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. 3

1. Recorde a propriedade universal do combinador $[f\,,g]$,

   Recall the universal property of the $[f\,,g]$ combinator,

$$k = [f\,,g] \quad \equiv \quad \left\{ \begin{array}{l} k \cdot i_1 = f \\ k \cdot i_2 = g \end{array} \right.$$

   da qual, como sabe, podem ser derivadas todas as outras que aparecem no respectivo grupo, no formulário.
   Use esta lei para demonstrar a lei

   from which, as you know, all the others appearing in the corresponding group of the reference sheet can be derived.
   Use this property to prove the law

$$[i_1\,,i_2] = id$$

   conhecida por *reflexão*-$+$.

   known as $+$-reflexion.

---

2. Seja dada a função $\mathsf{coswap} = [i_2\,,i_1]$. Faça um diagrama que explique o tipo de coswap e mostre, usando o cálculo de programas, que $\mathsf{coswap} \cdot \mathsf{coswap} = id$.

   Let function $\mathsf{coswap} = [i_2\,,i_1]$ *be given. Draw a diagram explaining the type of* coswap *and show, using the algebra of programming, that* $\mathsf{coswap} \cdot \mathsf{coswap} = id$ *holds.*

---

3. Demonstre a igualdade

   Prove the equality

$$[\underline{k}\,,\underline{k}] = \underline{k} \tag{F1}$$

   recorrendo à propriedade universal acima e a uma lei que qualquer função constante $\underline{k}$ satisfaz. (Ver no formulário.)

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

---

4. Considere a função

   Let function

$$\alpha = [\langle \underline{\textsc{False}}, id \rangle\,, \langle \underline{\textsc{True}}, id \rangle] \tag{F2}$$

   Determine o tipo de α e mostre, usando a propriedade *universal*-$+$, que α se pode escrever em Haskell da forma seguinte:

   be given. Infer the type of α and show, using the $+$-universal law, *that α can be written in pointwise Haskell as follows:*

$$\alpha\ (\mathsf{Left}\ a) = (\textsc{False}, a)$$
$$\alpha\ (\mathsf{Right}\ a) = (\textsc{True}, a)$$

---

5. Recorra às leis dos coprodutos para mostrar que a definição que conhece da função factorial,

Use the coproduct laws to show that the usual definition of the factorial function,

$$\begin{array}{l} fac\ 0 = 1 \\ fac\ (n+1) = (n+1) * fac\ n \end{array}$$

é equivalente à equação seguinte

is equivalent the following equation,

$$fac \cdot [\underline{0}, \mathsf{succ}] = [\underline{1}, \mathsf{mul} \cdot \langle \mathsf{succ}, fac \rangle]$$

onde

where

$$\begin{array}{l} \mathsf{succ}\ n = n + 1 \\ \mathsf{mul}\ (a, b) = a * b \end{array}$$

---

6. O combinador funcional *soma* define-se por: $f + g = [i_1 \cdot f, i_2 \cdot g]$. Identifique no formulário os nomes das propriedades que se seguem e demonstre-as usando o cálculo de programas.

The sum of two functions f and g is defined by $f + g = [i_1 \cdot f, i_2 \cdot g]$. Check the names of the three laws that are given below in the reference sheet and prove them using the algebra of programming.

$$id + id = id \qquad \text{(F3)}$$
$$(f + g) \cdot i_1 = i_1 \cdot f \qquad \text{(F4)}$$
$$(f + g) \cdot i_2 = i_2 \cdot g \qquad \text{(F5)}$$

---

7. Considere o isomorfismo

Consider the isomorphism

$$\begin{tikzcd} (A+B)+C \arrow[rr, bend left, "\mathsf{coassocr}"] \arrow[r, phantom, "\cong"] & & A+(B+C) \arrow[ll, bend left, "\mathsf{coassocl}"] \end{tikzcd}$$

onde $\mathsf{coassocr} = [id + i_1, i_2 \cdot i_2]$. Calcule a sua conversa resolvendo em ordem a $\mathsf{coassocl}$ a equação,

where $\mathsf{coassocr} = [id + i_1, i_2 \cdot i_2]$. *Find its converse* $\mathsf{coassocl}$ *by solving the equation,*

$$\mathsf{coassocl} \cdot \mathsf{coassocr} \quad = \quad id$$

isto é, a equação

that is, the equation

$$\mathsf{coassocl} \cdot \underbrace{[id + i_1, i_2 \cdot i_2]}_{\mathsf{coassocr}} = id$$

Finalmente, exprima $\mathsf{coassocl}$ sob a forma de um programa em Haskell *não recorra* ao combinador "either".

Finally express $\mathsf{coassocl}$ *in pointwise Haskell code not using the "either"combinator.*

---

8. A *lei da troca* (identifique-a no formulário) permite-nos exprimir determinadas funções sob duas formas alternativas, conforme desenhado no respectivo diagrama:

The exchange law *(check this in the reference sheet) allows one to express certain functions in two alternative forms, as pictured in the following diagram:*

$$[\langle f, g\rangle, \langle h, k\rangle] = \langle [f, h], [g, k]\rangle$$

$$\begin{tikzcd}
A \arrow[r, "i_1"] \arrow[d, "f"'] \arrow[drr, "g"] & A + B & B \arrow[l, "i_2"'] \arrow[d, "k"] \arrow[dll, "h"'] \\
C & C \times D \arrow[l, "\pi_1"] \arrow[r, "\pi_2"'] & D
\end{tikzcd} \tag{F6}$$

Demonstre esta lei recorrendo às propriedades (e.g. universais) dos produtos e dos co-produtos.

Prove this law using the (e.g. universal) properties of products and co-products.

---

9. Use a lei da troca (F6) para exprimir o isomorfismo

Use (F6) to express the isomorphism

$$\mathsf{undistl} = [i_1 \times id, i_2 \times id]$$

sob a forma de um 'split' de alternativas.

in the form of a 'split' of alternatives.
