Cálculo de Programas
Algebra of Programming

UNIVERSIDADE DO MINHO
Lic. em Engenharia Informática (3º ano)
Lic. Ciências da Computação (2º ano)

2022/23 - Ficha *(Exercise sheet)* nr. 10

1. O isomorfismo $\mathsf{in} : \mathsf{B}\ (A, \mathsf{T}\ A) \to \mathsf{T}\ A$ construtor dos habitantes de um tipo recursivo (paramétrico) de base B é ele próprio paramétrico em A. Complete o seguinte diagrama que capta a propriedade natural (grátis) de in:

   Isomorphism $\mathsf{in} : \mathsf{B}\ (A, \mathsf{T}\ A) \to \mathsf{T}\ A$ *constructing inhabitants of a recursive (parametric) base type* B *is itself parametric on* A*. Complete the following diagram that captures the natural (free) property of* in*:*

$$\begin{tikzcd}
\mathsf{T}\ A \arrow[d, "\mathsf{T}\ f"'] & \mathsf{B}\ (A, \mathsf{T}\ A) \arrow[l, "\mathsf{in}"'] \arrow[d, "?"] \\
\mathsf{T}\ A' & \mathsf{B}\ (A', \mathsf{T}\ A') \arrow[l, "\mathsf{in}"]
\end{tikzcd}$$

   Instancie essa propriedade para listas, em que

   Instantiate this property for lists, where

$$\begin{cases} \mathsf{T}\ A = A^* \\ \mathsf{B}\ (X, Y) = 1 + X \times Y \\ \mathsf{T}\ f = \mathsf{map}\ f \end{cases}$$

   Desenvolva essa igualdade até chegar à sua formulação sem qualquer dos construtores *pointfree* estudados nesta disciplina. O que é que obteve, afinal?

   Unfold the equality until a formulation is reached involving none of the pointfree *constructors studied in this course. What did you get, after all?*

---

2. O formulário desta disciplina apresenta duas definições alternativas para o functor $\mathsf{T}\ f$, uma como *catamorfismo* e outra como *anamorfismo*. Identifique-as e acrescente justificações à seguinte prova de que essas definições são equivalentes:

   The reference sheet of this course presents two alternative definitions for functor $\mathsf{T}\ f$*, one as a* catamorphism *and another as an* anamorphism*. Identify them and fill in justifications in the following proof that such definitions are equivalent:*

$$\begin{array}{ll}
& \mathsf{T}\ f = (\!|\ \mathsf{in} \cdot \mathsf{B}\ (f, id)\ |\!) \\
\equiv & \quad \{ \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad \} \\
& \mathsf{T}\ f \cdot \mathsf{in} = \mathsf{in} \cdot \mathsf{B}\ (f, id) \cdot \mathsf{F}\ (\mathsf{T}\ f) \\
\equiv & \quad \{ \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad \}
\end{array}$$

$$\begin{array}{cl} & \mathsf{T}\ f \cdot \mathsf{in} = \mathsf{in} \cdot \mathsf{B}\ (id, \mathsf{T}\ f) \cdot \mathsf{B}\ (f, id) \\ \equiv & \qquad \{ \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad \} \\ & \mathsf{out} \cdot \mathsf{T}\ f = \mathsf{F}\ (\mathsf{T}\ f) \cdot \mathsf{B}\ (f, id) \cdot \mathsf{out} \\ \equiv & \qquad \{ \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad \} \\ & \mathsf{T}\ f = [\![\mathsf{B}\ (f, id) \cdot \mathsf{out}]\!] \\ \Box & \end{array}$$

---

3. Mostre que o catamorfismo de listas $\mathsf{length} = (\![[\mathsf{zero}\ , \mathsf{succ} \cdot \pi_2]]\!)$ é a mesma função que o *anamorfismo* de naturais $[\!((id + \pi_2) \cdot \mathsf{out}_{\mathsf{List}})\!]$.

Show that the list catamorphism $\mathsf{length} = (\![[\mathsf{zero}\ , \mathsf{succ} \cdot \pi_2]]\!)$ *is the same function as the* $\mathbb{N}_0$-anamorphism $[\!((id + \pi_2) \cdot \mathsf{out}_{\mathsf{List}})\!]$.

