

Processamento de linguagens (3º ano de MIEI)

Trabalho Prático 1

Relatório de Desenvolvimento

Célia Figueiredo
(a67637)

Diogo Tavares
(a61044)

Gil Gonçalves
(a67738)

5 de Abril de 2016

Resumo

Neste relatório será apresentado o desenvolvimento de um filtro de texto com a posterior aplicação sob um ficheiro BibTex, pretende-se que seja produzido um normalizador de ficheiros BibTex.

Implementou-se um filtro que permite fazer a contagem das categorias das referências bibliográficas, também foi implementado um filtro que permite a troca para chavetas do campo que está entre aspas. E ainda foi implementado um filtro que coloca os nomes dos autores escritos no formato "N. Apelido". Por fim, de modo a tornar a leitura mais fácil implementou-se um ferramenta de *pretty-printing*.

Será também mostrado um grafo que ilustra para um dado autor (escolhido pelo utilizador) todos os autores que publicam normalmente com o autor em causa. Utilizou-se a linguagem *Dot* do *GraphViz* esta, que gerou um ficheiro com um grafo de modo a que posteriormente fosse usada uma das ferramentas que processam *Dot* para desenhar o dito grafo de associações de autores.

Conteúdo

1	Introdução	2
2	Análise e Especificação	3
2.1	Descrição informal do problema	3
2.2	Especificação do Requisitos	3
2.2.1	Dados	3
3	Concepção/desenho da Resolução	5
3.1	Estruturas de Dados	5
3.1.1	Algoritmo alinea a)	5
4	Codificação e Testes	6
4.1	Alternativas, Decisões e Problemas de Implementação	6
4.1.1	Expressões Regulares	6
4.1.2	Makefile	6
4.2	Testes realizados e Resultados	6
4.2.1	Testes e Resultados alinea a)	6
5	Conclusão	8
A	Código do Programa	9

Capítulo 1

Introdução

Este trabalho envolverá o desenvolvimento de um normalizador de ficheiros *BibTex*, este é o tema do problema 2.2 do enunciado fornecido.

Enquadramento Utilização de expressões regulares e filtros de texto com o objetivo de produzir novos documentos a partir de padrões existentes no ficheiro de input.

Conteúdo do documento O presente documento contém a explicação do problema, assim como a apresentação das soluções produzidas.

Resultados Os resultados deste desafio serão as alíneas pedidas, sendo que serão apresentados em ficheiros *.html* e grafos.

Estrutura do Relatório

Este documento está dividido em seis partes No capítulo 2 faz-se uma análise detalhada do problema proposto de modo a poder-se especificar as entradas, resultados e formas de transformação.
etc. ...

No capítulo 3 será feita a conceção e No capítulo 4 pretende-se que No capítulo 5 termina-se o relatório com uma síntese do que foi dito, as conclusões e o trabalho futuro

Capítulo 2

Análise e Especificação

2.1 Descrição informal do problema

BibTex é uma ferramenta de formatação usada em documentos \LaTeX , foi criada em 1985, para a criação de citações e referências bibliográficas em documentos \LaTeX , o ficheiro (*.bib*) é uma base de dados que contém os dados bibliográficos (autor, título, ano de publicação, etc.) das fontes citadas no documento \LaTeX .

Deixámos um exemplo de um ficheiro com a extensão *.bib*:

```
@inbook{Val90a,
author = "Jos\`e M. Valen\c{c}a",
title = "Processos, {0}bjectos e {C}omunica\c{c}\~ao
({0}p\c{c}\~ao I - {MCC})",
chapter = 2,
year = 1990,
month = Oct,
publisher = gdcc,
address = um,
annote = "programacao oobjectos, proc comunicantes, espec formal"
}
```

O ficheiro BibTex contém várias categorias de referência estas inicializadas sempre pelo caracter @, deixamos aqui alguns exemplos de entradas que podem ser encontradas no ficheiro BibTex:

```
@inbook
@misc
@incollection
@inproceedings
@techreport
@unpublished
```

2.2 Especificação do Requisitos

2.2.1 Dados

Os dados fornecidos são um ficheiro *.bib*, este que é um ficheiro com as características de um ficheiro *BibTex*.

Cada tipo de categoria tem os seus campos obrigatórios, neste acaso o objetivo de uma das tarefas será pesquisar através dos campo **author** o nome do autor e transformá-lo no formato "N. Apelido".

```
@phdthesis{Mos75a,
```

```

author = "P. D. Mosses",
title = "Mathematical Semantics and Compiler Generation",
year = 1975,
school = "Oxford University",
annotate = "compilacao incremental, atributos, ambientes prog"
}

```

É também fornecido o nome de ferramentas de apoio à resolução do problema, neste caso o *Graph Viz*, que permitirá colocar gráficamente a informação dos grafos criados, sendo que tornará as iterações entre os autores mais percetíveis.



