Processamento de Linguagens (3º ano de MIEI)

Trabalho Prático 2: Conversor de GEDCOM para HTML

Relatório de Desenvolvimento

Grupo 64

Hugo Cunha (a84656)

Maria Pires (a86268)

27 de Junho de 2020

Resumo

O segundo trabalho prático no âmbito da unidade curricular *Processamento de Linguagens* consiste na elaboração de um reconhecedor do formato GEDCOM 5.5, que posteriormente o transforme em HTML para que possa ser representada uma árvore genealógica graficamente.
Deste modo, primeiro será efetuada uma introdução ao projeto, bem como uma descrição do seu enunciado. De seguida é explicada toda a implementação da solução, nomeadamente as estruturas de dados utilizadas, o construção da gramática, entre outos. Serão também ilustrados exemplos da solução obtida. Finalmente, apresentaremos uma apreciação crítica ao trabalho desenvolvido.

Conteúdo

Lista de Figuras

Capítulo 1

Introdução

No âmbito da Unidade Curricular de Processamento de Linguagens, foi desenvolvido um conversor do formato GEDCOM 5.5 em HTML. Serão apresentados todo o raciocínio e estratégias utilizadas para a realização do exercício 2.5: *Conversor de GEDCOM*.

1.1 Problema e Objetivo

O formato GEDCOM foi desenvolvido com o objetivo de ser uma ferramenta simples para a representação de dados genealógicos. Um ficheiro deste formato é constituído por texto simples, com registos para cada indivíduo da árvore, tais como o nome e data de nascimento, e dados que possibilitam as ligações entre os vários elementos. Posto isto, é possível construir facilmente uma representação gráfica de qualquer ficheiro recorrendo a HTML.

1.2 Estrutura do Relatório

No capítulo 1 é feito um enquadramento e contextualização do trabalho prático e, em seguida, é feita uma descrição do problema.
No capítulo 2 é feita uma descrição informal do problema e uma especificação dos requisitos necessários para uma correta resolução do problema.
No capítulo 3 são expostas as estruturas de dados utilizadas e é feita uma descrição detalhada de todo o desenvolvimento do projeto até à solução final.
No capítulo 4 são apresentados alguns testes realizados e os resultados obtidos.
No capítulo 5 termina-se o relatório com uma síntese e análise do trabalho realizado.

Capítulo 2

Análise e Especificação

2.1 Descrição informal do problema

Desenvolvimento de um reconhecedor de GEDCOM com recurso a uma gramática tradutora de modo a desenvolver um conjunto de páginas web que represente graficamente a família reconhecida e as informações dos seus constituintes.

2.2 Especificação de Requisitos

Os dados a representar numa árvore genealógica dividem-se em dois grupo, os dados de cada indivíduo e os dados de cada família. Embora os dados apresentados de seguida sejam todos relevantes para a identificação dos elementos da árvore, estes são também todos opcionais.

Dados relevantes de um indivíduo:

- Nome
- Género
- Idade
- Nascimento
- Falecimento
- Nacionalidade
- Relações familiares

Dados relevantes de uma família:

- Marido
- Mulher
- Filhos biológicos do casal
- Filhos adoptivos do casal

Capítulo 3

Concepção/desenho da Resolução

3.1 Estruturas de Dados

A estrutura *Individual* representa cada individuo presente na árvore. Cada individuo é constituído por um identificador, nome e uma lista de casamentos.

```
typedef struct Individual{
  int id;
  GString* name;
  GArray* marrs;
}Individual;
```

A estrutura *Family* representa cada família da árvore, constituída por um marido, mulher e pelos filhos, quer biológicos quer adoptivos, do casal.

```
typedef struct Family{
    int id;
    int h;
    int w;
    GArray* chld;
}Family;
```

Foram também criadas duas *hashtables* para guardar os códigos dos indivíduos e famílias.

```
GHashTable* inds;
GHashTable* fams;
```

A função *buildTree* é responsável por criar uma árvore de descendência para cada família através da *hashtable* que as guarda. Embora esta estrutura não seja guardada em memória é escrita para o ficheiro.