---

4. Recorra às leis dos catamorfismos para verificar a propriedade natural

Use the laws of catamorphisms to check the natural property

$$(\mathsf{LTree}\ f) \cdot mirror \quad = \quad mirror \cdot (\mathsf{LTree}\ f) \tag{F1}$$

onde $mirror$ é o catamorfismo

where $mirror$ is the catamorphism

$$\begin{cases} mirror :: \mathsf{LTree}\ a \rightarrow \mathsf{LTree}\ a \\ mirror = (\![\mathsf{in} \cdot (id + \mathsf{swap})]\!) \end{cases} \tag{F2}$$

que "espelha"uma árvore e

that "mirrors" a tree, and

$$\mathsf{LTree}\ f = (\![\mathsf{in} \cdot (f + id)]\!)$$

é o correspondente functor de tipo.

is the corresponding type functor.

---

5. Mostre que o anamorfismo que calcula os sufixos de uma lista

Show that the anamorphism that computes the suffixes of a list

$$suffixes = [\!(g)\!]\ \textbf{where}\ g = (id + \langle \mathsf{cons}, \pi_2 \rangle) \cdot \mathsf{out}$$

é a função:

is the function:

$$\begin{array}{l} suffixes\ [\,] = [\,] \\ suffixes\ (h : t) = (h : t) : suffixes\ t \end{array}$$

---

6. Mostre que a função $mirror$ (F2) se pode definir como o anamorfismo

Show that the function $mirror$ (F2) can be defined as the anamorphism

$$mirror = [\!((id + \mathsf{swap}) \cdot \mathsf{out})\!] \tag{F3}$$

onde out é a conversa de in. Volte a demonstrar a propriedade $mirror \cdot mirror = id$, desta vez recorrendo à lei de fusão dos anamorfismos.

where out *is the converse of* in. *Prove $mirror \cdot mirror = id$ again, this time resorting to the fusion-law of anamorphisms.*

---

7. O facto de $\mathsf{length} : A^* \to \mathbb{N}_0$ poder ser definida tanto como *catamorfismo* de listas como *anamorfismo* de naturais (questão 3) pode generalizar-se da forma seguinte: sejam dados dois tipos indutivos

The fact that $\mathsf{length} : A^* \to \mathbb{N}_0$ *is both a list-*catamorphism *and a* $\mathbb{N}_0$-anamorphism *(question 3) generalizes as follows: let two inductive types be given*

$$\begin{tikzcd} \mathsf{T}_1 \arrow[rr, bend left, "\mathsf{out}_1"] \arrow[rr, phantom, "\cong"] & & \mathsf{F}\ \mathsf{T}_1 \arrow[ll, bend left, "\mathsf{in}_1"] \end{tikzcd} \qquad \begin{tikzcd} \mathsf{T}_2 \arrow[rr, bend left, "\mathsf{out}_2"] \arrow[rr, phantom, "\cong"] & & \mathsf{G}\ \mathsf{T}_2 \arrow[ll, bend left, "\mathsf{in}_2"] \end{tikzcd}$$

e $\alpha : \mathsf{F}\ X \to \mathsf{G}\ X$, isto é, α satisfaz a propriedade *grátis*:

and $\alpha : \mathsf{F}\ X \to \mathsf{G}\ X$*, that is,* α *satisfying the* free *property:*

$$\mathsf{G}\ f \cdot \alpha = \alpha \cdot \mathsf{F}\ f \tag{F4}$$

Então $(\!|\, \mathsf{in}_2 \cdot \alpha \,|\!) = [\!(\alpha \cdot \mathsf{out}_1)\!]$, como se mostra a seguir (complete as justificações):

Then $(\!|\, \mathsf{in}_2 \cdot \alpha \,|\!) = [\!(\alpha \cdot \mathsf{out}_1)\!]$*, as shown below (complete the justifications):*

$$\begin{array}{ll} & k = (\!|\, \mathsf{in}_2 \cdot \alpha \,|\!) \\ \equiv & \quad \{ \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad \} \\ & k \cdot \mathsf{in}_1 = \mathsf{in}_2 \cdot \alpha \cdot \mathsf{F}\ k \\ \equiv & \quad \{ \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad \} \\ & \mathsf{out}_2 \cdot k = \mathsf{G}\ k \cdot \alpha \cdot \mathsf{out}_1 \\ \equiv & \quad \{ \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad \} \\ & k = [\!(\alpha \cdot \mathsf{out}_1)\!] \end{array}$$

$\square$

Finalmente, identifique T_1, T_2 e α para o caso de $k = \mathsf{length}$.

Finally, identify T_1*,* T_2 *and* α *in the case* $k = \mathsf{length}$.