

Universidade do Minho

Trabalho Prático III

Mestrado Integrado em Engenharia Informática

Processamento de Linguagens.

2º Semestre — 2018/2019

a77523 - Alexandre Martins
a74814 - João Bernardo Freitas
a74806 - João Amorim

8 de Junho 2019
Braga

Resumo

Este projeto tem como objetivo a interação dos alunos com as ferramentas *Yacc/Bison* e *Flex* permitindo o aumento da capacidade destes relativamente à criação de gramáticas independentes de contexto (*GIC*) e desenvolvimento de processadores de linguagens suportado numa gramática tradutora (*GT*). Este relatório pretende sumarizar os esforços efetuados para alcançar o objetivo proposto utilizando as ferramentas *Yacc* e *Flex*.

Contents

1	Introdução	2
1.1	Estrutura do Relatório	2
2	Problema Proposto	2
2.1	Descrição do problema	2
3	Solução	3
3.1	Resposta às questões do enunciado	3
3.1.1	FLEX	3
3.1.2	YACC	4
4	Conclusão	11

1 Introdução

1.1 Estrutura do Relatório

O enunciado foi atribuído de acordo com o maiormenor número de aluno de todos os elementos do grupo. Como tal o relatório debruçar-se-à sobre o enunciado número quatro **Linguagem para definição de dados genealógicos**.

Começaremos por introduzir o problema proposto, passaremos para a concepção da solução e, finalmente, iremos realizar uma análise crítica ao trabalho elaborado.

Na secção enunciada como **Problema Proposto** apresentaremos o problema retratando o formato sobre o qual iremos ter de extrair informação.

Na secção **Solução**, iremos abordar os métodos utilizados na resposta ao enunciado.

Finalmente, na secção **Conclusão** enumeraremos as dificuldades sentidas na resolução do enunciado, mencionando como foram ultrapassadas.

2 Problema Proposto

2.1 Descrição do problema

O nosso problema prático aborda ficheiros de texto que incluem informações relativa a uma família, sendo que o objetivo é criar uma linguagem que consiga, através do texto-fonte, fornecer toda a informação pertinente referente a um indivíduo e sua família.

Para testar a nossa linguagem criamos o seguinte ficheiro.

```
Manuel da Silva *1977 +2011 [3]
M Maria da Silva +2009
P Joaquim Oliveira da Silva
MM Joaquina *1930
MP [45]
FOTO f.jpg
HIST h1.tex
CC 2000 [2]
Maria Felisbina *1980 [8]
F Serafim da Silva *2004
F Ana da Silva *2006 [7]{
    FOTO f1.jpg
    HIST h1.tex
}
```

3 Solução

3.1 Resposta às questões do enunciado

- Alínea A)

Para podermos processar este ficheiro temos de criar um ficheiro em *Flex*, que irá analisar lexicalmente o ficheiro, e um ficheiro em *Yacc* que irá fazer a análise sintáctica do ficheiro com ajuda do analisador léxico.

3.1.1 FLEX

A nível léxico, devem-se referir algumas palavras importantes da nossa linguagem.

- **4 n^o repetidos** para os anos
- ***** para os nascimentos
- ***c** para os nascimentos aproximados
- **+** para os falecimentos
- **+c** para os falecimentos aproximados
- **CC** para os casamentos
- **:x** para outro tipo de evento
- **nomes** para os nomes
- **XXXX xxx.xxx** para os anexos
- **M/P/MM/PP/etc** para as relações

Sempre que o analisador léxico encontra estas palavras ele irá substituí-las pelos respectivos tokens que irão ser recebidos pelo analisador sintáctico. A análise lexical é efectuada através do seguinte código.

```

    %{
    #include <stdlib.h>
    #include "y.tab.h"
    char *s;
    %}
%option noyywrap

%%
[0-9]{4}
    {yyval.s=strdup(yytext);return DATA;}
\*
    {yyval.s=strdup(yytext);return NASC;}
\*(?i:c)
    {yyval.s=strdup(yytext);return CNASC;}
\+
    {yyval.s=strdup(yytext);return FALC;}
\+(?i:c)
    {yyval.s=strdup(yytext);return CFALC;}
^(?i:cc)[ ]?
    {yyval.s=strdup(yytext);return CASA;}
\: [A-Za-z]
    {yyval.s=strdup(yytext);return TIPO;}
([A-Z][a-z]+[ ])?([a-z]+[ ])?(\\\/)?+(%[0-9]+)?
    {yyval.s=strdup(yytext);return NOME;}
[A-Z]{4}[ ].*$
    {yyval.s=strdup(yytext);return ANEXO;}
^[A-Z]{1,2}[ ]
    {yyval.s=strdup(yytext);return REL;}
\[ [0-9]+\]
    {yytext[yytext-1]='\0';yyval.s=strdup(yytext+1);return ID;}
[\\[\\(\\)\\/\\%\\{\\}\\n]
    {return yytext[0];}
(ev)
    {yyval.s=strdup(yytext);}
[ \t\r]
    {}
.
    {yyerror("Carater invlido");}
%%

```

3.1.2 YACC

Através do enunciado chegamos a várias conclusões que nos irão assistir a resolver o problema.

