# Cálculo de Programas
# *Algebra of Programming*

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

2023/24 - Ficha *( Exercise sheet )* nr. 11

1. Considere o anamorfismo $r = [\!(g)\!]$ em que

   *Consider the anamorphism $r = [\!(g)\!]$ where $g$ is*

$$
\begin{array}{l}
g\ [\,] = i_1\ ()\\
g\ x = i_2\ (\mathsf{last}\ x, \mathsf{init}\ x)
\end{array}
$$

   onde $\mathsf{last}\ x$ dá o último elemento da lista $x$ e $\mathsf{init}\ x$ dá $x$ sem esse último elemento. O que faz a função $r$? Responda informalmente desenhando o diagrama de $r$.

   *in which $\mathsf{last}\ x$ gives the last element of list $x$ and $\mathsf{init}\ x$ gives $x$ without this last element. What does $r$ do? Answer informally by drawing the diagram of $r$.*

---

2. Considere a função:

   *Let function*

$$x \ominus y = \textbf{if}\ x \leqslant y\ \textbf{then}\ 0\ \textbf{else}\ 1 + x \ominus (y+1)$$

   Quais os valores das expressões $(3 \ominus 2) \ominus 3$ e $(3 \ominus 4) + 4$? Codifique $\widehat{\ominus} : \mathbb{N}_0 \times \mathbb{N}_0 \to \mathbb{N}_0$ como um anamorfismo de naturais e faça o respectivo diagrama.

   *be given. Evaluate $(3 \ominus 2) \ominus 3$ and $(3 \ominus 4) + 4$ and encode $\widehat{\ominus} : \mathbb{N}_0 \times \mathbb{N}_0 \to \mathbb{N}_0$ as an anamorphism over $\mathbb{N}_0$. Draw the corresponding diagram.*

---

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

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

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

   Instancie essa propriedade para listas, em que

   *Instantiate this property for lists, where*

$$
\left\{
\begin{array}{l}
\mathsf{T}\ A = A^*\\
\mathsf{B}\ (X, Y) = 1 + X \times Y\\
\mathsf{T}\ f = \mathsf{map}\ f
\end{array}
\right.
$$

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?*

---

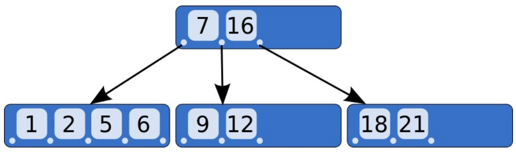
4. Uma "B-tree"é uma generalização das árvores binárias do módulo BTree a mais do que duas sub-árvores por nó:

*A "B-tree"is a generalization of the binary trees of the* BTree *module to more than two subtrees per node:*

$$\mathbf{data}\ B\_tree\ a = Nil \mid Block\ \{ leftmost :: B\_tree\ a, block :: [(a, B\_tree\ a)] \}$$

Por exemplo, a B-tree

*For instance, the B-tree*




é representada no tipo acima por:

*is represented by the data type above as:*

$$\begin{array}{l}
t = Block\ \{ \\
\quad leftmost = Block\ \{ \\
\quad\quad leftmost = Nil, \\
\quad\quad block = [(1, Nil), (2, Nil), (5, Nil), (6, Nil)] \}, \\
\quad block = [ \\
\quad\quad (7, Block\ \{ \\
\quad\quad\quad leftmost = Nil, \\
\quad\quad\quad block = [(9, Nil), (12, Nil)] \}), \\
\quad\quad (16, Block\ \{ \\
\quad\quad\quad leftmost = Nil, \\
\quad\quad\quad block = [(18, Nil), (21, Nil)] \}) \\
\quad\quad ] \}
\end{array}$$

Identifique, justificando, o functor de base

*Identify the base functor*

$$\left\{ \begin{array}{l} \mathsf{B}\ (X, Y) = \ldots \\ \mathsf{B}\ (f, g) = \ldots . \end{array} \right. \tag{F1}$$

que capta o padrão de recursividade da declaração de $B\_tree$ dada acima, em Haskell, bem como o isomorfismo:

*which captures the recursion pattern of the declaration of $B\_tree$ given above, in Haskell, as well as the isomorphism:*

$$\mathsf{in} : \mathsf{B}\ (A, B\_tree\ A) \rightarrow B\_tree\ A$$

---

5. O algoritmo "bubble-sort" é o ciclo-for

*The "bubble-sort" algorithm is a for-loop:*

$$\begin{array}{l}
bSort\ xs = \mathsf{for}\ \mathsf{bubble}\ xs\ (\mathsf{length}\ xs)\ \mathbf{where} \\
\quad \mathsf{bubble}\ (x : y : xs) \\
\quad\quad \mid x > y = y : \mathsf{bubble}\ (x : xs) \\
\quad\quad \mid \mathsf{otherwise} = x : \mathsf{bubble}\ (y : xs) \\
\quad \mathsf{bubble}\ x = x
\end{array}$$

cujo corpo de ciclo é um hilomorfismo $\mathsf{bubble} = [\![ conquer, divide ]\!]$. Identifique os genes *divide* e *conquer* desse hilomorfismo.

*Its loop-body is a hylomorphism* $\mathsf{bubble} = [\![ conquer, divide ]\!]$. *Identify the genes divide and conquer of this hylomorphism.*

---

6. Todo o ciclo-*while* que termina pode ser definido por

*Every terminating* while-*loop can be defined by*

$$\mathbf{while}\ p\ f\ g = \mathbf{tailr}\ ((g + f) \cdot (\neg \cdot p)?) \tag{F2}$$

recorrendo ao combinador de *"tail recursion"*

*using the* "tail recursion" *combinator*

$$\mathbf{tailr}\ f = [\![ \mathsf{join}, f ]\!] \tag{F3}$$

que é um hilomorfismo de base $\mathsf{B}\ (X, Y) = X + Y$, para $\mathsf{join} = [id\ , id]$.

*which is a hylomorphism of basis* $\mathsf{B}\ (X, Y) = X + Y$, *for* $\mathsf{join} = [id\ , id]$.

Derive a definição *pointwise* de $\mathbf{while}\ p\ f\ g$, sabendo que qualquer $h = [\![ f, g ]\!]$ que termina é tal que $h = f \cdot \mathsf{F}\ h \cdot g$.

*Derive the* pointwise *definition of* $\mathbf{while}\ p\ f\ g$, *knowing that any terminating* $h = [\![ f, g ]\!]$ *is such that* $h = f \cdot \mathsf{F}\ h \cdot g$.

---

7. Considere a seguinte lei de fusão de $\mathbf{tailr}$, válida sempre que $(\mathbf{tailr}\ g) \cdot f$ termina:

*Consider the following fusion-law of* $\mathbf{tailr}$, *valid whenever* $(\mathbf{tailr}\ g) \cdot f$ *terminates:*

$$(\mathbf{tailr}\ g) \cdot f = \mathbf{tailr}\ h \ \Leftarrow\ (id + f) \cdot h = g \cdot f \tag{F4}$$

Complete a seguinte demonstração dessa lei.

*Complete de following proof of (F4).*

$$
\begin{array}{ll}
 & (\mathbf{tailr}\ g) \cdot f = \mathbf{tailr}\ h \\
\equiv & \quad \{\ \ \dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\ \ \} \\
 & (\!| \nabla |\!) \cdot [\!( g )\!] \cdot f = (\!| \nabla |\!) \cdot [\!( h )\!] \\
\Leftarrow & \quad \{\ \ \dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\ \ \} \\
 & [\!( g )\!] \cdot f = [\!( h )\!] \\
\Leftarrow & \quad \{\ \ \dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\ \ \} \\
 & g \cdot f = (id + f) \cdot h \\
\Box &
\end{array}
$$