Processamento de Linguagens
LEI (3ºano) + LCC (2ºano)

Trabalho Prático nº 1
(Lex)

Ano lectivo 09/10

1 Objectivos e Organização

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

- aumentar a experiência de uso do ambiente Linux, da linguagem imperativa C (para codificação das estruturas de dados e respectivos algoritmos de manipulação), 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 *geradores de filtros/processadores de texto*, como o Flex

Para o efeito, esta folha contém 5 enunciados, dos quais deverá resolver pelo menos um. 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 (3 alunos), 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 e a implementação, 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 lhe 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 Processador de Texto para fazer o reconhecimento dos padrões identificados e proceder à transformação pretendida, com recurso ao Gerador Flex.

2.1 BibTeXPro — Um processador de BibTeX

BibTeX é uma ferramenta de formatação de citações bibliográficas em documentos LaTeX, criada com o objectivo de facilitar a separação da base de dados da bibliografia consultada da sua apresentação no fim do documento LaTeX em edição. BibTeX foi criada por Oren Patashnik e Leslie Lamport em 1985, tendo cada entrada nessa base de dados textual o aspecto que se ilustra a seguir

Listing 1: Exemplo de entrada em BibTex

```
@InProceedings{CPBFH07e,
  author =    {Daniela da Cruz and Maria João Varanda Pereira
               and Mário Béron and Rúben Fonseca and
               Pedro Rangel Henriques},
  title =     {Comparing Generators for Language-based Tools},
  booktitle = {Proceedings of the 1.st Conference on Compiler
               Related Technologies and Applications, CoRTA'07
               --- Universidade da Beira Interior, Portugal},
  year =      {2007},
  editor =    {},
  month =     {Jul},
}
```

De modo a familiarizar-se com o formato do BibTeX poderá consultar o ficheiro `lp.bib` disponível em `http://www.di.uminho.pt/~prh/lp.bib` e ainda a página oficial do formato referido (`http://www.bibtex.org/`), devendo para já saber que a primeira palavra (logo a seguir ao caracter "@") designa a categoria da referência (havendo em BibTeX pelo menos 14 diferentes).

As tarefas que deverá executar neste trabalho prático são:

a) Analise o documento BibTeX referido acima e faça a contagem das categorias (`phDThesis`, `Misc`, `InProceeding`, etc.), que ocorrem no documento. No final, deverá produzir um documento em formato HTML com o nome das categoriasencontradas e respectivas contagens.

b) Complete o processador de modo a filtrar, para cada entrada de cada categoria, a respectiva chave (a 1ªpalavra a seguir à chaveta), autores e título. O resultado final deverá ser incluido no documento HTML gerado na alínea anterior.

c) Melhore o seu processador acrescentando-lhe uma funcionalidade de pesquisa, por exemplo por nome de um autor (se preferir, em vez deste pode escolher outro campo de procura a seu gosto).
Para o efeito o seu processador deve gerar um índice de autores, que mapeie cada autor nos respectivos registos, de modo a que posteriormente uma ferramenta de procura do Linux possa fazer a pesquisa.

d) Construa um Grafo que mostre, para um dado autor (escolhido pelo utilizador) todos os autores que publicam normalmente com o autor em causa.
Recorrendo à linguagem Dot do GraphViz[1], gere um ficheiro com esse grafo de modo a que possa, posteriormente, usar uma das ferramentas que processam Dot [2] para desenhar o dito grafo de associações de autores.

2.2 Pré-processador para LaTeX ou HTML

Desenvolver um documento em LaTeX ou mesmo em HTML é uma actividade inteligente e intelectualmente interessante enquanto estruturante das ideias e sistematizante dos processos. Porém o acto de editar o respectivo documento é por vezes fastidioso devido ao peso das marcas (as *tags*) que tem de ser inseridas para anotar o texto com indicações de forma, conteúdo ou formato.

Por isso apareceram editores sensíveis ao contexto que sabendo que se está a escrever um documento LaTeX ou HTML nos facilitam a vida inserindo as ditas marcas, ou anotações. Uma alternativa mais simples mas também muito usada

[1] Disponível em `http://www.graphviz.org`
[2] Disponíveis em `http://www.graphviz.org/Resources.php`

é permitir o uso de anotações mais leves e simples (até de preferência independentes do tipo de documento final) e de pois recorrer ao pré-processamento para substituir essa notação ligeira, abreviada, pelas marcas finais correctas.

Este é o caso do conhecido PPP[3], desenvolvido há alguns anos por José Carlos Ramalho, ou mesmo do mais actual e não menos conhecido sistema Wiki para construção interactiva e via web de páginas HTML.

O que se lhe pede neste trabalho é que, depois de investigar os tais pré-processadores PPP e Wiki, especifique uma sua linguagem de anotação para abreviar a escrita de **formatação** (*negrito*, *itálico*, *sublinhado*, vários níveis de *títulos*) e **listas de tópicos (items)** *não-numerados*, *numerados* ou tipo *entradas de um dicionário*. Deve, depois, criar, com a ferramenta Flex, um processador que transforme a sua notação em LaTeX ou HTML[4].