Figura 2.1: Exemplo de geração de um grafo com os nomes dos autores com recurso à ferramenta

Capítulo 3

Concepção/desenho da Resolução

3.1 Estruturas de Dados

Será necessário utilizar listas ligadas

3.1.1 Algoritmo alinea a)

Nesta alinea optou-se por utilizar listas ligadas porque o nosso objetivo é armazenar as categorias das referências do *BibTex*, estas que são strings e o objetivo é contá-las o que nos tornou a tarefa mais simples.

```
void* procura(char* c){
int flag = 1;

Est* aux;
aux = est;
while(aux!=NULL && flag && aux->cat != NULL){
if(!strcmp(minusculas(aux->cat), minusculas(c+1))){
aux->i++;
flag=0;
}
aux=aux->next;
}

if(flag){
Est* novo = (Est*) malloc(sizeof(Est));
novo->cat = strdup(c+1);
novo->i = 1;
novo->next = est;
est = novo;
}
}
```

Capítulo 4

Codificação e Testes

4.1 Alternativas, Decisões e Problemas de Implementação

4.1.1 Expressões Regulares

Alinea a)

4.1.2 Makefile

4.2 Testes realizados e Resultados

Mostram-se a seguir alguns testes feitos (valores introduzidos) e os respectivos resultados obtidos:

4.2.1 Testes e Resultados alinea a)

Utilizámos este ficheiro de teste para testar as diferentes opções que apareciam no ficheiro

```
1 @book{
2 @book{
3 @cenas{
4 @book{
5 @coisas{
6 @Book{
7 @book{
8 @BOOK{
9     @BOOK{
10     @Book{
11     @author{
12
13 @AuThor{
14 @AUTHOR{
```

Mostrámos de seguida o ficheiro *.html* produzido:

```
proceeding -> 1
mastersthesis -> 2
proceedings -> 4
misc -> 61
manual -> 13
incollection -> 6
unpublished -> 15
```


inproceedings -> 209
article -> 142
phdthesis -> 21
book -> 47
inbook -> 3
techreport -> 140
string -> 31

Capítulo 5

Conclusão

Síntese do Documento.

Estado final do projecto; Análise crítica dos resultados.

Trabalho futuro.

Apêndice A

Código do Programa

Lista-se a seguir o código *flex* que foi desenvolvido:

```
1  %{
2
3  typedef struct Est{
4      char* cat;
5      int i;
6      struct est *next;
7  }Est;
8
9  Est* est;
10
11 void* novo() {
12     est =(Est*) malloc(sizeof(Est));
13     est->next = NULL;
14     est->cat = NULL;
15     est->i = 0;
16 }
17
18 char* minusculas(char* d){
19     char* novo = strdup(d);
20     int j=0;
21     while(novo[j] != '\0'){
22         novo[j]= tolower(novo[j]);
23         j++;
24     }
25     return novo;
26 }
27
28 void* procura(char* c){
29     int flag = 1;
30
31     Est* aux;
32     aux = est;
33     while(aux!=NULL && flag && aux->cat != NULL){
34         if(!strcmp(minusculas(aux->cat), minusculas(c+1))){
35             aux->i++;
36             flag=0;
37         }
38         aux=aux->next;
39     }
40
41     if(flag){
```

```

42         Est* novo = (Est*) malloc(sizeof(Est));
43         novo->cat = strdup(c+1);
44         novo->i = 1;
45         novo->next = est;
46         est = novo;
47     }
48 }
49
50 %}
51
52 %%
53 @[^\{]* {procura(yytext);}
54 .|\n    {;}
55
56 %%
57
58 void print(){
59     FILE* fp;
60     fp = fopen("ficheiro.html", "w");
61     fprintf(fp, "<html>");
62     while(est->next!=NULL){
63         fprintf(fp, "%s -> %d<br>", est->cat, est->i);
64         est = est->next;
65     }
66     fprintf(fp, "<\html>");
67     fclose(fp);
68 }
69
70 int yywrap() {return 1;}
71 int main(){
72     est = novo();
73     yylex();
74     print();
75     free(est);
76     return 0;
77 }
78 }

```

Listing A.1: Exemplo de uma Listagem

```

1  llll sanjdb c kjnjcnjnjj mmmmmmmmmmm hhhhhhhhhhhhhhhhhhhhhhh jjjjjjjjjjjjjjjjjjjjjjjj
   kkkkkkkkkkkkkkkkkkk
2      ou então aparecer aqui neste sítio um pouco de matematica $
3      como alternativa ao anterior.
4      e aqui mais um teste  $\varepsilon$ 

```

É ainda possível importar diretamente o ficheiro:

Bibliografia