Processamento de Linguagens (3º ano de Engenharia Informática)

Trabalho Prático Nº3

Relatório de Desenvolvimento

Ana Ribeiro	Carla Cruz	Jéssica Lemos
(a82474)	(a80564)	(a82061)

10 de Junho de 2019

Resumo

O trabalho que será apresentado tem como principal objetivo implementar uma ferramenta de análise de documentos em formato *Thesaurus ISO 2788*. Para tal efeito, utilizamos o par de ferramentas geradoras flex/yacc de forma a desenvolver o reconhecedor léxico e sintático pretendido.

Conteúdo

Capítulo 1

Introdução

No terceiro trabalho prático da Unidade Curricular de *Processamento de Linguagens*, o exercício que nos foi atribuído foi o 3, que consiste na realização de uma ferramenta capaz de analisar documentos em formato *Thesaurus* ISO 2788 T(2788). A informação extraída de ficheiros deste formato deve ser colocada de um modo organizado em ficheiros HTML.
Deste modo, pretende-se que seja elaborada uma gramática independente de contexto, bem como o respetivo analisador léxico, utilizando a ferramenta *Flex*, que permita reconhecer um *Thesaurus* ISO 2788 T(2788). É ainda necessário desenvolver uma analisador sintático recorrendo à ferramenta *Yacc*.
Assim, no presente relatório, iremos apresentar as estratégias utilizadas, bem como a linha de pensamento seguida para resolver todos os exercícios propostos. Para além disso, serão explicados alguns pontos extras, que decidimos implementar para enriquecer o projeto desenvolvido.

Estrutura do Relatório

Inicialmente, no capítulo 2 é apresentada uma descrição detalhada do enunciado proposto para a realização deste projeto, bem como os objetivos deste.
De seguida, no capítulo 3 é explicada de que forma é que os documentos de texto Thesaurus ISO 2788 são apresentados.
No capítulo 4 explicamos detalhadamente a implementação da solução de forma a dar resposta ao que fora proposto no enunciado. Para além disto, apresentamos e especificamos algumas funcionalidades extra, que consideramos relevantes no âmbito do trabalho.
No capítulo 5, apresentamos os exemplos de utilização da ferramenta desenvolvida.
Por fim, no capítulo 6 elaboramos uma análise crítica ao trabalho realizado.

Capítulo 2

Descrição do Problema

Neste projeto pretende-se implementar um programa capaz de processar ficheiros do formato *Thesaurus*. Dado que este tipo de ficheiros contém um conjunto de metadados e de conceitos, será necessário armazenar a informação em estruturas apropriadas de modo a ser possível apresentá-las posteriormente.
Desta forma, é expectável que sejam realizadas as seguintes tarefas ao longo do projeto:

- Armazenar a informação numa estrutura de dados em memória
- Escrever uma GIC para o formato T2788
- Escrever um parser/analisador-léxico para reconhecer um Thesaurus
- Recolhida a informação, criar uma página HTML para cada conceito e hiperligações de acordo com as relacões conceptuais.

Capítulo 3

Thesaurus ISO 2788 - Formato do Ficheiro

O documento de texto *Thesaurus* é utilizado na representação de ontologias. Desta forma, contém metadados (indicação das línguas, relações entre conceitos, inversas das relações, etc) e um conjunto de conceitos. Apresentamos assim um exemplo de um documento Thesaurus.

```
%language PT EN
%baselang EN
%inv NT BT

animal
PT animal
NT cat, dog, cow, fish, ant, camel
BT Life being

cat
PT gato
BT animal
SN animal que tem sete vidas e meia
```

Os metadados são inicializadas com o símbolo '%', definindo relações e propriedades matemática.

⇒ **language**: Indica a linguagem utilizada, podendo ser multi-linguístico, contendo assim o ficheiro diversas linguagens.

⇒ **baselanguage**: Indica a língua base do ficheiro, o que nos permite obter a informação sobre a linguagem em que o Thesaurus se encontra escrito.

⇒ **inv**: Permite definir as relações inversas existentes no ficheiro.

