

Métodos de Programação III

LESI + LMCC (3º ano)

Trabalho Prático nº 1
(Lex: um gerador de autómatos)

Ano lectivo 2003/2004

O trabalho deve ser realizado em grupo, e deve ser entregue acompanhado dum relatório, na forma que será indicada pelos docentes, até 31 de Outubro de 2003 ao fim da tarde.

Objectivos e Organização

Este trabalho prático tem como **objectivos** levar os alunos, através da resolução de problemas reais, a:

- praticar a programação baseada em Sistemas de Produção, ou seja Sistemas de Regras Condição-Acção;
- utilizar a expressividade total das Expressões Regulares para definir padrões de textos (sequências de caracteres);
- utilizar o gerador de reconhedores/tradutores baseados em autómatos reactivos Flex, aplicando-o a problemas de processamento de textos (filtros, simuladores, interpretadores de comandos, etc.)

Para o efeito, esta folha contém dois enunciados, dos quais deverá resolver pelo menos um. O **relatório** a elaborar em \LaTeX deve ser claro, completo e muito bem organizado, servindo com objectividade para a avaliação do seu trabalho, mas sobretudo constituindo um bom elemento de estudo, quer para si, quer para colegas seus; além do respectivo enunciado, análise do problema, decisões e detalhes de implementação, deverá conter exemplos de utilização, isto é, listagens de resultados obtidos que demonstrem bem a forma como concretizou a tarefa que lhe foi colocada.

Enunciados

1 Corrector Ortográfico

Neste trabalho pretende-se desenvolver um corrector ortográfico, para detectar erros e gralhas nos documentos \LaTeX .

O corrector deve trabalhar com várias línguas; para cada língua tem que existir o respectivo dicionário, ou será retornado um erro ao utilizador.

1.1 Funcionalidades

1. O corrector recebe um texto \LaTeX e devolve todas as palavras encontradas que não estejam no dicionário.
2. O corrector tem que ser indiferente aos comandos \LaTeX , pois não faz sentido verificar a sua correcção. Mesmo comandos definidos pelo utilizador devem ser tratados, pois são sempre começados por `'\'`.
3. Deve ser indiferente às maiúsculas e minúsculas (use adequadamente as funções C `toupper()` ou `tolower()`).
4. Deve respeitar a acentuação das palavras.
5. Ao ser invocado, pode ser indicado o dicionário usando uma opção `-d` seguida do nome do dicionário (sem extensão).
Exemplo: `curretor -d spanish trabajo01.lhs`
6. Se não for indicado nenhum dicionário, o corrector deve dizer qual é o que vai utilizar.
7. Deve ignorar as linhas de comentário (em \LaTeX , correspondem às linhas começadas por `%`).
8. Deve ignorar todo o conteúdo do ambiente `code`, utilizado para escrever código Haskell.

1.2 Dicionários

Para cada língua deverá existir um dicionário e, eventualmente um dicionário pessoal, com as palavras adicionadas pelo utilizador (para não alterar o dicionário base).

Um dicionário é uma lista de palavras, ordenado alfabeticamente, para permitir procuras mais eficientes. Os dicionários pessoais, também devem ter esta forma.

Para cada língua deverá ser criado um dicionário, a partir dos dicionários disponíveis para o `ispell` (consulte o <http://fmg-www.cs.ucla.edu/geoff/ispell-dictionaries.html>).

A criação dos dicionários (só com uma palavra por linha, ordenados, sem repetições) deve ser feita exclusivamente com comandos Unix (`sort`, `awk`, `grep`, `sed`, etc) e o processo de conversão deve ficar registado no relatório. Os dicionários tratados podem ser usados noutras ferramentas, como o `WinEdt`.

1.3 Objectivos mínimos

- Preparar 2 dicionários
- Implementar todas as funcionalidades da 1 à 6.

1.4 Características recomendadas

- Preparar 5 dicionários
- Implementar todas as funcionalidades.
- Além do ambiente `code`, também não deve processar as palavras dentro do ambiente `verbatim`.
- Deve ter a possibilidade de acrescentar as palavras devolvidas a um dicionário pessoal, com o mesmo nome do dicionário de base, mas com uma extensão diferente. Ou seja, depois de corrigido, as eventuais palavras que o corrector devolva, são incorporadas no dicionário pessoal, e não voltam a ser devolvidas. Isto pressupõe que a verificação se faz consultando o dicionário da língua e simultaneamente o dicionário pessoal respectivo. (Pode implementar esta funcionalidade com uma opção a passar na linha de comandos.)

1.5 Características de sonho

- Acrescente a possibilidade do corrector ser interactivo, perguntando para cada palavra não presente no dicionário, se:
 - a quer substituir, escrevendo uma nova
 - se a quer acrescentar ao dicionário pessoal
 - se quer que o corrector ignore esta ocorrência
 - se quer que o corrector a ignore durante todo o texto

Nesta opção interactiva, deixa de ser retornada a lista de palavras no final do processamento.