3.2 Implementação

A construção do conversor de GEDCOM foi iniciada com a construção de um analisador léxico, de seguida uma gramática independente de contexto, que implementa a lógica associada à linguagem a reconhecer, e finalmente, através de uma gramática tradutora, são geradas as páginas HTML com a representação da família.

3.2.1 Analisador Léxico

O analisador léxico permite traduzir a sequência de caracteres de entrada num conjunto de símbolos léxicos que constituem a linguagem a reconhecer. Devido à extensão da linguagem foi definido um grande número de símbolos. Associado a cada um desses símbolos existe uma expressão regular para o reconhecer. Apresentam-se de seguida, a título de exemplo, alguns dos símbolos reconhecidos pelo analisador.

```
<tags>NAME{sep}     {sb=strdup("");yy_push_state(name);return NAME;}
<name>[^/_\n]*      {asprintf(&sb," %s%s",sb,yytext);}
<name>\/            asprintf(&sb,"%s%s",sb, " ");
<name>_             asprintf(&sb,"%s ",sb);
<name>[^/_\n]*/\n   {if(yyleng>0)  asprintf(&sb,"%s%s",sb, yytext);
                    yylval.sval=sb;yy_pop_state();return nam;}

<tags>BIRT{sep}?  {yy_pop_state();return BIRT;}
<tags>DEAT{sep}?  {yy_pop_state();return DEAT;}
```

Recorremos ao uso das seguintes condições de contexto para auxiliar o reconhecimento.

```
%x line tags content name tagid  debug  nochan
```

3.2.2 Gramática Independente de Contexto

A abordagem seguida na criação da gramática independente de contexto(GIC) foi top-down.

Cada ficheiro GEDCOM inicia-se com uma tag de inicialização, seguida do *header* que contém informações sobre o ficheiro e o seu autor, de seguida o corpo do ficheiro onde é construida a árvore, e por fim, uma tag que indica o final do ficheiro.

```
Gedcom: INIT Header Body END
      ;
```

O *header* do ficheiro, representado pelo simbolo não terminal Header, é constituído, de modo geral, por linhas que se iniciam por uma tag seguida da informação relativa às mesmas. As tags do cabeçalho são representadas pelo símbolo não terminal HeaderTag.

```
Header: HeaderLine HeaderLines
      ;
HeaderLines: HeaderLine HeaderLines
           |
           ;
HeaderLine: HeaderTag
          ;
HeaderTag: SOUR cont SdList
         | DEST cont SdList
         | DATE cont Cont
         | FileStruct Cont
         | SUBM sub
         | AUTH cont
         | GEDC SdList
         ;
```

O objetivo das tags de continuação é possibilitar a escrita de informação extensa, uma vez que no formato GEDCOM o número de caracteres permitidos por linha é bastante reduzido.

```
Cont:  Cont Continuation
    |
    ;
Continuation: CONT cont
            | CONC cont
            ;
```

A informação contida no cabeçalho segue o formato que se pode observar no seguinte excerto da GIC construída.

```
SdList: Sd SdList
      |
      ;
Sd: NAME nam Cont
  | VERS cont Cont
  | CORP cont Cont CorpList
  | FORM cont
  ;
CorpList: Corp Cont CorpList
        |
        ;
Corp: ADDR cont
    | PHON cont
    ;
FileStruct: FIL cont
          | CHAR cont
          | LANG cont
          ;
```

O corpo do ficheiro é constituído por várias linhas que se iniciam por uma tag que indica se a informação que se segue é relativa a um individuo, uma família ou ao autor do ficheiro.

```
Body: BodyLine BodyList
    ;
BodyList: BodyLine BodyList
        |
        ;
BodyLine: TagId
        ;
TagId: fam Family
     | indi Tags
     | sub  Tags
     ;
```

Cada elemento da árvore por ter associado apenas uma tag ou um conjunto destas.

```
Tags: Tag TagList
    ;
TagList: Tag TagList
       |
       ;
```

As 137 tags presentes no GEDCOM 5.5 podem ser divididas em duas categorias, representadas pelos símbolos não terminais *ContextlessTag* e *Event*.

```
Tag: ContextlessTag Cont
   | Event
   ;
```

