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. 6

1. Prove a igualdade | *Prove the equality*

$$\overline{f \cdot (g \times h)} \;=\; \overline{\mathsf{ap} \cdot (id \times h)} \cdot \overline{f} \cdot g \tag{F1}$$

usando as leis das exponenciais e dos produtos. | *using the laws of products and exponentials.*

---

2. É dada a definição | *Let* flip *be defined by*

$$\mathsf{flip}\ f = \overline{\widehat{f} \cdot \mathsf{swap}} \tag{F2}$$

de acordo com: | *according to:*

$$\begin{array}{ccccccc}
(C^B)^A & \cong & C^{A \times B} & \cong & C^{B \times A} & \cong & (C^A)^B \\
f & \mapsto & \widehat{f} & \mapsto & \widehat{f}.\mathsf{swap} & \mapsto & \overline{\widehat{f} \cdot \mathsf{swap}} = \mathsf{flip}\ f
\end{array}$$

Mostre que flip é um isomorfismo por ser a sua própria inversa: | *Show that it is an isomorphism because it is its own inverse:*

$$\mathsf{flip}\ (\mathsf{flip}\ f) = f \tag{F3}$$

Mostre ainda que: | *Furthermore show:*

$$\mathsf{flip}\ f\ x\ y = f\ y\ x$$

---

3. Mostre que | *Show that*

$$junc \cdot unjunc = id \tag{F4}$$

$$unjunc \cdot junc = id \tag{F5}$$

se verificam, onde | *hold for*

$$\begin{tikzcd} A^{B+C} \arrow[r, bend left, "unjunc"] \arrow[r, phantom, "\cong"] & A^B \times A^C \arrow[l, bend left, "junc"] \end{tikzcd} \qquad \left\{ \begin{array}{l} junc\ (f, g) = [f\ , g] \\ unjunc\ k = (k \cdot i_1, k \cdot i_2) \end{array} \right. \tag{F6}$$

---

4. O código que se segue, escrito em Haskell, implementa a noção de ciclo-for, onde b é o corpo ("body") do ciclo e i é a sua inicialização:

   The following code in Haskell implements a for *-loop where b is the loop-body and i is its initialization:*

$$\left\{ \begin{array}{l} \mathsf{for}\ b\ i\ 0 = i \\ \mathsf{for}\ b\ i\ (n+1) = b\ (\mathsf{for}\ b\ i\ n) \end{array} \right. \tag{F7}$$

   Mostre que $\mathsf{for}\ b\ i = (\!|\, g \,|\!)$ para um dado g (descubra qual) de acordo com o diagrama seguinte:

   Show that $\mathsf{for}\ b\ i = (\!|\, g \,|\!)$ for some g (find which) according to the following diagram:

$$\begin{tikzcd} \mathbb{N}_0 \arrow[d, "\mathsf{for}\ b\ i"'] \arrow[rr, bend left, "\mathsf{out}"] \arrow[rr, phantom, "\cong"] & & 1+\mathbb{N}_0 \arrow[ll, bend left, "\mathsf{in}"] \arrow[d, "id+(\mathsf{for}\ b\ i)"] \\ B & & 1+B \arrow[ll, bend left, "g"] \end{tikzcd}$$

$$\text{where} \quad \left\{ \begin{array}{l} \mathsf{in} = [\mathsf{zero}\ , \mathsf{succ}] \\ \mathsf{zero}\ \_ = 0 \\ \mathsf{succ}\ n = n+1 \end{array} \right.$$

---

5. Mostre que $(a+)$ dada a seguir é um ciclo $\mathsf{for}\ b\ i$ (F7) para um dado b e um dado i — descubra quais:

   Show that $(a+)$ given next is a for-loop $\mathsf{for}\ b\ i$ *(F7) for b and i to be calculated:*

$$\left\{ \begin{array}{l} a + 0 = a \\ a + (n+1) = 1 + (a+n) \end{array} \right. \tag{F8}$$

---

6. **Questão prática** — Este problema não irá ser abordado em sala de aula. Os alunos devem tentar resolvê-lo em casa e, querendo, publicarem a sua solução no canal **#geral** do Slack, com vista à sua discussão com colegas.
   Dão-se a seguir os requisitos do problema.

   Open assignment *— This assignment will not be addressed in class. Students should try to solve it at home and, whishing so, publish their solutions in the* ***#geral*** *Slack channel, so as to trigger discussion among other colleagues.*
   The requirements of the problem are given below.

Problem requirements*: We are back to*

$$nub :: (Eq\ a) \Rightarrow [\,a\,] \rightarrow [\,a\,]$$

available from the Data.List library in Haskell and already addressed in the open assignment of exercise sheet nr.3.

We wish to re-implement nub as a pointfree *pipeline of functions which starts as indicated below*

$$nub = p \cdot \mathsf{flip\ zip}\ nat \tag{F9}$$

where $nat = [0\,..]$ is the infinite list of natural numbers.

The challenge is that neither explicit recursion nor fold operators are allowed this time.

As before, use the combinators available from library Cp.hs as well as functions over lists already available in the Haskell standard libraries. Draw the type diagram of your proposal.

□