- A mesma possibilidade anterior, mas calculando as palavras parecidas que constam no dicionário (investigue uma forma de o fazer, mesmo que não seja muito eficiente). Depois acrescente a possibilidade de escolher uma das palavras propostas para substituir a errada.
- Permitir que partes do textos sejam corrigidas com um dicionário específico. Para isso, o utilizador deve criar no início do texto `LATEX` um novo ambiente para cada uma das línguas que quer utilizar.

Exemplo:

```
\newenvironment{dic:english}{}{}  
\newenvironment{dic:spanish}{}{}
```

Depois, usando o ambiente definido, pode indicar que dicionário deve ser usado em cada parte do texto.

Exemplo:

```
...aqui está texto no Português de Camões, mas vejam que
\begin{dic:english} the program must understand
spanish, and process words like
\end{dic:english}\begin{dic:spanish}dinero, comida,
trabajo\end{dic:spanish} mas depois voltamos ao Português...
```

No exemplo anterior, o programa podia ser invocado com o dicionário de português, mas dentro dos ambientes `dic:english` e `dic:spanish`, usaria os dicionários de Inglês e de Castelhana, respectivamente. Depois, voltaria a usar o dicionário de Português.

- Suportar os dicionários do `ispell`, sem nenhum tratamento, a partir das suas localizações no sistema (se o `ispell` estiver instalado).
- Dar asas à imaginação!

2 O meu telemóvel

Pretende-se que utilize a ferramenta `Lex` para implementar uma máquina de estados que modele a interacção de um utilizador com o seu telemóvel. O telemóvel reage aos seguintes comandos:

LIGAR - marca o início de uma interacção; qualquer operação realizada antes desta deverá ter como resposta `TELEFONE DESLIGADO`.

CARREGAR `<num-inteiro>` - aumenta o saldo em `num-inteiro` euros; o saldo entre sessões deverá ser mantido, ou seja, se no fim de uma sessão o saldo fôr de 12 euros, a próxima sessão deverá arrancar com um saldo inicial de 12 euros; uma sessão corresponde a uma execução do programa.

T=numero - liga para o número (este deverá ser composto por 9 dígitos excepto se iniciar por "00"); os diferentes tipos de chamada deverão ser tratados da seguinte maneira:

- para números iniciados por "601" ou "641" a chamada é "barrada" e a mensagem "NÚMERO NÃO AUTORIZADO" é devolvida ao utilizador.
- para chamadas internacionais (iniciadas por "00") o utilizador tem que ter um saldo igual ou superior a 1,25 euros, caso contrário deverá ser avisado que o saldo é insuficiente e a máquina volta ao estado anterior; se a chamada fôr realizada tem um custo de 1,25 euros.
- para chamadas nacionais (iniciadas por "2") o saldo mínimo e custo da chamada é de 20 cêntimos.
- para chamadas verdes (iniciadas por "800") o custo é 0.
- para chamadas azuis (iniciadas por "808") o custo é de 10 cêntimos.

T-num - O telemóvel deverá guardar os últimos 10 números utilizados numa estrutura LIFO ("last in first out", ex: uma stack). Se $num = 0$ a chamada deverá ser feita para o último número usado, se $num = 1$ para o penúltimo, ..., até $num = 9$; Qualquer outro valor para num deverá originar a mensagem "NÚMERO FORA DO REGISTO"; o custo da chamada será em função do tipo de chamada; Esta lista de números não é preservada entre sessões.

T=nome - faz a chamada para o número associado a **nome** na Lista Telefónica interna; o custo da chamada será em função do tipo de chamada.

LT ins nome numero - Insere o par (*nome, numero*) na Lista Telefónica; o estado desta lista deverá ser preservado entre sessões.

LT rem nome - Remove o par correspondente a *nome* da Lista Telefónica.

LT in - Entra em modo de Lista; neste estado podem-se fazer as seguintes operações sobre a lista telefónica:

ins nome numero - Insere o par (*nome, numero*) na Lista Telefónica.

rem nome - Remove o par correspondente a *nome* da Lista Telefónica.

procura nome - Se *nome* pertencer à lista telefónica o respectivo número é enviado para a saída.

t - faz a chamada para o número associado ao último par que foi consultado (resultado de uma procura ou listagem); o custo da chamada será em função do tipo de chamada.

list letra - Lista todos os pares (nome e número) da lista telefónica cujo nome começa por *letra*; Se *letra = ** todos os pares deverão ser listados.

exit Volta ao modo normal de processamento; Sai do modo de Lista.

DESLIGAR - A sessão termina indicando o saldo que transita para a próxima sessão e listando o conteúdo da lista telefónica.

Se quiserem podem acrescentar mais funcionalidades a este telemóvel. Usem a imaginação.