Processamento de Linguagens e Compiladores
LCC+MiEFis (3ºano)

Trabalho Prático nº 1 (GAWK)

Ano lectivo 17/18

1 Objectivos e Organização

Este trabalho prático tem como principais **objectivos**:

- aumentar a experiência de uso do ambiente Linux e de algumas ferramentas de apoio à programação;
- aumentar a capacidade de escrever *Expressões Regulares (ER)* para descrição de *padrões de frases*;
- desenvolver, a partir de ERs, sistemática e automaticamente *Processadores de Linguagens Regulares*, que filtrem ou transformem textos;
- utilizar o sistema de produção para *filtragem de texto* GAWK.

Para o efeito, esta folha contém 5 enunciados, dos quais deverá resolver um escolhido em função do número do grupo (NGr) usando a fórmula $exe = (NGr \ \% \ 5) \ + \ 1$.
Neste 1º TP que se pretende que seja resolvido rapidamente (1 semana), os resultados pedidos são simples e curtos. Aprecia-se a imaginação/criatividade dos grupos ao incluir outros processamentos!

Deve entregar a sua solução **até Domingo dia 15 de Outubro**. O ficheiro com o relatório e a solução deve ter o nome "`plc17TP1GrNN`-– *em breve serão dadas indicações precisas sobre a forma de submissão.*

O programa desenvolvido será apresentado aos membros da equipa docente, totalmente pronto e a funcionar (acompanhado do respectivo relatório de desenvolvimento) e será defendido por todos os elementos do grupo, em data a marcar.
O **relatório** a elaborar, deve ser claro e, além do respectivo enunciado, da descrição do problema, das decisões que lideraram o desenho da solução e sua implementação (incluir a especificação GAWK), deverá conter exemplos de utilização (textos fontes diversos e respectivo resultado produzido). Como é de tradição, o relatório será escrito em LaTeX.

2 Enunciados

Para sistematizar o trabalho que se pede em cada uma das propostas seguintes, considere que deve, em qualquer um dos casos, realizar a seguinte lista de tarefas:

1. Especificar os padrões de frases que quer encontrar no texto-fonte, através de ERs.
2. Identificar as acções semânticas a realizar como reacção ao reconhecimento de cada um desses padrões.
3. Identificar as Estruturas de Dados globais que possa eventualmente precisar para armazenar temporariamente a informação que vai extraindo do texto-fonte ou que vai construindo à medida que o processamento avança.
4. Desenvolver um Filtro de Texto para fazer o reconhecimento dos padrões identificados e proceder à transformação pretendida, com recurso ao Sistema de Produção GAWK.

2.1 Processador de Thesaurus 1

Os ficheiros em anexo {comida,veiculos,vestuario}.mdic descrevem numa sintaxe muito simples entradas de um Thesaurus que se pretende criar automaticamente. Cada linha tem 1 ou mais termos. Os termos são separados por ':' representado as relações entre eles (instancia, subclasse, iof, etc.) ou por '—' representando elementos alternativos, como se mostra a seguir:

```
prato principal          : prato de peixe | prato de carne
carne de aves de capoeira: frango| carne de peru| carne de pato
```

que significa que 'prato de peixe' ou ´prato de carne´ são instâncias do termo 'prato principal' e que 'frango', 'carne de peru' ou 'carne de pato' são instâncias de 'carne de aves de capoeira'.
Contudo a interpretação de cada linha de um bloco (referente à descrição de um determinado 'domínio') depende totalmente da linha cabeçalho que inicia cada bloco e que é da forma

```
%the(dom=>NOME-DO-DOMINIO)(syn:REL)
```

'NOME-DO-DOMÍNIO' identifica o tema ou domínio do bloco e em que 'Rel' pode ser: 'inst' (instância); 'nt' (subclasse); 'iof' (instance-of); 'has' (tem).
Nota: Blocos que se iniciem com outras relações que não estas**devem ser ignorados**.
O cabeçalho pode também ser da forma

```
%the(dom=>NOME-DO-DOMINIO)(syn=TERMO:REL)
```

e neste caso o termo mais lato que figura à esquerda no inicio de cada linha (ver exemplos nas linhas anteriores) é sempre o mesmo e igual a TERMO. Por exemplo

```
%the(dom=>culinária)(syn=Molho:iof)
Molho de Hortelã :
Molho Balsâmico :
Molho Bearnês :
```

significa que 'Molho de Hortelã', 'Molho Balsâmico' e 'Molho Bearnês' são instâncias (instance-of) de 'Molho'.
Note que as linhas começadas por '#' são comentários e devem ser ignoradas.
Depois de analisar com cuidado a explicação supra e o conteúdo dos ficheiros em anexo, pretende-se que desenvolva um Processador de Texto com o GAWK para ler esses (dentro do mesmo pacote) e:

a) criar um conjunto de páginas HTML, uma para para cada domínio(cujo nome deve intitular a pagina), que mostre os termos de entrada no thesaurus (sem repetições) associados a cada termo com ele relacionados, agrupados por tipo de relação.

b) criar uma página extra com 'WARNINGS', listando todos os termos *subclasse* que nunca são expandidos.

c) calcular o número máximo de instâncias de um termo.