As ContextlessTag compreendem várias palavras reservadas, entre as quais, NAME, que representa o nome, TITL que representa o título de um individuo, entre outras.

```
ContextlessTag: NAME nam
              | TITL cont
              | NATI cont
              | NOTE cont
              | NATU cont
              | ALIA cont
              | EMAIL cont
              | OCCU cont
              | Famx
              | FamElem
              | ADDR cont
              | PHON cont
              | DEBUG
              | DEST cont
              | AGE cont
              | SEX cont
              ;
```

O não terminal Famx encontra-se associado a um individuo indicando se este pertence ao casal ou é filho de uma família com determinado identificador reconhecido pelo simbolo terminal fam.

```
Famx: FAMS fam
    | FAMC fam
    ;
```

Uma família é constituída por um ou mais elementos, cada elemento da desempenho um papel nela.

```
Family: FamElem  FamList
      ;
FamList: FamElem  FamList
       |
       ;
FamElem: HUSB indi
      | WIFE indi
```

```
      | CHIL indi
      | MARR ParamList
      | DIV cont
      | ADOP indi
      ;
```

Os eventos podem possuir um conjunto de características associadas, tais como a data e o local e são tratados da seguinte maneira.

```
Event: DEAT EventTail
     | BIRT EventTail
     | BURI EventTail
     | CHR  EventTail
     | BAPL  EventTail
     | BAPM  EventTail
     | BARM  EventTail
     | BASM  EventTail
     | EVEN cont EventTail
     ;
EventTail: Param  ParamList
         ;
ParamList: Param  ParamList
         |
         ;
Param:  PLAC cont
     |  CITY cont
     |  CTRY cont
     |  DATE cont
     |  CAUS cont
     ;
```

3.2.3 Gramática Tradutora

Para a criação da gramática tradutora recorremos ao yacc que nos permitiu preencher as estruturas de dados e criar as páginas web recorrendo a HTML e adicionalmente algum CSS.

3.3 Makefile

De modo a facilitar a compilação do programa criamos a Makefile que apresentamos de seguida. Esta possui a opção de clean, que limpa o projeto removendo os binários e os ficheiros HTML gerados.

```
gedcom.exe : y.tab.o lex.yy.o
gcc -o gedcom.exe y.tab.o lex.yy.o   -ll `pkg-config --cflags --libs glib-2.0`
y.tab.o : y.tab.c
gcc -c -g  y.tab.c  `pkg-config --cflags --libs glib-2.0`
lex.yy.o : lex.yy.c
```

```
gcc -c -g lex.yy.c
y.tab.c y.tab.h y.output : gedcom.y
yacc -d -v --debug gedcom.y
lex.yy.c : gedcom.l y.tab.h
flex gedcom.l
clean: assets gedcom.exe y.tab.o y.tab.c y.tab.h lex.yy.c y.output
rm -rf assets gedcom.exe y.tab.o y.tab.c y.tab.h lex.yy.c y.output
reset: assets
rm -rf assets
```

Capítulo 4

Codificação e Testes

4.1 Testes realizados e Resultados

Apresentamos de seguida um excerto de um dos ficheiros usados para testar o produto final. O ficheiro em questão é bastante extenso uma vez que contém as várias famílias da bíblia.

```
0 HEAD
1 SOUR PAF
2 NAME Personal Ancestral File
2 VERS 5.2.18.0
2 CORP The Church of Jesus Christ of Latter-day Saints
3 ADDR 50 East North Temple Street
4 CONT Salt Lake City, UT 84150
4 CONT USA
1 DEST PAF
1 DATE 9 Aug 2002
2 TIME 16:11:45
1 FILE bible.ged
1 GEDC
2 VERS 5.5
2 FORM LINEAGE-LINKED
1 CHAR UTF-8
1 LANG English
1 SUBM @SUB1@
0 @SUB1@ SUBM
1 NAME Jeffrey Philip Stoker
1 ADDR 1 Broadfields
2 CONT Astley Village
2 CONT Chorley
2 CONT Lancashire
2 CONT England
1 PHON (+44) 01257 263901
1 EMAIL jeffstoker@btconnect.com
0 @I1@ INDI
1 NAME Mizraim //
2 GIVN Mizraim
```

```
1 SEX M
1 FAMS @F1@
1 FAMC @F2@
0 @I2@ INDI
1 NAME Phut //
2 GIVN Phut
1 SEX M
1 FAMC @F2@
0 @I3@ INDI
1 NAME Cush //
2 GIVN Cush
1 SEX M
1 FAMS @F3@
1 FAMC @F2@
0 @I4@ INDI
1 NAME Sabtah //
2 GIVN Sabtah
1 SEX M
1 FAMC @F3@
0 @I5@ INDI
1 NAME Raamah //
2 GIVN Raamah
1 SEX M
1 FAMS @F4@
1 FAMC @F3@
```