O ficheiro também contém relações sendo estas representadas pela sua abreviatura e após esta deverá ser indicada a lista de conceitos, separados por vírgulas. Podem ser definidas novas relações no inicio do ficheiro ou utilizar as previamente existentes.

⇒ **NT**: Representa os termos específicos relativamente ao termo base.

⇒ **BT**: Representa o termo genérico do termo corrente.

⇒ **SN**: Representa a definição do termo.

Assim, os conceitos estão separados por uma linha em branco (LB), sendo a primeira linha o termo base e as restantes todas as relações existentes.
O carater '#' representa o inicio de uma comentário que irá até ao final da linha.

Capítulo 4

Implementação da solução

4.1 Estrutura de Dados

Para a implementação das estruturas optamos por recorrer à biblioteca *Glib*. Tendo em conta as diferentes informações que devemos armazenar definimos várias estruturas. Para guardar os conceitos optamos por criar uma estrutura *Conceito* que contém o nome do conceito, uma *hash table* que irá armazenar todas as relações associadas e o nome do ficheiro HTML que será criado.

```
typedef struct conceito{
    char* nome;
    GHashTable* relacoes;
    FILE* f;
} *Conceito;
```

Criamos a estrutura *Relacoes* que armazena o tipo da relação (NT, BT, PT, EN, ES,...) e os seus termos que serão guardados numa lista ligada.

```
typedef struct relacoes{
    char* tipo;
    GSList*  termos;
} *Relacoes;
```

A estrutura principal será a *Thesaurus*, que irá suportar toda a informação do ficheiro do tipo Thesaurus. Esta contém uma *hash table* para os conceitos, línguas e para as relacões inversas. Para além disso, armazenamos a língua base.

```
typedef struct thesaurus{
    GHashTable* conceitos;
    GHashTable* linguas;
    char* lingua;
    GHashTable* inv;
} *Thesaurus;
```

4.2 Gramática Independente de Contexto

Nesta secção é apresentada e explicada a gramática independente de contexto desenvolvida para o formato T2788. De modo a evitar conflitos e erros de interpretação, utilizamos na definição da *GIC* recursividade à esquerda.
O Thesaurus é constituído por metadados e um conjunto de conceitos como já foi referido anteriormente. O símbolo terminal LB representa a linha em branco, que separa os metadados dos conceitos.

```
Thesaurus : Metadados LB Conceitos
```

O Metadados é uma lista de Meta. Cada Meta é iniciado pelo símbolo terminal '%', seguido do nome da diretiva e de uma lista de Ids, que representam as instruções correspondentes.

```
Metadados : Metadados Meta
          | Meta

Meta: '%' TERMO Ids

Ids: Ids ID
    | ID
```

Conceitos também representa uma lista de *Conceito*, separados por linhas em branco daí a utilização mais uma vez do símbolo terminal LB. Cada conceito possui em primeiro lugar o nome, neste caso representado pelo símbolo terminal TERMO, sendo que de seguida podem surgir um conjunto de relacões. Relacoes representa um lista, constituída pelo nome da relação e os termos associados. Assim, os termos surgem como uma lista, sendo que os elementos se encontram separados vírgula, surgindo o símbolo terminal ','.

```
Conceitos: Conceitos LB Conceito
         | Conceito

Conceito: TERMO Relacoes

Relacoes: Relacoes ID Termos
        | ID Termos

Termos : Termos ',' TERMO
       | TERMO
```

4.3 Analisador Léxico

Para procedermos à análise léxica dos documentos no formato Thesaurus utilizamos a ferramenta *Flex*. Este deverá transformar o texto original de modo a ser interpretado pela GIC anteriormente criada. Assim, os termos de relações ou conceitos deverão ser transformados em *tokens* TERMO e os tipos das relações em *tokens* ID.

Começamos por definir algumas expressões regulares:

```
MINUSCULAS [(á)(à)(ã)(â)(é)(ê)(í)(õ)(ô)(ó)(ú)(ç)a-z0-9]
MAIUSCULAS [(Á)(À)(Ã)(Â)(É)(Ê)(Í)(Õ)(Ô)(Ó)A-Z]
```

Estas foram utilizadas para apanhar os termos e as relações, sendo necessário guardar o seu valor em cada *token*. Para além disso, foi necessário adaptar o mesmo para o caso dos comentários que começam com o símbolo "#". Neste caso, poderão ser ignoradas as linhas. Os simbolos "%"e ","já estão presentes no texto original enquanto que para as linhas em branco optamos pelo símbolo terminal LB.