2.2 Processador de Thesaurus 2

Os ficheiros em anexo {comida,veiculos,vestuario}.mdic descrevem numa sintaxe muito simples entradas de um Thesaurus que se pretende criar automaticamente. Cada linha tem 1 ou mais termos. Os termos são separados por ':' representado as relações entre eles (instancia, subclasse, iof, etc.) ou por '—' representando elementos alternativos, como se mostra a seguir:

```
prato principal          : prato de peixe | prato de carne
carne de aves de capoeira: frango| carne de peru| carne de pato
```

que significa que 'prato de peixe' ou ´prato de carne´ são instâncias do termo 'prato principal' e que 'frango', 'carne de peru' ou 'carne de pato' são instâncias de 'carne de aves de capoeira'.
Contudo a interpretação de cada linha de um bloco (referente à descrição de um determinado 'domínio') depende totalmente da linha cabeçalho que inicia cada bloco e que é da forma

```
%the(dom=>NOME-DO-DOMINIO)(syn:REL)
```

'NOME-DO-DOMÍNIO' identifica o tema ou domínio do bloco e em que 'Rel' pode ser: 'inst' (instância); 'nt' (subclasse); 'iof' (instance-of); 'has' (tem).
Nota: Blocos que se iniciem com outras relações que não estas**devem ser ignorados**.
O cabeçalho pode também ser da forma

```
%the(dom=>NOME-DO-DOMINIO)(syn=TERMO:REL)
```

e neste caso o termo mais lato que figura à esquerda no inicio de cada linha (ver exemplos nas linhas anteriores) é sempre o mesmo e igual a TERMO. Por exemplo

```
%the(dom=>culinária)(syn=Molho:iof)
Molho de Hortelã :
Molho Balsâmico :
Molho Bearnês :
```

significa que 'Molho de Hortelã', 'Molho Balsâmico' e 'Molho Bearnês' são instâncias (instance-of) de 'Molho'.
Note que as linhas começadas por '#' são comentários e devem ser ignoradas.
Depois de analisar com cuidado a explicação supra e o conteúdo dos ficheiros em anexo, pretende-se que desenvolva um Processador de Texto com o GAWK para ler esses (dentro do mesmo pacote) e:

a) gere um ficheiro dot para processar com o interpretador Dotty associado ao GraphViz de modo a desenhar a árvore de classes.

b) gere outro ficheiro dot para desenhar o grafo que associada a um dado termo as suas instâncias.

c) calcular o número total de domínios distintos processados.

2.3 Processador de Thesaurus 3

O ficheiro em anexo `diaADia.mdic` descreve numa sintaxe muito simples entradas de um Thesaurus que se pretende criar automaticamente. Cada linha tem 1 ou mais termos. Os termos são separados por ':' representado as relações entre eles (instancia, subclasse, relacionado-com, tradução, etc.) ou por '—' representando elementos alternativos, como se mostra a seguir:

```
prato principal          : prato de peixe | prato de carne
carne de aves de capoeira: frango| carne de peru| carne de pato
```

que significa que 'prato de peixe' ou ´prato de carne´ são instâncias do termo 'prato principal' e que 'frango', 'carne de peru' ou 'carne de pato' são instâncias de 'carne de aves de capoeira'.
Contudo a interpretação de cada linha de um bloco (referente à descrição de um determinado 'domínio') depende totalmente da linha cabeçalho que inicia cada bloco e que é da forma

```
%the(dom=>NOME-DO-DOMINIO)(syn:REL)
```

'NOME-DO-DOMÍNIO' identifica o tema ou domínio do bloco (quando nada é dito entre parêntesis considera-se que o domínio é o 'universo') e em que 'Rel' pode ser: 'inst' (instância); 'nt' (subclasse); 'bt' (superclasse); 'iof' (instance-of); 'has' (tem); 'rt' (related-term); 'EN' (tradução para inglês); 'DE' (tradução para alemão); 'gn-de' (género-de).
Nota: um cabeçalho pode especificar que em relação ao termo base ('syn') mais do que uma relação será definida e nesse caso essas várias relações separam-se também por ':'. Por exemplo se o cabeçalho (para indicar que a seguir ao termo se define o seu género, e as traduções para inglês e alemão) for

```
%the(dom=>NOME-DO-DOMINIO)(syn:gn-de:EN:DE)
```

uma possível linha do bloco é

```
papá : m : daddy | dad : Vati | Papa | Papi
```

O cabeçalho também pode ser da forma

```
%the(dom=>NOME-DO-DOMINIO)(syn=TERMO:REL)
```

e neste caso o termo mais lato que figura à esquerda no inicio de cada linha (ver exemplos nas linhas anteriores) é sempre o mesmo e igual a TERMO. Por exemplo

```
%the(dom=>culinária)(syn=Molho:iof)
Molho de Hortelã :
Molho Balsâmico :
Molho Bearnês :
```

significa que 'Molho de Hortelã', 'Molho Balsâmico' e 'Molho Bearnês' são instâncias (instance-of) de 'Molho'.
Note que as linhas começadas por '#' são comentários e devem ser ignoradas.

