lFafe);
27
28 mapaInstanciaProp(pAntonio, m5, "Antonio");
29 mapaInstanciaProp(pAntonio, m6, "20");

```

Observando a linguagem verificamos que ela é de fácil leitura, descrevendo cada instrução temos:

- `conceito("nome do conceito")` - é através desta produção onde se pode definir todos os conceitos.
- `associacao("nome da associacao")` - para relacionar os conceitos define-se as várias associações que podem existir.
- `propriedade("nome da propriedade")` - características de um conceito.
- `mapaConceitos(id, "nome do conceito pai", "nome da associacao", "nome do conceito filho")` - é aqui que se define as relações entre conceitos, é preciso ter em atenção que os nomes dos conceitos e das associações têm de estar declarados antes.
- `mapaConceitosProp(id, "nome do conceito", "nome da propriedade")` - permite dizer que um conceito pode ter uma determinada propriedade.
- `instancia(id, "nome do conceito")` - permite declarar uma instância de um conceito.
- `mapaInstancias(id Instancia Pai, id do mapaConceitos, id da instancia Filho)` - permite dizer que uma instância de um conceito está associada a outra instância de outro conceito.
- `mapaInstanciaProp(id da Instancia, id do mapaConceitoProp, "valor")` - aqui é possível definir os valores das propriedades de uma instância.

3.2 Gramática

3.3 Modelo de Dados

3.4 WebSite

