Universidade do Minho
Escola de Engenharia

Trabalho Prático 1

Mestrado Integrado em Engenharia Informática

Processamento de Linguagens
2º Semestre
2017-2018

António Jorge Monteiro Chaves,A75870
Carlos José Gomes Campos,A74745
Luis Miguel Bravo Ferraz,A70824

March 2018

Conteúdo

1 Introdução **3**

2 Processador Thesaurus 2 **4**
2.1 Análise do texto-fonte 4
2.1.1 Exemplo1 4
2.1.2 Exemplo2 4
2.1.3 Exemplo3 4
2.1.4 Conclusão da Análise 5
2.2 Ações semânticas 5
2.3 Estruturas de Dados Globais 5
2.4 Filtro de texto- Sistema de Produção Gawk 6
2.4.1 Exemplo HTML-Lista dos domínios usados 6
2.4.2 Filtro de Domínios 6
2.4.3 Filtro de Relações Inversas 7
2.4.4 Filtro de Relações 7
2.4.5 Triplos 7

3 Conclusão **8**

1 Introdução

Este trabalho tem como objetivos o aumento da experiência de uso do ambiente Linux e de algumas ferramentas que dão suporte à programação, como o sistema de produção para filtragem de texto GAWK. Através da utilização deste sistema, tivemos de por à prova os nossos conhecimentos sobre Expressões Regulares, uma vez que, tal como estava predefinido nos objetivos, através destas desenvolvemos um Processador de Linguagem Regular que filtra e/ou transforma os textos que o enunciado propõe. Foram propostos 5 enunciados, segundo as regras de atribuição destes a cada grupo, o nosso grupo ficou com o "Processador de Thesaurus 2".

2 Processador Thesaurus 2

Este enunciado consiste em analisar ficheiros ".mdic"que descrevem numa sintaxe simples as entradas "triplos termo1, relação, termo2". Com isto pretende-se criar um dicionário automaticamente(Thesaurus).

2.1 Análise do texto-fonte

Os vários ficheiros fornecidos para a realização desta tarefa, tem uma estrutura definida, no entanto, alguns deles possuem algumas exceções que irão influenciar a estratégia com que os iremos abordar.

2.1.1 Exemplo1

```
\%inv:  syn: syn
\%inv:  dom: voc

\%dom:corpo humano

\%THE:has

corpo humano: aparelho digestivo| aparelho circulatório| aparelho respiratório
corpo humano: aparelho urinário| aparelho reprodutivo
```

2.1.2 Exemplo2

```
\%inv: lida_com: profissao_associada

\%dom : medicina

\%THE<profissão : bt : lida_com

médico especialista : médico :
médico : : doentes| doenças | fármacos

andrologista :  :
```

2.1.3 Exemplo3

```
\%dom:profissões

#---( Profissão : profissãoEspecífica)--------------------------------

\%THE<profissão : nt<profissão

professor: professor do ensino superior politécnico| professor do ensino superior
```

2.1.4 Conclusão da Análise

Depois de analisados os vários exemplos, verificamos que temos de ter cuidado com alguns casos especiais. No primeiro exemplo, que é o que seguiu de base para determinarmos os padrões que seguimos, verificamos que a informação está organizada uniformente. No entanto, no segundo exemplo verificamos que a estruutra pode não estar assim tão bem organizada, nomeadamente devido aos espaços em branco e a inserção do simbolo «"nas relações. No terceiro exemplo, é inserida uma linha que funciona como uma espécie de comentário, a qual também teremos de tratar. Contudo, apesar destas exceções conseguimos definir uma estrutura, que resumidamente, consiste na existência de linhas do tipo "%inv:"(inicio de linha que contém uma ou mais relações inversas), "%dom:"(inicio de linha que contém dominio), "%THE:"(inicio de linha que contém uma ou mais relações),"%THE<"(inicio de linha que contém uma ou mais relações), "#"(inicio de linha que é um comentário) e de linhas do tipo "corpo humano: aparelho urinário| aparelho reprodutivo"(inicio de linha que contém os termos que se relacionam).

2.2 Ações semânticas

Depois da análise cuidada do texto-fonte, reparamos que em todas as linhas não vazias, existe informação que pode ser dividida em dois campos diferentes e que cada linha corresponde a um registo, ou seja,

- **FS**(Field Separator) - Foi útil definir como ":"uma vez que os termos que se relacionam estão separados por este simbolo.
- **RS**(Record Separator) - Foi útil definir como "\n "uma vez que um termo, possui todos os termos com que se relaciona na sua linha.
- **gsub** - Utilizamos esta função de modo a que quando procuramos algo a informação nos chegue apenas com aquilo que pretendemos, como por exemplo, informação sem espaços em branco.
- **match** - Para certas relações compostas, foi necessário isolar os seus componentes(relação<termo), para isso, recorremos a esta função de modo, a encontrar as relações onde o carater «"estava presente(posteriormente fez-se a divisão da relação e do termo).
- **split** - Nos texto-fonte disponibilizados, encontramos campos que tinham mais que um termo, e esta função ajudou-nos a isola-los, por exemplo.

