

Processamento de Linguagens
Mestrado e Curso de Especialização em Informática
Universidade do Minho

Relatório do Trabalho Prático do 1º Módulo

Jorge Filipe Pereira da Cruz

jcruz71@gmail.com
16 de Junho de 2007

Conteúdo

1	Enunciado do problema	2
2	Ferramenta ProGrammar	4
3	Conclusão	6
A	Diapositivos da apresentação	7

Capítulo 1

Enunciado do problema

Lavanda é uma Linguagem de Domínio Específico (DSL) que se destina a descrever as remessas de sacos de roupa que os Pontos de Recolha (PR) de uma Lavandaria enviam diariamente da Central (LC) para lavar. Cada saco tem um número de identificação e o nome do cliente que o deixou; o seu conteúdo está dividido em lotes. Cada lote corresponde a um tipo de peça (roupa-de-corpo, ou roupa-de-casa), um tipo de tinto (branco, ou cor) e um tipo de fio (algodão, lã e fibra), registando-se então o número de peças entregues que pertencem a esse lote. A gramática independente de contexto G, abaixo apresentada, define a linguagem Lavanda pretendida. O Símbolo Inicial é Lavanda, os Símbolos Terminais são escritos em minúsculas (pseudoterminais), ou em maiúscula (palavras-reservadas), ou entre apostrofes (sinais de pontuação) e a string nula é denotada por &; os restantes serão os Símbolos Não-Terminais.

```
p1: Lavanda --> Cabec Sacos
p2: Cabec --> data IdPR
p3: Sacos --> Saco '.'
p4: | Sacos Saco '.'
p5: Saco --> num IdCli Lotes
p6: Lotes --> Lote Outros
p7: Lote --> Tipo Qt
p8: Tipo --> Classe Tinto Fio
p9: Outros --> &
p10: | ';' Lotes
p11: IdPR --> id
p12: IdCli --> id
p13: Qt --> num
p14,15: Classe--> corpo | casa
p16,17: Tinto --> br | cor
p18,19,20: Fio --> alg | la | fib
```

A gramática abstracta é a que está descrita a seguir. Para isso eliminaram-se todos os terminais sem carga semântica (palavras-reservadas e sinais). A gramática, independente de contexto, G será simplificada eliminando produções sem alternativas em que no lado direito só aparece um terminal - neste caso: p11, p12, p13.

```
p1a: Lavanda --> Cabec Sacos
p2a: Cabec --> data id
p3a: Sacos --> Saco
```

p4a: | Sacos Saco
p5a: Saco --> num id Lotes
p6a: Lotes --> Lote Outros
p7a: Lote --> Tipo num
p8a: Tipo --> Classe Tinto Fio
p9a: Outros --> &
p10a: | Lotes
p11a: Classe--> corpo
p12a: | casa
p13a: Tinto --> br
p14a: | cor
p15a: Fio --> alg
p16a: |la
p17a: |fib

Capítulo 2

Ferramenta ProGrammar

Nesta ferramenta pretende-se:

- testar os exemplos disponíveis;
- usar a linguagem Lavanda.

Consideremos para o teste o seguinte exemplo:

```
14-06-2007 ontem  1 dani (corpo-cor-la 1, casa-cor-alg 2)
                  2 pedro (casa-br-fib 4)
                  3 celina (corpo-cor-alg 2, corpo-cor-la 3, corpo-cor-fib 1, casa-cor-alg 2,
                           casa-cor-la 3, casa-cor-fib 1)

15-06-2007 hoje   1 jorge (corpo-br-alg 1, casa-br-fib 4, corpo-cor-la 6)
```

para a gramática Lavandaria, que implementa a linguagem Lavanda, no ProGrammar:

```
//Gramática Lavandaria

grammar Lavandaria
{

Lavanda ::= {LavandaList};

LavandaList ::= Cabec Sacos;

Cabec ::= Data id;

Sacos ::= Saco [{Saco}]; // O ProGrammar não permite Left-recursion Sacos Saco

Saco ::= num id "(" Lotes;

Lotes ::= Lote Outros;

Lote ::= Tipo num;

Tipo ::= Classe "-" Tinto "-" Fio;
```

```

Outros ::= "," Lotes
        |");

Classe ::= "corpo"
        |"casa";

Tinto ::= "br"
        |"cor";

Fio ::= "alg"
       |"la"
       |"fib";

Data ::= dia "-" mes "-" ano;

num ::= numeric;

id ::= alpha;

dia ::= numeric<1,2>; //1 ou 2 dígitos

mes ::= numeric<2>; //obrigatório 2 dígitos

ano ::= numeric<4>; //obrigatório 4 dígitos

};

```

O ProGrammar é uma ferramenta de desenvolvimento de parsers. É constituída pelos seguintes componentes:

- Interactive Development Environment (IDE) - Ambiente visual para criar, testar e debugging parsers.
- Grammar Definition Language (GDL) - Notação de alto-nível para expressar a sintaxe de dados.
- Parse Engine (PE) - Componente de runtime que faz o parse dos dados de entrada.
- Application ProGrammar Interface (API) - Interface de programação para chamar o mecanismo de parse em tempo de execução (runtime).

Capítulo 3

Conclusão

O ProGrammar é uma ferramenta de desenvolvimento de Parser que:

- é de fácil utilização;
- possui um bom ambiente gráfico;
- possibilita a sua utilização em ferramentas de programação como o Visual Basic, C++ e Delphi através da sua API.

A gramática lavandaria que permitiu implementar a linguagem lavanda sofreu algumas alterações de sintaxe resultantes da implementação na ferramenta ProGrammar. O seu teste para o exemplo fornecido decorreu com sucesso.

Apêndice A

Diapositivos da apresentação