O WebSite criado acaba por ser apenas uma página mas que com a ajuda de JavaScript torna-se interactiva e agradável de usar.

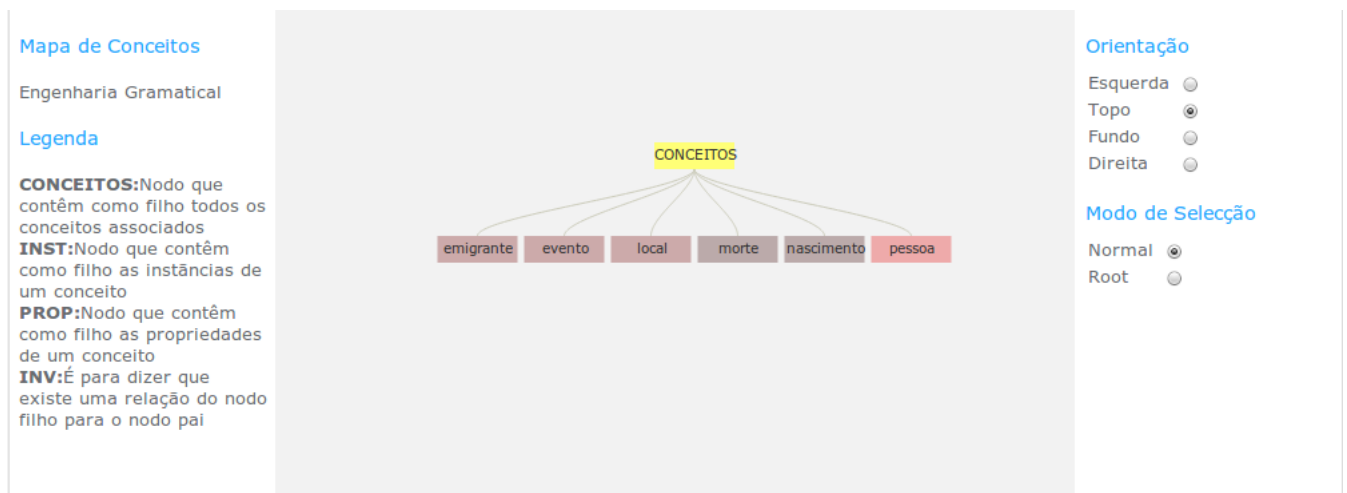


Figura 1: Imagem do site

É possível escolher a orientação da árvore e quando se clica num nodo aparece mais informação caso ela exista, tal como se pode confirmar nas imagens seguintes:

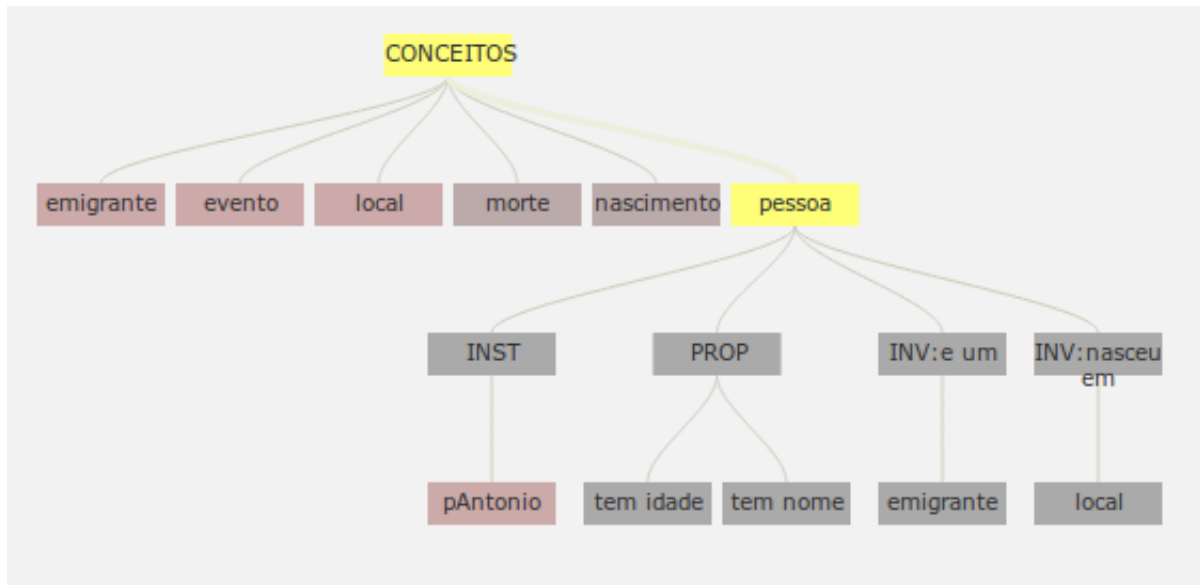


Figura 2: Mais informação disponível na árvore

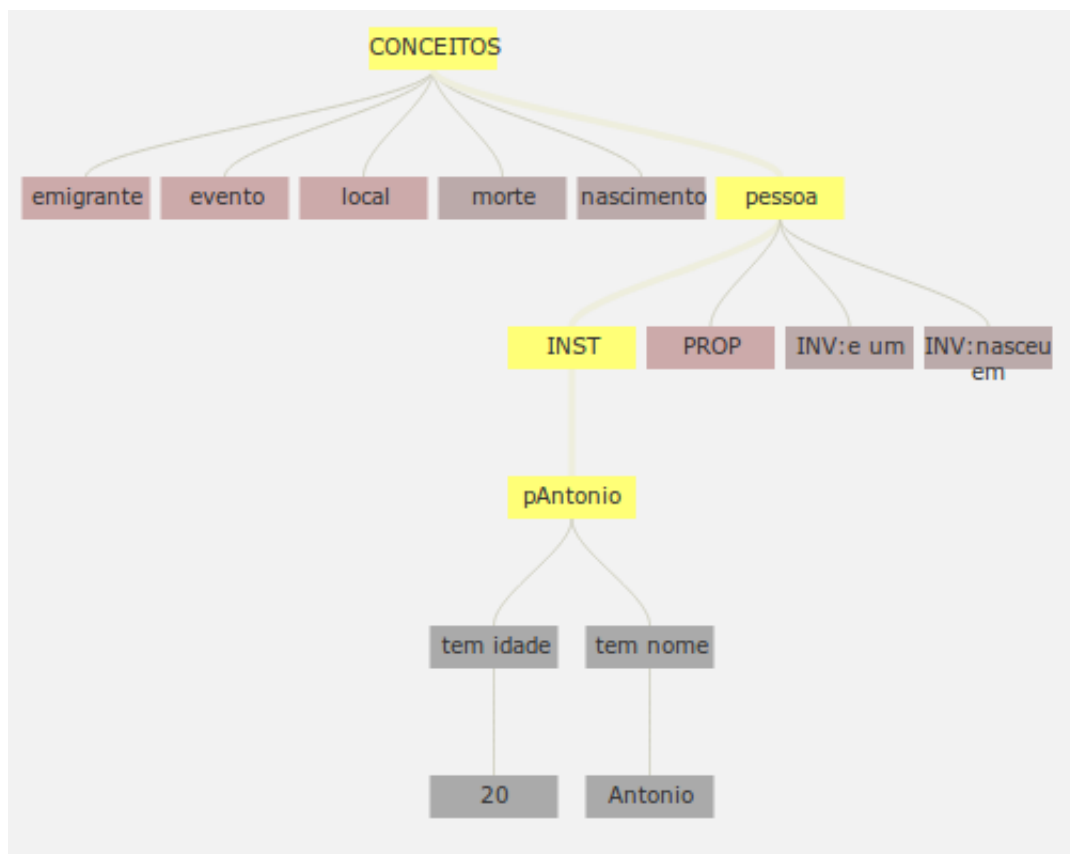


Figura 3: Detalhes de uma instância

As palavras escritas apenas em maiúsculas.... dizer que sao nossas para estruturar a infomração da arvore

4 Conclusões

A resolução deste exercício permitiu perceber melhor a forma como as linguagens podem ser úteis para gerar um programa, que dependendo do input que irá receber, o resultado final seja o esperado sem ter de estar a alterar o código do programa que é automaticamente gerado.

Um das dificuldades foi perceber como o Antlr fazia o parser das frases de forma a não haver ambiguidade e conseguir na mesma produção termos acesso ao valor de dois símbolos terminais, tal como acontece, por exemplo, quando queremos saber a dimensão do terreno, em que a solução foi inserir labels para o compilador saber qual o valor pretendido.

Serviu de consolidação da matéria dada até agora no módulo de Engenharia de Linguagens, tendo em conta que conseguimos resolver os exercícios com sucesso.