

Métodos de programação III

LMCC & LESI, Universidade do Minho

Ano lectivo 2004/2005

Ficha Teórico-Prática N°3

Este texto está escrito em **literate Haskell**. Isto é, pode ser interpretado como um documento \LaTeX ou como um puro programa na linguagem Haskell. Responda às perguntas sobre Haskell neste próprio ficheiro para assim produzir o programa e a sua documentação.

1 Autómatos Finitos Deterministas

1.1 Defina um autómato finito determinista para as seguintes linguagens regulares:

1. $a + ab + abc$
2. a^+b^*
3. $\{aa\}^*$

1.2 Escreva em Haskell o tipo de dados que modela um autómato finito determinista.

Solução

```
--
-- Módulo de Autómatos Finitos Deterministas em Haskell
--
-- Métodos de Programação III
-- Universidade do Minho
-- 2004/2005
--

module Dfa where

import RegExp

data Dfa ...
```

1.3 Defina as seguintes funções sobre autómatos finitos deterministas:

- Função **dfawalk** que "caminha" num autómato finito determinista consumindo os símbolos de uma string dada como argumento. A função recebe como argumentos a função de transição de estados, um estado e a string a processar. **dfawalk** dá como resultado o estado atingido após consumir todos os símbolos da string.
- Função **dfaaccept** que dado um autómato e uma string indica se a string pertence à linguagem definida pelo autómato.

Solução

```
-- dfawalk ::
```

```
-- dfaaccept ::
```

1.4 Recorde a matéria de Métodos de Programação I e mostre que a função **dfaaccept** pode ser escrita como um catamorfismo. Mais precisamente recorrendo à função (pre-definida) **foldl** ("folding" a partir da esquerda).

```
foldl :: (a -> b -> a) -> a -> [b] -> a
foldl f st [] = st
foldl f st (x:xs) = foldl f (f st x) xs
```

Solução

1.5 Considere o seguinte autómato finito determinista $\mathcal{A}_1 = (\mathcal{V}, \mathcal{Q}, \mathcal{S}, \mathcal{Z}, \delta)$, com $\mathcal{V} = \{a, b, c\}$, $\mathcal{Q} = \{1, 2, 3, 4\}$, $\mathcal{S} = 1$, $\mathcal{Z} = \{4\}$ e

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

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

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

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

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

Modele este autómato em Haskell e responda às seguintes perguntas:

1. "abcc" $\in \mathcal{L}(\mathcal{A}_1)$?
2. Qual o estado atingido processando a string "ab" e partindo do estado 1?
3. A string "aba" pertence à linguagem $\mathcal{L}(\mathcal{A}_1)$? Qual o resultado produzido pela função `dfaaccept`? Como pode ser resolvido este "problema"?

Solução _____

```
-- a_1 =
```

1.6 Prove que a expressão regular $p = abc^* + c$ e o autómato finito determinista \mathcal{A}_1 não definem a mesma linguagem, i.e., $\mathcal{L}(p) \neq \mathcal{L}(\mathcal{A}_1)$. Recorra à implementação em Haskell para efectuar a prova.

Solução _____

2 AFD para a linguagem dos números reais

2.1 Considere a linguagem dos números reais definida pela seguinte expressão regular:

$$('+' + '- ')? \text{digito}^* ('. ')? \text{digito}^+$$

Modele esta linguagem em Haskell como uma expressão regular e como um autómato finito determinista. Verifique se a string "+123.3456" pertence à linguagem utilizando as funções `matches'` e `dfaaccept`, respectivamente. Indique qual das duas funções é mais eficiente a produzir o resultado, utilizando para tal o número de reduções efectuado pelo interpretador Hugs.

Solução _____