4.4 Analisador Sintático

Tendo a GIC já sido escrita e o analisador léxico já preparado para a conversão de texto, é possível guardar nos *tokens* os valores relevantes, utilizando o *Yacc* como ferramenta de análise sintática.
Após a análise léxica do ficheiro de texto, é necessário garantir que se encontra de acordo com as regras da gramática previamente definida e ainda processar e tratar os dados fornecidos de modo a dar resposta ao pretendido, sendo estas as principais funcionalidades do nosso analisador sintático. Assim, sempre que é encontrado, por exemplo, um conceito ou uma relaçao, estes são adicionados à nossa estrutura de dados. Desta forma, após armazenar toda a informação relevante, esta será apresentada em ficheiro HTML de forma a visualização ser mais simples.

4.4.1 Carregamento das Estruturas

Foi necessário guardar a informação presente nos *tokens* analisados e introduzir esta nas estruturas criadas para depois serem percorridas de modo a obter a informação pretendida. Serão apresentadas de seguida as ações elaboradas através da ferramenta *Yacc* bem como a forma que é realizado o carregamento das estruturas.

⇒ **Yacc**

Foi utilizada uma lista ligada para o armazenamento da informação dos metadados, de forma a guardar os *tokens* ID que constituem os termos dos metadados. A função *adicionaMeta*, que insere um termo na lista ligada, permite adicionar o metadado do tipo que é determinado pelo valor do *token* TERMO à estrutura.

```
Metadados   : Metadados Meta {}
            | Meta  {}
            ;

Meta        : '%' TERMO Ids {
                adicionaMeta(th,$2,termos);
            }
            ;

Ids         : Ids ID {
                termos = g_slist_append(termos,$2);
            }
            | ID {
                termos = NULL
                termos = g_slist_append(termos,$1);
            }
            ;
```

Para guardar as seguintes informações o racícionio é semelhante. Assim, para a adição do conceito, armazenamos os termos de cada relação numa lista ligada recorrendo à *adicionaTermo* e por sua vez, através da *adicionaRelacao* acrescentamos cada relação à *HashTable* de relações. Tendo todas as relações existentes obtidas, é possível adicionar o conceito ao *Thesaurus* através da *adicionaConceito*.

```
Conceitos   : Conceitos LB Conceito {}
            | Conceito {}
            ;

Conceito    : TERMO Relacoes {
                addConceito(th,$1,relacoes);
            }
            ;

Relacoes    : Relacoes ID Termos {
                addRelacao(relacoes,$2,termos);
            }
            | ID Termos {
                relacoes=g_hash_table_new(g_str_hash,g_str_equal);
                addRelacao(relacoes,$1,termos);
            }
            ;

Termos      : Termos ',' TERMO  {
                addTermo($3,termos);
            }
            | TERMO {
                termos=g_slist_alloc();
                addTermo($1,termos);
            }
            ;
```

⇒ **Métodos Utilizados**

Com referido em cima, de modo a carregar a informação nas estruturas utilizamos quatro métodos, nomeadamente, *adicionaMeta*, *adicionaRelacao*, *adicionaTermo* e *adicionaConceito*.

- *adicionaMeta*: Adiciona os metadados existentes à respetiva estrutura.
- *adicionaRelacao*: Cria a estrutura Relacoes e adiciona a relação na *HashTable* de relações. Na eventulidade de esta já existir, concatenamos os termos com os que já estão armazenados.
- *adicionaTermo*: Adiciona o termo à lista de termos.
- *adicionaConceito*: Cria o Conceito e introduz a informação respetiva que é recebida como argumento da função. Após a inserção deste ainda é criado o ficheiro HTML com o nome deste conceito para mais tarde ser introduzida toda a informação respetiva sobre este, abrindo-o em modo de escrita e guardando ainda o ficheiro na estrutura.

