

Cálculo de Programas

Algebra of Programming

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

2026/27 — Ficha (*Worksheet*) nr.º 3

1. Considere o diagrama

Consider the following diagram

$$(A \times B) \times C \begin{array}{c} \xrightarrow{\text{assocr}} \\ \cong \\ \xleftarrow{\text{assocl}} \end{array} A \times (B \times C)$$

onde $\text{assocl} = \langle id \times \pi_1, \pi_2 \cdot \pi_2 \rangle$. Apresente justificações para o cálculo que se segue em que se resolve em ordem a assocr a equação $\text{assocl} \cdot \text{assocr} = id$:

where $\text{assocl} = \langle id \times \pi_1, \pi_2 \cdot \pi_2 \rangle$. The reasoning below solves the equation $\text{assocl} \cdot \text{assocr} = id$ for variable assocr . Fill in justifications for each step in the reasoning:

$$\begin{aligned} & \text{assocl} \cdot \text{assocr} = id \\ \equiv & \quad \{ \dots\dots\dots \} \\ & \left\{ \begin{array}{l} (id \times \pi_1) \cdot \text{assocr} = \pi_1 \\ \pi_2 \cdot \pi_2 \cdot \text{assocr} = \pi_2 \end{array} \right. \\ \equiv & \quad \{ \dots\dots\dots \} \\ & \left\{ \begin{array}{l} \left\{ \begin{array}{l} \pi_1 \cdot \text{assocr} = \pi_1 \cdot \pi_1 \\ \pi_1 \cdot \pi_2 \cdot \text{assocr} = \pi_2 \cdot \pi_1 \end{array} \right. \\ \pi_2 \cdot \pi_2 \cdot \text{assocr} = \pi_2 \end{array} \right. \\ \equiv & \quad \{ \dots\dots\dots \} \\ & \left\{ \begin{array}{l} \pi_1 \cdot \text{assocr} = \pi_1 \cdot \pi_1 \\ \left\{ \begin{array}{l} \pi_1 \cdot \pi_2 \cdot \text{assocr} = \pi_2 \cdot \pi_1 \\ \pi_2 \cdot \pi_2 \cdot \text{assocr} = \pi_2 \end{array} \right. \end{array} \right. \\ \equiv & \quad \{ \dots\dots\dots \} \\ & \left\{ \begin{array}{l} \pi_1 \cdot \text{assocr} = \pi_1 \cdot \pi_1 \\ \pi_2 \cdot \text{assocr} = \langle \pi_2 \cdot \pi_1, \pi_2 \rangle \end{array} \right. \\ \equiv & \quad \{ \dots\dots\dots \} \\ & \text{assocr} = \langle \pi_1 \cdot \pi_1, \pi_2 \times id \rangle \end{aligned} \tag{F1}$$

2. (a) Codifique (F1) directamente em Haskell e verifique o comportamento dessa função no GHCi; (b) De seguida, converta — por igualdade extensional — (F1) para notação Haskell <i>pointwise</i> que não recorra a nenhum combinador nem projecção e verifique no GHCi que as duas versões dão os mesmos resultados.	(a) Encode (F1) directly in Haskell and check its behavior in GHCi; (b) Then convert — by extensional equality — (F1) to pointwise Haskell code dispensing with combinators or projections, and let GHCi check that both versions give the same results.
3. Considere a função	Let
	$\beta = \text{swap} \cdot (\text{id} \times \text{swap}) \quad (\text{F2})$
Determine o tipo mais geral de β e mostre que a seguinte propriedade se verifica:	be given. Infer the most general type of α and prove the following property of β :
	$\beta \cdot (f \times (g \times h)) = ((h \times g) \times f) \cdot \beta \quad (\text{F3})$
NB: sugere-se a utilização de uma propriedade de swap que foi demonstrada na ficha anterior.	Suggestion: recall a property of swap studied in the previous worksheet.
4. Considere a função:	Consider function xr defined as follows:
	$\text{xr} = \langle \pi_1 \times \text{id}, \pi_2 \cdot \pi_1 \rangle \quad (\text{F4})$
(a) Determine o tipo de xr ; (b) Mostre que xr satisfaz a propriedade	Show that property
	$\text{xr} \cdot \langle \langle f, g \rangle, h \rangle = \langle \langle f, h \rangle, g \rangle \quad (\text{F5})$
para todo o f, g e h .	holds for all f, g and h .
5. Defina no GHCi o seguinte tipo de dados:	Define the following data type on GHCi:
	<code>data X = B Bool P (Bool, Int)</code>
Peça ao GHCi informação sobre os tipos de B e de P e deduza que são funções tais que $f = [B, P]$ faz sentido. Qual é o tipo de f ?	What are the types of B and P given by GHCi? Infer from them that $f = [B, P]$ makes sense. What is the type of f ?
NB: em Haskell a alternativa $[f, g]$ escreve-se <code>either f g</code> .	NB: $[f, g]$ is written <code>either f g</code> in Haskell.
6. Determine o tipo mais geral da função α que se segue:	Infer the most general type of the following function α :
	$\alpha = [\langle \text{FALSE}, \text{id} \rangle, \langle \text{TRUE}, \text{id} \rangle] \quad (\text{F6})$
7. Procure no formulário a propriedade universal do combinador $[f, g]$:	Recall the universal property of the $[f, g]$ combinator:
	$k = [f, g] \equiv \begin{cases} k \cdot i_1 = f \\ k \cdot i_2 = g \end{cases}$

