

# Cálculo de Programas

2.º ano

Lic. Ciências da Computação e Mestrado Integrado em Engenharia Informática  
UNIVERSIDADE DO MINHO

2019/20 - Ficha nr.º 13 (última)

1. Repare que as projecções  $A \xleftarrow{\pi_1} A \times B \xrightarrow{\pi_2} B$  são funções binárias e como tal podem ser “curried”,  $A^B \xleftarrow{\overline{\pi_1}} A \xrightarrow{\overline{\pi_2}} B^B$ . Verifica-se que:

$$\overline{\pi_1} = \text{const} \quad (\text{F1})$$

$$\overline{\pi_2} = \text{id} \quad (\text{F2})$$

onde  $\text{const } a = \underline{a}$ , a função constante que dá  $a$  como resultado. Apresente justificações para os passos das provas respectivas que se seguem:

|  |   |
|--|---|
| $\begin{aligned} & \overline{\pi_1} = \text{const} \\ \equiv & \{ \dots \} \\ & \text{ap} \cdot (\text{const} \times \text{id}) = \pi_1 \\ \equiv & \{ \dots \} \\ & \text{ap} \cdot (\text{const} \times \text{id}) (a, b) = \pi_1 (a, b) \\ \equiv & \{ \dots \} \\ & \text{ap} (\text{const } a, b) = a \\ \equiv & \{ \dots \} \\ & \text{const } a \ b = a \\ \equiv & \{ \dots \} \\ & \underline{a} \ b = a \\ & \square \end{aligned}$ | $\begin{aligned} & \overline{\pi_2} = \text{id} \\ \equiv & \{ \dots \} \\ & \text{ap} \cdot (\text{id} \times \text{id}) = \pi_2 \\ \equiv & \{ \dots \} \\ & \text{ap} ((\text{id} \times \text{id}) (a, b)) = b \\ \equiv & \{ \dots \} \\ & \text{ap} (\text{id}, b) = b \\ \equiv & \{ \dots \} \\ & b = b \\ & \square \end{aligned}$ |
|--|---|

2. Com base em (F2) demonstre

$$\underline{g} = \overline{g \cdot \pi_2} \quad (\text{F3})$$

desenhando um diagrama explicativo dos tipos em jogo.

3. Baseie-se em (F3) para mostrar que, no mónade de acções sobre um estado,  $\text{St } S \ A = (A \times S)^S$ , se tem

$$\text{do } \{x; y\} = x \gg y = y \cdot \pi_2 \cdot x \quad (\text{F4})$$

ou seja: o resultado da acção  $x$  é ignorado pela acção  $y$ . **Sugestão:** recorde das aulas teóricas que, neste mónade,  $(\gg f) = \widehat{f}^S$ .

4. Considere as seguintes acções que habitam o mónade de estado  $\text{St } S \ A = (A \times S)^S$ , para um dado espaço de estados  $S$ :

$$\text{query } f = \langle f, \text{id} \rangle \quad (\text{F5})$$

$$\text{modify } g = \langle !, g \rangle \quad (\text{F6})$$

A acção *query*  $f$  interroga o estado aplicando-lhe a observação  $f$  e deixando-o inalterado; já a acção *modify*  $g$  recorre a  $g$  para actualizar o valor corrente do estado, dando como resultado um mero “acknowledgement” da acção realizada.

Mostre que as igualdades

$$\text{do } \{ \text{modify } \text{id}; \text{query } g \} = \text{query } g \quad (\text{F7})$$

$$\text{do } \{ \text{modify } f; \text{modify } g \} = \text{modify } (g \cdot f) \quad (\text{F8})$$

se verificam.

5. Nos vídeos ds aulas teóricas aparece a função *mmap* (localize-a), que resultou da *monadificação* da função *map* das listas, e a função *sequence* (localize-a também). Verifica-se que *sequence* = *mmap*  $f$  para um dado  $f$ . Qual? Responda com base na inspecção dos tipos das duas funções.
6. Localize a definição das acções *get* e *put* sobre o mónade de estado nos vídeos das aulas teóricas (ou nos apontamentos da cadeira) e demonstre a igualdade

$$\text{do } \{ a \leftarrow \text{get}; \text{put } a \} = \text{get} \gg \text{put} = \text{modify } \text{id} \quad (\text{F9})$$

7. O seguinte esboço de código diz respeito a uma função

```
accum :: Num b => LTree b -> LTree b
accum t = ..... B.....
```

que, a partir de uma árvore de folhas de inteiros, produz uma outra árvore (com a mesma forma) em que o conteúdo de cada folha é substituído pela sua soma com os conteúdos de todas as folhas à sua esquerda, recorrendo à função auxiliar:

```
aux :: Num a => LTree a -> St a (LTree a)
aux (Leaf x) = ..... A.....
aux (Node esq dir) = do {
    esq' <- aux esq;
    dir' <- aux dir
    return (Node esq' dir')
}
```

Por exemplo, *accum* (*Node* (*Leaf* 1) (*Node* (*Leaf* 2) (*Leaf* 3))) resulta em

*Node* (*Leaf* 1) (*Node* (*Leaf* 3) (*Leaf* 6)).

Com base em exemplos semelhantes das aulas teóricas, complete o código em *A* e *B*.