4.5 Extras

Com o intuito de enriquecer o trabalho e dado que implementamos as funcionalidades base, decidimos acrescentar extras à solução implementada. Assim, criamos uma homepage e uma página de estatísticas do nosso ficheiro Thesaurus e ainda processamos as relações inversas.

⇒ **Homepage**

A HomePage irá apresentar toda a listagem de conceitos presentes e ainda o número de tipos de relações que esse contém. A partir desta é possível aceder à página das estatísticas bem como à página referente a cada conceito.

⇒ **Estatísticas**

A página de estatística irá conter toda a informação sobre o nosso ficheiro, desde o número de conceitos ao número de relações inversas. É ainda possível regressar à página inicial.

⇒ **Relações Inversas**

Para o caso das relações inversas tivemos o cuidado de permitir o acesso à página do conceito associado como inverso a este. Desta forma, foi necessário ter em atenção as relações inversas incompletas, isto é, as relações que apenas se encontram definidas apenas num dos conceitos. Nesta situação, decidimos associar a relação ao conceito que ainda não a possui. Foi ainda fundamental, criar os conceitos que ainda não existem mas que estão presentes em relações inversas.

Capítulo 5

Exemplos de Utilização

De seguida iremos expor exemplos da utilização da ferramenta desenvolvida, de modo a demonstrar como a mesma funciona.

⇒ **Página Inicial**

Nesta página HTML é apresentada listagem de todos os conceitos contidos no ficheiro Thesaurus bem como o número do tipo de relações existentes. É ainda possível através da página inicial aceder à página das estatísticas.

Home

Estatísticas

Indice:

Nº Tipo de Relações	Conceitos
4	large intestine
1	skin
1	Régua
1	Serra do Urbião
4	organ
1	doctor
1	Alessandro Piccinini
2	key instrument
1	dentist
1	Barragem do Pocinho
3	animal
1	cuisine
1	kidney
3	tejo river
1	United States of America
1	medical specialist
4	cow
1	plant
2	brain
3	musical instrument

Figura 5.1: Página Principal

⇒ **Estatísticas**

Neta página irá ser apresentado ao utilizador as estatíticas do ficheiro, nomeadamente, o número de conceitos, número de relações, número de línguas, as multi-linguísticas existentes, a língua base e ainda o número de relações inversas. É possível também regressar à página inicial.

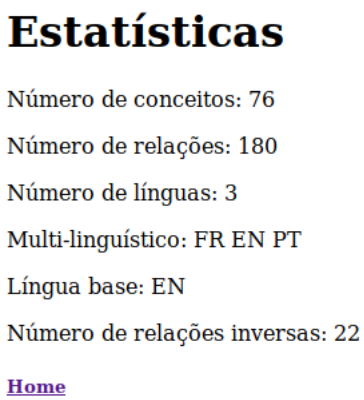

Figura 5.2: Página com as estatísticas

⇒ **Página do Conceito**

A página do conceito começa por apresentar o conceito e, caso contenha, a sua definição. De seguida são apresentadas todas as suas relações permitindo ainda aceder aos conceitos existentes em relações inversas. Por fim, é possível regressar à página inicial.

human body

Relações:

PT

- corpo humano

TERMINOLOGIAESPECIFICA

- arm bones
- arm
- digestive tract
- veins
- human body
- organ

DOMINIO

- human body

HAS

- arm
- organ
- skin
- leg
- head
- muscle

Home

Figura 5.3: Página do conceito human body

Capítulo 6

Conclusão

A elaboração deste trabalho prático permitiu-nos consolidar a matéria lecionada tanto nas aulas teóricas como práticas relativamente à utilização do *Flex* e do *Yacc*. Assim, foi possível percebermos como é o funcionamento destas ferramentas. Para além disso, foi possível desenvolver as nossas capacidades para escrever gramáticas independentes de contexto.
Em última instância consideramos que os objetivos pretendidos foram atingidos e que foi notória a nossa evolução durante o desenvolvimento do projeto na utilização destas ferramentas o que nos permitiu a realização de alguns extras que achamos relevantes para o exercício em questão.