Processamento de Linguagens e Compiladores
LCC (3ºano) + MiEFis (4ºano)

Trabalho Prático nº 1 (FLex)

Ano lectivo 19/20

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 com base no conceito de regras de produção ¡Condição-Ação¿;
- utilizar o FLex para gerar *filtros de texto em C*.

Para o efeito, esta folha contém 6 enunciados, dos quais deverá resolver um escolhido em função do maior número de aluno (Nal) entre os membros de cada grupo usando a fórmula $exe = (Nal \ \% \ 6) \ + \ 1$.
Neste 1º TP que se pretende que seja resolvido rapidamente (2 semanas), 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 13 de Outubro**. O ficheiro com o relatório e a solução deve ter o nome "`plc19TP1GrNN`– *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 FLex), 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 Gerador FLex.

2.1 EnameXPro — Um processador de Enamex

O Reconhecimento de Entidades Nomeadas (em inglês *NER, Named Entities Recognition*) é uma atividade muito complexa que constitui ainda hoje um enorme desafio para os investigadores em PLN (Processamento de Lingua Natural). Existem inclusive concursos internacionais para verificar quem é capaz de desenvolver reconhecedores com maior precisão e acuidade (que marquem corretamente todos os casos de entidades cujo nome surge num dado texto de entrada, em análise).
O problema todo está em criar reconhecedores capazes de identificar os nomes num dado texto e indicar se se trata de uma *pessoa*, *local* (cidade ou país), ou *organização*.
Neste trabalho o que se lhe pede é para pós-processar um ficheiro já criado por um processador NER que anotou o texto de entrada com etiquetas (tags XML) da norma ENAMEX—veja o anexo `'exemplo-Enamex.xml'`—e responder às alíneas seguintes:

1. Criar uma página HTML com todos os nomes de pessoas encontrados (por ordem alfabética e sem repetições).
2. Tal como na alinea anterior, criar uma página com todos os locais identificados indicando se é uma cidade ou país.
3. Ajudar a resolver as indefinições, isto é, os casos em que o processador NER original identificou um nome (palavra começada por maiúscula) mas não conseguiu saber que tipo de entidade estava a ser nomeada (a maioria das vezes porque não era mesmo o nome de uma entidade).

2.2 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

```
@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},
  note  =     {}
}
```

De modo a familiarizar-se com o formato do BibTeX poderá consultar o ficheiro `exemplo-utf8.bib` que se anexa 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 categorias encontradas 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 incluído no documento HTML gerado na alínea anterior.

c) Crie 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 (definido à partida) 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.

2.3 Pré-processador para LaTeX

Desenvolver um documento em LaTeX é uma actividade inteligente e intelectualmente interessante enquanto estruturante das ideias e sistematizadora dos processos. Porém o acto de editar o respectivo documento é por vezes fastidioso devido ao peso das marcas (os comandos do LaTeX) 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 nos facilitam a vida inserindo as ditas marcas, ou anotações. Uma alternativa mais simples e muito vulgar é permitir o uso de anotações mais leves e simples (até de preferência independentes do tipo de documento final) e depois recorrer ao pré-processamento para substituir essa notação ligeira, abreviada, pelas marcas finais correctas.

Este é o caso do conhecido PPP[3].

O que se lhe pede neste trabalho é que, depois de investigar os tais pré-processadores PPP, especifique uma sua linguagem de anotação para abreviar a escrita de **formatação** (exemplos: *negrito*, *itálico*, *sublinhado*, *títulos* de vários níveis como capítulo, secção, subsecção...) e **listas de tópicos (items)** *não-numerados*, *numerados* ou tipo *entradas de um dicionário*.

Deve a seguir criar, com a ferramenta Flex, um processador que transforme a sua notação em LaTeX.

2.4 Pré-processador para HTML

Desenvolver um documento em HTML é uma actividade inteligente e intelectualmente interessante enquanto estruturante das ideias e sistematizadora 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 HTML nos facilitam a vida inserindo as ditas marcas, ou anotações. Uma alternativa mais simples mas também muito usada 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 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 a linguagem de um Wiki, especifique uma sua linguagem de anotação para abreviar a escrita de **formatação** (exemplos: *negrito*, *itálico*, *sublinhado*, *títulos* de vários níveis como capítulo, secção, subsecção...) e **listas de tópicos (items)** *não-numerados*, *numerados* ou tipo *entradas de um dicionário*.

Deve a seguir criar, com a ferramenta Flex, um processador que transforme a sua notação em HTML.

2.5 XML to dot

Pretende-se neste trabalho criar e desenhar um grafo de dependências entre os elementos de um documento anotado em XML.

Para isso e dado um ficheiro XML, cada vez que encontrar um elemento X com um subelemento Y, exemplo:

```
<X> ...
     <Y> ...<Z> ... </Z>... </Y>
...  <W> .... </W> .... <Y> ... </Y>
</X>
```

[1] Disponível em `http://www.graphviz.org`
[2] Disponíveis em `http://www.graphviz.org/Resources.php` ou a ferramenta Web `http://www.webgraphviz.com/`
[3] Consultar detalhes no manual da linguagem em `http://www.di.uminho.pt/~jcr/AULAS/plc2008/tp1/ppp.html`

deve gerar um ramo de um grafo orientado que ligue X a Y de modo a que no final se possa obter uma árvore documental com a estrutura de elementos. Cuidado pois deve evitar ramos repetidos.
Para o exemplo acima teríamos os 3 ramos abaixo

```
strict  digraph g {
x -> y ;
y -> z ;
x -> w ;
}
```

Recorrendo à linguagem Dot do GraphViz[4], gere um ficheiro com esse grafo de modo a que possa, posteriormente, usar uma das ferramentas que processam Dot [5] para desenhar o dito grafo.

2.6 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.

[4]Disponível em `http://www.graphviz.org`
[5]Disponíveis em `http://www.graphviz.org/Resources.php` ou a ferramenta Web `http://www.webgraphviz.com/`