2.3 Estruturas de Dados Globais

Durante a realização da tarefa, de modo a onter os resultados pretendidos, sentimos a necessidade de criar estruturas de modo a armazenar a informação para posteriormente a podermos organizar.

- **termos** - Array que armazena os termos presentes nos ficheiros.

- **relacoes** - Array que armazena todas as relações presentes nos ficheiros.
- **relacoesComInv** - Array que armazena todas as relações que possuem uma relação inversa.
- **inversas** - Array que armazena todas a relações inversas.

2.4 Filtro de texto- Sistema de Produção Gawk

2.4.1 Exemplo HTML-Lista dos domínios usados

Dominios	Ficheiro
corpo humano	corpoHumano.mdic
dia a dia	diaADia.mdic
desporto	diaADia.mdic
ferramentas	diaADia.mdic
cozinha	diaADia.mdic
lojas e comércio	diaADia.mdic
família	diaADia.mdic
vida	diaADia.mdic
medicina	medicina.mdic
profissões	profissoes-lojas.mdic
geografia	riospt.mdic
vestuário	vestuario.mdic

2.4.2 Filtro de Domínios

A lista dos domínios é obtida pelas linhas que correspondam com a seguinte expressão regular.

```
$1 ~/\s*%dom\s*/
```

O uso da função gsub serve para corrigir alguns erros de conversão alfabética, pois não são aceites os acentos em hiperligações, por exemplo.

2.4.3 Filtro de Relações Inversas

```
$1 ~/\s*%inv\s*/
```

A expressão regular acima dá a indicação de que os campos dessa linha definem um conjunto de duas palavras antónimas, que se encontram nos campos $2 e $3. Cada uma destas é inserida num array, em concordância com a posição, ou seja, o array que guarda $2 fá-lo no mesmo índice numérico que o array que guarda $3.

2.4.4 Filtro de Relações

```
$1 ~/\s*%THE\s*/
```

Com o separador de campos definido como o caracter ":", sempre que o primeiro campo de uma linha corresponda com a expressão regular acima o compilador tratará os campos dessa linha como relações entre termos. Assim, até uma nova indicação, todas as relações das próximas linhas serão definidas a partir desta indicação. Para evitar que "%THE"seja tido conta como relação, começa-se a leitura dos campos a partir de $2. No caso de o campo ser do tipo "relação<designação", divide se o campo pelo separador «"e retira-se o primeiro elemento. Para além de guardar a relação num array para futura referência, é enviada uma linha para o ficheiro relacoes.html com a relação e o ficheiro na qual se insere.

2.4.5 Triplos

Para a construção dos triplos atribui-se ao primeiro campo de cada linha (que não coincida com os filtros anteriores e não seja comentário) a designação de termo1, e itera-se pelos restantes campos, dividindo cada um pelo caracter "| ". A cada campo encontrado, constroem-se 3 strings: o triplo encontrado, o domínio a inserir e o respetivo nome do ficheiro e, se existir, o triplo inverso do inicial. Todos os termos dos triplos inseridos nos ficheiros .html têm ligação à sua página de pesquisa no site Wikipedia. Na primeira string são geradas, para uma entrada de tabela, as referências para as Wikis do primeiro e segundo termos, com substituição dos espaços por "_", inserindo, na coluna do meio, a relação previamente filtrada entre os termos. De seguida, é gerada a entrada de duas colunas no ficheiro Domínios.html, com a informação do Domínio em que está inserido o documento e o seu nome. Em último, é percorrida a lista de relações que possuem inversa e, em caso de coincidência, incluída, no ficheiro Triplos.html uma string semelhante à primeira, com os termos e relação inversa.

3 Conclusão

Com a elaboração deste trabalho prático conseguimos aprofundar os conhecimentos sobre Expressões Regulares e sobre a ferramenta GAWK, uma vez que a estrutura dos ficheiros nos obrigou a tratar a informação de uma forma diferente daquela que estavamos habituados. Também ficamos a saber que devido ao poder de processamento de texto do GAWK, podemos ficar a contar com uma nova ferramenta para futuros trabalhos. No entanto, as nossas maiores dificuldades estiveram nos diversos pormenores que os diferentes ficheiros continham e uniformizar o código de modo a não conter linhas desnecessárias.