

Processamento de Linguagens

MiEI (3ºano)

Trabalho Prático nº 2 (FLex)

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 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 5 enunciados, dos quais deverá resolver um escolhido em função do mais baixo número aluno do grupo ($NAlu$) usando a fórmula $exe = (NAlu \% 5) + 1$.

Neste 2º 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 XXX de Abril**. O ficheiro com o relatório e a solução deve ter o nome 'p117TP2GrNN' — *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 **L^AT_EX**.

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 BibTeXPro — Um processador de BibTeX

BibTeX é uma ferramenta de formatação de citações bibliográficas em documentos L^AT_EX, criada com o objectivo de facilitar a separação da base de dados da bibliografia consultada da sua apresentação no fim do documento L^AT_EX 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},  
}
```

De modo a familiarizar-se com o formato do BibTeX poderá consultar o ficheiro `lp.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 carácter “@”) designa a categoria da referência (havendo em BibTeX pelo menos 14 diferentes).

As tarefas que deverá executar neste trabalho prático são:

- 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.
- Complete o processador de modo a filtrar, para cada entrada de cada categoria, a respectiva chave (a 1^a palavra a seguir à chave), autores e título. O resultado final deverá ser incluído no documento HTML gerado na alínea anterior.
- 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.
- 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 GraphViz¹, 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

Desenvolver um documento em L^AT_EX 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 (os comandos do L^AT_EX) 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 L^AT_EX 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 Markdown ³.

¹Disponível em <http://www.graphviz.org>

²Disponíveis em <http://www.graphviz.org/Resources.php>

³Consultar detalhes no manual da linguagem em <https://github.com/adam-p/markdown-here/wiki/Markdown-Cheatsheet>.

O que se lhe pede neste trabalho é que, depois de investigar Markdown, 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 a seguir criar, com a ferramenta Flex, um processador que transforme a sua notação em \LaTeX .

2.3 Pré-processador para HTML

Desenvolver um documento 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 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** (*negrito, itálico, sublinhado*, vários níveis de *títulos*) 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.

2.4 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> ... </Y>
  ...
</X>
```

deve gerar uma linha Dot-graphviz (vem `man dot`) de modo a que no final se possa obter uma árvore documental com a estrutura de elementos. No final gere a imagem respectiva com o comando `dot`.

Exemplo de ficheiro dot a gerar:

```
strict digraph g {
x -> y ;
a -> b ;
a -> x ;
x -> c ;
}
```

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.