Nomes podem ser:

- Nome
- Nome/Nome
- Nome%Num

Eventos podem ser:

- *DATA
- +cDATA
- +DATA
- +cDATA
- cc DATA ID
- ev(DATA:TIPO)

Identificação é:

- [ID]

Anexo é:

- NOME ficheiro

E por fim as relações são:

- M
- P
- F
- MM
- etc

Com base nesta informação criamos as seguintes produções Yacc.

```

    Z: pessoa {
    ;
    pessoa: nome eventos identificacao
        | extras
        ;
    identificacao: ID
        ;
    extras: extra '\n' extras
        | %empty
        ;
    extra: pessoa
        | parentesco
        | eventos
        | anexos
        ;
    nome: NOME '/' NOME
        | NOME '%' NUM
        | NOME
        ;
    eventos: evento eventos
        | evento
        ;
    evento: NASC DATA
        | FALC DATA
        | CFALC DATA
        | CNASC DATA
        | CASA DATA identificacao
        | "ev("DATA': 'TIPO')'"
        ;
    parentesco: relacao nome
        | relacao nome eventos
        | relacao identificacao
        | relacao nome eventos identificacao '{' extras '}'
        ;
    relacao: REL
        ;
    anexos: anexo anexos
        | anexo
        ;
    anexo: ANEXO
        ;

```

Através do ficheiro criado no início deste relatório e o analisador léxico/sintático conseguimos este output.

```
#I3 nome Manuel da Silva
#I3 data-nascimento 1977
#I3 data-falecimento 2011
#I3 tem-como-M #aut1
#aut1 nome Maria da Silva
#aut1 data-nascimento 2009
#I3 tem-como-P #aut2
#aut2 nome Joaquim Oliveira da Silva
#I3 tem-como-MM #aut3
#aut3 nome Joaquina
#aut3 data-nascimento 1930
#I3 tem-como-MP #I45
#I3 FOTO f.jpg
#I3 HIST h1.tex
#F2 = #I3 #I8
#F2 data-casamento 2000
#I8 nome Maria Felisbina
#I8 data-nascimento 1980
#F2 tem-como-F #aut4
#aut4 nome Serafim da Silva
#aut4 data-nascimento 2004
#I7 nome Ana da Silva
#I7 data-nascimento 2006
#F2 tem-como-F #I7
#I7 FOTO f1.jpg
#I7 HIST h1.tex
```

- Alínea B)

Foi-nos também pedido criar uma árvore genealógica através do output da **Alínea A**, como tal criamos uma outra linguagem começando mais uma vez pelo analisador léxico.

```

^\\#F[0-9]
    {yyval.s=strdup(yytext+1);return CASA;}
([A-Z][a-z]+[ ]?([a-z]+[ ])?)+(\\[0-9]+)?
    {yyval.s=strdup(yytext);return NOME;}
-[A-Z]+
    {yyval.s=strdup(yytext+1);return REL;}
\\#aut[0-9]+
    {yyval.s=strdup(yytext+1);return ID;}
\\#I[0-9]+
    {yyval.s=strdup(yytext+1);return ID;}
nome
    {yyval.s=strdup(yytext);return LIXO;}
tem-como
    {yyval.s=strdup(yytext);return LIXO;}
data-.*[ ]{0-9}{4}
    {yyval.s=strdup(yytext);return LIXO;}
[A-Z]{4}[ ]\\.\\.+.
    {yyval.s=strdup(yytext);return LIXO;}
[\\%]
    {return yytext[0];}
[ \\n\\t\\r\\.]
    {}
.
    {yyerror("Carater invlido");}

