

Métodos de Programação III

LESI + LMCC (3º ano)

Exame de 1ª Época – 1ª Chamada

Data: 24 de Janeiro de 2003

Hora: 09:30

Dispõe de 2:30 horas para realizar este exame

1 Questão: Definição de ERs

- a) Considere um alfabeto definido por $\{a, b\}$. Escreva uma expressão regular que defina uma linguagem cujas frases podem ter qualquer número de símbolos 'a' e 'b', mas garantindo um número ímpar de símbolos 'b'.
- b) Sejam r e s duas expressões regulares, não anuláveis. Considere a equação $X = rX + s$. Encontre uma expressão regular X que seja a solução da equação.

2 Questão: ERs, Sistemas de Produção e Lex

Suponha que o texto abaixo:

Era uma vez um Rei, nascido em Agosto de 1310, que tinha sido baptizado em Janeiro de 1311 por 2 Fadas. 1 Boa e outra Má.

Este Rei vivia em ALMEDINA. Sempre deprimido, e triste. Sem ânimo, nem vontade para nada.

Seu Reino decaia de dia para dia.

foi passado, como entrada, a um *filtro de texto* que o devolveu, à saída, com o seguinte conteúdo:

era uma vez um rei nascido em 8 de #N# que tinha sido baptizado em 1 de #N# por #N# fadas. #N# boa e outra má!

este rei vivia em almedina. sempre deprimido e triste. sem ânimo nem vontade para nada!

seu reino decaia de dia para dia!

Responda, então, às questões seguintes:

- a) Escreva uma especificação Lex que pudesse ser usada para gerar automaticamente um *filtro de texto* com o comportamento acima exemplificado.
- b) Diga porque é que se pode afirmar que um programa, como os *filtros de texto* gerados pelo Lex, pode ser encarado como um *sistema de produção*, justificando precisamente com base na definição de Sistema de Produção.

3 Questão: Modelação com Autómatos

Desenhe um Autómato Determinista Reactivo (com acções associadas às transições) que modele o sistema de controlo de um elevador (SCE) num prédio de 3 andares, cujo funcionamento se descreve a seguir de forma simplificada.

O SCE recebe como entrada: os sinais de chamada a cada um dos 3 andares possíveis (originado dentro da cabine ou no seu exterior); os sinais de aproximação a cada um dos 3 andares; o sinal de paragem total; e o sinal de portas livres. Como saída pode produzir as seguintes ordens: subir; descer; parar (quando se aproxima do andar pretendido); abrir portas (quando está parado); e fechar portas (quando estas estão livres). O SCE só recebe e processa um sinal de chamada quando está parado num andar de portas fechadas e, então, avança para o andar indicado (se for aquele onde está, apenas abre as portas) até cumprir totalmente a ordem (estacionar no andar pretendido e voltar a ficar com as portas fechadas); ignora qualquer outro sinal de chamada até ter completado a acção determinada pelo sinal recebido .

4 Questão: ERs, ANDs e ADs

Considere as expressões regulares abaixo,

$$p = (a^* + b^*) (a + b) c (c + b^+)$$

$$q = (a^+ + b^+) (c c + c + c b^+)$$

$$r = c (a b^* + \epsilon) c$$

e responda às seguintes alíneas:

- a) Mostre que as ERs p e q não geram a mesma linguagem.
- b) Desenhe (informalmente) o autómato determinista que define a mesma linguagem que a ER p.
- c) Converta a ER r num autómato determinista, passando pelo respectivo autómato não-determinista, mostrando cada passo formal dos processos de conversão.

- d) Escreva a expressão regular correspondente ao seguinte autômato determinista, definido em Haskell, sistematizando o processo de conversão (se possível, apresente regras que sistematizem a conversão em causa):

$\mathcal{A}_1 = (\mathcal{V}, \mathcal{Q}, \mathcal{S}, \mathcal{Z}, \delta)$, com $\mathcal{V} = \{a, b, c\}$, $\mathcal{Q} = \{1, 2, 3, 4\}$, $\mathcal{S} = 1$, $\mathcal{Z} = \{4\}$ e

$$\delta \ 1 \ a = 2$$

$$\delta \ 1 \ c = 4$$

$$\delta \ 2 \ a = 3$$

$$\delta \ 2 \ b = 3$$

$$\delta \ 3 \ b = 3$$

$$\delta \ 3 \ c = 4$$

- e) Considere o autômato \mathcal{A}_1 da alínea anterior, e o autômato \mathcal{A}_1' , obtido pela inversão de \mathcal{A}_1 (de acordo com as regras definidas nas fichas práticas), e depois transformado em determinista.

Escreva uma função Haskell `dfaintercept`, que recebe os dois autômatos, \mathcal{A}_1 e \mathcal{A}_1' , e uma frase como parâmetros, e verifica se essa frase pertence à linguagem $L(\mathcal{A}_1) \cap L(\mathcal{A}_1')$.

5 Questão: escrita de uma gramática

Pretende-se desenvolver uma linguagem para descrever os cães concorrentes à **Exposição Canina de Raças Portuguesas do Soajo**. A linguagem permitirá descrever um ou mais concorrentes, devendo fornecer para cada cão os seguintes dados: nome do animal (string), raça (string), idade (número), peso (número), nome do dono (string) e, opcionalmente, uma pequena descrição do animal (texto). Responda, então, às seguintes alíneas:

- a) Escreva, usando a notação do Lex, as expressões regulares `string` (apenas um identificador, só com letras), `número` (inteiro ou decimal) e `texto` (qualquer texto livre entre chavetas) que definem os símbolos pseudo-terminais (também designados por terminais variáveis ou classes terminais).
- b) Escreva uma gramática independente de contexto (GIC) que defina a linguagem pretendida.
- c) Dê um exemplo de uma frase válida da sua linguagem, apresentando a respectiva árvore de derivação.

6 Questão: Gramáticas, Linguagens e Parsing

Analise atentamente a seguinte GIC, cujo Símbolo Inicial é `Tarefas`, em que todos os símbolos escritos em minúsculas são pseudo-terminais (não há palavras-reservadas e os sinais estão entre apóstrofes) e `&` representa a string nula.

```
Tarefas --> Tarefa RT '.'
RT      --> ';' Tarefa RT
RT      --> &
Tarefa  --> Aluno Projecto
Aluno   --> '[' Nome ',' Morada bi ',' nif ']'
Projecto --> '[' Titulo NomeResp DescTrab ']'
Nome    --> str
Morada  --> str ','
        | &
Titulo  --> str
NomeResp --> str
DescTrab --> str
```

Responda, então, às alíneas seguintes:

- Explique, por palavras suas, qual é a linguagem gerada por essa GIC.
- Comenta a seguinte afirmação, avaliando a sua veracidade: *um parser recursivo descendente puro para a GIC acima terá 18 funções de reconhecimento de símbolos.*
- Escreva a função de um parser recursivo descendente para reconhecer o símbolo `RT`.
- Admitindo que se pretende usar a linguagem Haskell para implementar as funções de reconhecimento e processamento das frases da linguagem gerada pela gramática acima, suponha que já existem funções para fazer o parsing dos símbolos `Aluno` e `Projecto` que retornam como resultado uma string (o nome do aluno no 1º caso e o título do projecto no 2º caso).

Escreva, então em Haskell e usando *combinadores de parsing*, uma função para reconhecer o símbolo `Tarefa` e retornar um par com o nome do aluno e o título do projecto.