2.3 LaTeX importer

Desenvolva um pré-processador que aceite texto LaTeX com mais uma marcação especial `umImport`,

```
\begin{umImport}{gnuplot}
...texto gnuplot....
\end{umImport}
```

e que receba, como parâmetro, um segundo elemento (no exemplo acima gnuplot) indicativo do processador a utilizar.

Como resultado o pré-processador deverá:

- copiar para um ficheiro auxiliar o texto em causa (marcado pelo novo elemento),
- executar um comando externo que construa uma imagem PDF aplicando a esse ficheiro o processador indicado,
- substituir o texto e a nova marca que o envolve pelo comando `includegraphics` para importar a imagem PDF produzida.

Sugestões: comece por considerar os formatos gnuplot e dot, mas guarde numa tabela os comandos externos a executar para produzir a imagem PDF, de modo a facilitar a definição de novos formatos de importação.

2.4 Processamento de Códigos Postais

Neste trabalho pretende-se que estruture, numa página HTML, por regiões os Códigos Postais de Portugal, a partir de uma base de dados textual que lhe será fornecida, atendendo a que dentro de cada região, os códigos devem vir agrupados por áreas ordenados por ordem crescente de frequesia. A BD original contém um código por linha no seguinte formato:

```
DDDD-DDD Ident, IDENT
```

por exemplo

```
4100-123 Boavista, PORTO
4710-057 Gualtar, BRAGA
```

Em relação ao radical, sabe-se que o 1º dígito (1 a 9) identifica uma das nove regiões postais do País (1=Lisboa, 9=Ilhas) e que os restantes 3 dígitos identificam a área postal cujo nome (em maiúsculas) é a ultima palavra da linha; a primeira palavra identifica a freguesia dentro da área. Os 3 dígitos da extensão designam o bairro ou rua dentro dessa freguesia.

Melhore o seu processador, acrescentado um facilidade adicional que gere um índice de procura que permita posteriormente o uso de outras ferramentas para fazer pesquisa de códigos postais por freguesia.

Recorrendo à linguagem Dot do GraphViz[5], gere um ficheiro com a árvore de regiões e áreas postais de modo a que possa, posteriormente, usar uma das ferramentas que processam Dot [6] para desenhar a dita árvore com a estrutura hierárquica da nossa (Portugal) organização postal—considere que a raíz da árvore tem o código 0 e corresponde a Portugal.

[3]Consultar detalhes no manual da linguagem em `http://www.di.uminho.pt/~jcr/AULAS/plc2008/tp1/ppp.html`.

[4]O mais interessante mesmo é que fosse possível escolher a saída final no inicio do próprio texto a pré-processar.

[5]Disponível em `http://www.graphviz.org`

[6]Disponíveis em `http://www.graphviz.org/Resources.php`

2.5 Processamento de Trilhos GPS

O formato GPX armazena *trilhos de GPS*. Milhares desses trilhos estão disponíveis na internet, podendo ser descarregados, por exemplo, a partir do site `www.openstreetmap.org`, escolhendo a opção `'GPS traces'`.
Quem tiver um telemóvel ou PDA com GPS pode também registar trilhos, e depois descarregá-los no formato GPX (dependendo do software que usar para o registo).

Desenvolva em Flex um filtro que transforme um documento em formato GPX no formato KML. O documento resultante, no formato KML, deverá ser visualizado no GoogleEarth, ou noutro visualizador qualquer.

2.6 Processamento do Livro de Contactos do GMail

Suponha que como utilizador do cliente de emails GMail, pretende exportar **todos** os seus contactos para o formato CSV[7], a fim de os importar para o Outlook. Depois de exportar os seus contactos, analise cuidadosamente o ficheiro e observe que na primeira linha encontra a definição dos campos de cada contacto existente no seu livro de endereços. Por exemplo:

```
First Name,Middle Name,Last Name,Title,Suffix,Initials,Web Page,Gender,Birthday,...,Priority,Private,Categories
```

As tarefas que deverá executar neste trabalho prático são:

a) Desenvolver um funcionalidade para *agrupar contactos de acordo com as suas categorias* (Mais contactados, Normal, etc); No final, deverá produzir um documento em formato HTML com as categorias e respectivos contactos.

b) Desenvolver um funcionalidade para *agrupar contactos de acordo com o tipo de email* (pode escolher 3 dos seus mais frequentes — p.e. GMail, Hotmail, Yahoo, etc). O resultado final deverá ser incluído no documento HTML gerado na alínea anterior.

c) Melhorar o processador acrescentando-lhe uma funcionalidade de *pesquisa por nome* (primeiro ou último). Para o efeito o seu processador deve gerar um índice de contactos, que mapeie cada nome nos respectivos emails, de modo a que posteriormente uma ferramenta de procura do Linux possa fazer a pesquisa.

[7] `http://pt.wikipedia.org/wiki/Comma-separated_values`