A página inicial do programa apresenta-se na figura seguinte. Nesta página encontra-se a informação do cabeçalho do ficheiro, o criador do ficheiro, que clicando nele é possível aceder à sua informação caso esta exista, e ainda, a lista de famílias e indivíduos presentes no ficheiro.

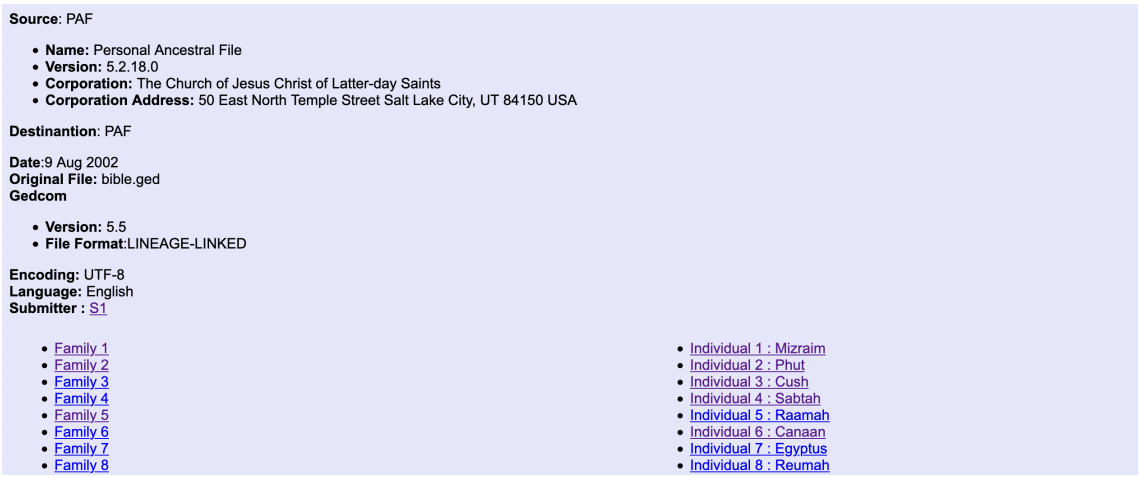

Figura 4.1: Página inicial

RETURN TO INDEX
Individual 1
Name: Jeffrey Philip Stoker
Address: 1 Broadfields
Astley Village Chorley Lancashire England
Phone Number: (+44) 01257 263901
Email: jeffstoker@btconnect.com

Figura 4.2: Informação sobre o criador do ficheiro

RETURN TO INDEX
Family 2
Husband: Ham
Wife: Egyptus
Biological child: Cush
Biological child: Phut
Biological child: Canaan
Biological child: Mizraim

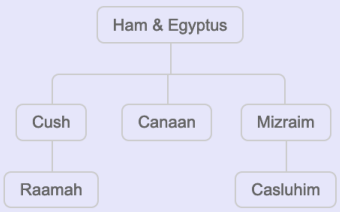

Figura 4.3: Árvore de uma família

Capítulo 5

Conclusão

A solução final permite, de uma forma eficaz reconhecer um especificação em GEDCOM 5.5. Adicionalmente, uma vez que este formato descreve uma hierarquia complexa de relacionamentos entre indivíduos, recorrendo a HTML e uma gramática tradutora, é criada uma página web com toda a informação textual e gráfica. Esta página permite a consulta da informação de elementos que constituem cada família e a sua árvore de descendentes.