Depois de analisar com cuidado a explicação supra e o conteúdo dos ficheiros em anexo, pretende-se que desenvolva um Processador de Texto com o GAWK para ler esses (dentro do mesmo pacote) e:

a) criar um conjunto de páginas HTML, uma para para cada domínio(cujo nome deve intitular a pagina), que mostre os termos de entrada no thesaurus (sem repetições) associados a cada termo com ele relacionados, agrupados por tipo de relação, excluindo as relações de tradução.

b) criar duas outras páginas com as traduções Português-Inglês e Português-Alemão, com um par (TERMO - TRADUÇÃO) por linha.

Ficheiros de internacionalização PO

Os ficheiros PO nasceram da necessidade de lidar com tradução de mensagens presentes nos programas e recursos ligados. Existem naturalmente várias bibliotecas, comandos e aplicações ligadas à sua criação, suporte, etc, (como gettext, poedit), que neste caso não iremos discutir nem utilizar.

Descrição do formato PO

O formato PO, simplificadamente contem metadados iniciais (o 1.o parágrafo), e blocos de tradução.
Exemplo dos metadados iniciais:

```
# JustinoAveiro <paulojustino.nec@gmail.com>, 2011.
# Sandro Amaral <sandro123iv@gmail.com>, 2011.
msgid ""
msgstr ""
"Project-Id-Version: TortoiseSVN\n"
"POT-Creation-Date: 1900-01-01 00:00+0000\n"
"PO-Revision-Date: 2012-03-21 18:55+0000\n"
"Last-Translator: JustinoAveiro <paulojustino.nec@gmail.com>\n"
"Language-Team: Portuguese <translators@tortoisesvn.tigris.org>\n"
"Content-Type: text/plain; charset=UTF-8\n"
"Language: pt\n"
"Plural-Forms: nplurals=2; plural=(n != 1)\n"
```

Cada bloco de tradução contêm `msgid "frase original"` (quase sempre o Inglês), e `msgstr "tradução"` para a língua em causa.

```
#. Resource IDs: (563)
#, c-format
msgid "%ld paths"
msgstr "%ld caminhos"
```

```
#. Resource IDs: (93)
#, c-format
msgid "%s (offline)"
msgstr "%s (fora de linha)"

#, c-format
msgid "%s : Remote file"
msgstr "%s : Ficheiro remoto"

#. Resource IDs: (138)
msgid "&Browse repository"
msgstr "&Navegar repositório"

#. Resource IDs: (1516)
msgid "Default URL:"
msgstr "URL por omissão:"

#. Resource IDs: (1007)
msgid "Default application menu. Appears when no documents are open."
msgstr "Menu da aplicação por omissão. Surge quando não há documentos abertos."
```

Complementarmente:

- Os blocos são separados por uma ou mais linha em branco.
- As linhas começadas por "#"são comentários (usados de forma muito heterogénea).
- Por vezes as mensagens (original ou tradução) podem estar quebradas por várias linhas (`""` seguido de várias "strings com conteúdo").
- Um bloco com tradução contendo apenas `msgsrt ""` corresponde a uma tradução ainda não realizada.

2.4 Ficheiros de internacionalização 1

Considere o formato PO atrás descrito.
Pode procurar na sua máquina, na rede, ficheiros PO das várias línguas. Em `natura.di.uminho.pt/~jj/plc-17/PO` colocamos alguns exemplos.

Tarefas a realizar

a) Calcular estatísticas: Crie um resumo contendo os metadados que achar relevantes, e que

   1. conte as mensagens ainda não traduzidas,
   2. desenhe uma progress-Bar do estado de tradução.

b) Dado um conjunto de POs:

   - crie um compendium com a junção dos POs da mesma língua,
   - marque como "fuzzy"as mensagens com mais que uma tradução (i.e. blocos da mesma língua com idêntico msgid, e diferentes msgstr – contando quantas vezes ocorreu cada tradução).
   - ... escolha um formato textual para esse compendium.

c) Reformate um ou mais POs em HTML.

2.5 Ficheiros de internacionalização 2

Considere o formato PO atrás descrito.

Pode procurar na sua máquina, na rede, ficheiros PO das várias línguas. Em `natura.di.uminho.pt/~jj/plc-17/PO` colocamos alguns exemplos.

Tarefas a realizar

a) Calcular estatísticas: Crie um resumo contendo os metadados que achar relevantes, e que

   1. conte os blocos de tradução existentes,
   2. dada uma lista de POs, crie uma lista com os tradutores encontrados (ver o bloco inicial dos metadados).

b) Calcular dicionários por transitividade/triangulação (Pt→Fr = join(PO(En→Pt), PO(En→Fr))).

c) Reformate um ou mais POs em LaTeX.