```

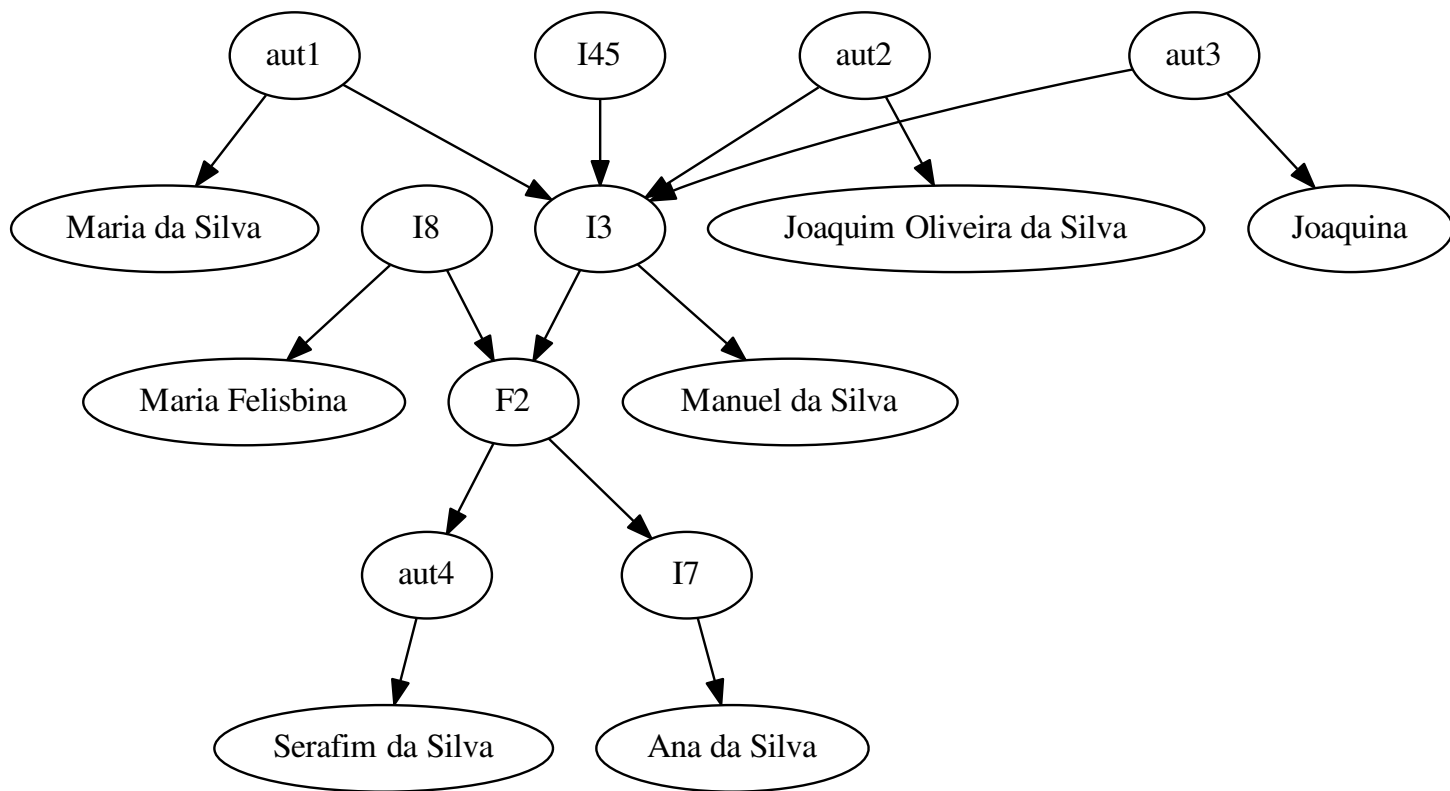
E tal como na alínea A criamos mais um analisador sintático.

```
arv: pessoas
    ;
pessoas: pessoa pessoas
        | %empty
        ;
pessoa: identificacao nome
        | identificacao relacao identificacao
        | casamento identificacao identificacao
        | casamento relacao identificacao
        | identificacao LIXO
        | casamento LIXO
        ;

identificacao: ID
              ;
casamento: CASA
           ;
nome: LIXO NOME
    ;
relacao: LIXO REL
        ;
```

Através destes ficheiros e do output da **Alínea A** obtemos o seguinte ficheiro **DOT**.

```
digraph{
rankdir=UD
I3->"Manuel da Silva";
aut1->I3;
aut1->"Maria da Silva";
aut2->I3;
aut2->"Joaquim Oliveira da Silva";
aut3->I3;
aut3->"Joaquina";
I45->I3;
I3->F2;
I8->F2;
I8->"Maria Felisbina";
F2->aut4;
aut4->"Serafim da Silva";
I7->"Ana da Silva";
F2->I7;
}
```



4 Conclusão

Visto que o objectivo deste trabalho era aumentar contacto com as ferramentas *YACC* e *FLEX*, capacidade de escrever *Expressões Regulares* e desenvolver linguagens

Porém, no decorrer da resolução deste trabalho, foram surgindo alguns desafios que foram ultrapassados, nomeadamente a criação de um analisador sintático e a criação de duas funções auxiliares que foram utilizadas de forma a contrariar algumas deficiências na nossa linguagem.

Em suma, o trabalho foi bastante benéfico para todos os elementos, sendo que aprofundou o entendimento das ferramentas **FLEX** e **YACC**.