Usando essa propriedade, mostre que α acima (F6) se pode escrever em Haskell da forma seguinte:

Then show, using the $+$ -universal law, that α can be written in pointwise Haskell as follows:

$$\begin{cases} \alpha (i_1 \ a) = (\text{FALSE}, a) \\ \alpha (i_2 \ a) = (\text{TRUE}, a) \end{cases}$$

Codifique α e teste-a no GHCi, onde i_1 (resp. i_2) se escreve `Left` (resp. `Right`).

Encode α and test it on GHCi, where i_1 (resp. i_2) is written `Left` (resp. `Right`).

8. Demonstre a igualdade

Prove the equality

$$[\underline{k}, \underline{k}] = \underline{k} \quad (\text{F7})$$

recorrendo à propriedade universal- $+$ e a uma lei que qualquer função constante $\underline{k}$ satisfaz. (Consultar o formulário.)

using the $+$ -universal property and a law that any constant function $\underline{k}$ satisfies. (Check the reference sheet.)

9. **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. Os requisitos do problema são dados abaixo. **NB:** usa-se a notação X^* para designar o tipo $[X]$ em Haskell.

Open assignment — This assignment will not be addressed in class. Students should try to solve it at home and, wishing so, publish their solution in the **#geral** Slack channel, so that it can be discussed among colleagues. The requirements of the problem are given below. **NB:** notation X^* is used to denote the type $[X]$ in Haskell.

Problem requirements:

The automatic generation of bibliographies in the $\text{\LaTeX}$ text preparation system is based bibliographic databases from which the following information can be extracted:

$$\text{Bib} = (\text{Key} \times \text{Aut}^*)^*$$

It associates authors (Aut) to citation keys (Key).

Whenever $\text{\LaTeX}$ processes a text document, it compiles all occurrences of citation keys in an auxiliary file

$$\text{Aux} = (\text{Pag} \times \text{Key}^*)^*$$

associating pages (Pag) to the citation keys that occur in them.

An **author index** is an appendix to a text (e.g. book) indicating, in alphabetical order, the names of authors mentioned and the ordered list of pages where their works are cited, for example:

Arbib, M. A. – 10, 11
 Bird, R. – 28
 Horowitz, E. – 2, 3, 15, 16, 19
 Hudak, P. – 11, 12, 29
 Jones, C. B. – 3, 7, 28
 Manes, E. G. – 10, 11
 Sahni, S. – 2, 3, 15, 16, 19
 Spivey, J.M. – 3, 7
 Wadler, P. – 2, 3

The above structure can be represented by the type

$$Ind = (Aut \times Pag^*)^*$$

listing authors (Aut) and the respective pages where they are mentioned (Pag).

Write a Haskell function $mkInd : Bib \times Aux \rightarrow Ind$ that generates author indices (Ind) from Bib and Aux .

Important: Structure your solution across the $f \cdot g$, $\langle f, g \rangle$ and $f \times g$ combinators that can be found in library `Cp.hs`. Use **diagrams** to plan your proposed solution, which should avoid re-inventing functions over lists already available in the Haskell standard libraries.

□