

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

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

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

$$\text{flip } f = \overline{\hat{f} \cdot \text{swap}} \tag{F2}$$

de acordo com:

according to:

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

Mostre que flip é um isomorfismo por ser a sua própria inversa:

Show that it is an isomorphism because it is its own inverse:

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

Mostre ainda que:

Furthermore show:

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

3. Mostre que

Show that

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

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

se verificam, onde

hold for

$$A^{B+C} \begin{array}{c} \xrightarrow{\text{unjunc}} \\ \cong \\ \xleftarrow{\text{junc}} \end{array} A^B \times A^C \quad \left\{ \begin{array}{l} \text{junc} (f, g) = [f, g] \\ \text{unjunc } k = (k \cdot i_1, k \cdot i_2) \end{array} \right. \tag{F6}$$

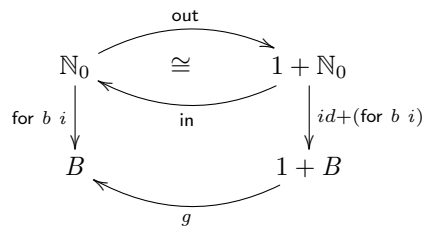
4. O código que se segue, escrito em Haskell, implementa a noção de ciclo-for, onde b é o corpo (“body”) do ciclo e i é a sua inicialização:

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

$$\begin{cases} \text{for } b \ i \ 0 = i \\ \text{for } b \ i \ (n + 1) = b \ (\text{for } b \ i \ n) \end{cases} \quad (\text{F7})$$

Mostre que $\text{for } b \ i = \langle g \rangle$ para um dado g (descubra qual) de acordo com o diagrama seguinte:

Show that $\text{for } b \ i = \langle g \rangle$ for some g (find which) according to the following diagram:



$$\text{where } \begin{cases} \text{in} = [\text{zero}, \text{succ}] \\ \text{zero } _ = 0 \\ \text{succ } n = n + 1 \end{cases}$$

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

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

$$\begin{cases} a + 0 = a \\ a + (n + 1) = 1 + (a + n) \end{cases} \quad (\text{F8})$$

6. **Questão prática** — Este problema não irá ser abordado em sala de aula. Os alunos devem tentar resolvê-lo em casa e, querendo, publicar 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, wishing 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

$$\text{nub} :: (Eq \ a) \Rightarrow [a] \rightarrow [a]$$

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

$$\text{nub} = p \cdot \text{flip zip nat} \quad (\text{F9})$$

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.

□