# Cálculo de Programas
## *Algebra of Programming*

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

2024/25 - Ficha nr.° 5

1. No cálculo de programas, as definições condicionais do tipo

*Conditional expressions of pattern*

$$h\ x = \textbf{if}\ p\ x\ \textbf{then}\ f\ x\ \textbf{else}\ g\ x \tag{F1}$$

são escritas usando o combinador ternário

*are expressed in the algebra of programming by the ternary combinator*

$$p \rightarrow f, g$$

conhecido pelo nome de *condicional de McCarthy*, cuja definição

*known as* the McCarthy conditionald*, whose definition*

$$p \rightarrow f, g \quad = \quad [f\ , g] \cdot p? \tag{F2}$$

vem no formulário. Baseie-se em leis desse formulário para demonstrar a chamada 2ª-lei de fusão do condicional:

*can be found in reference sheet. Use this reference sheet to prove the so-called 2nd fusion-law of conditionals:*

$$(p \rightarrow f, g) \cdot h \quad = \quad (p \cdot h) \rightarrow (f \cdot h), (g \cdot h)$$

---

2. Numa máquina paralela pode fazer sentido, em (F1), não esperar por $p\ x$ para avaliar ou $f\ x$ ou $g\ x$, mas sim correr tudo em paralelo,

*On a parallel machine it might make sense, concerning (F1), not to wait for $p\ x$ to evaluate either $f\ x$ or $g\ x$, but rather to run everything in parallel,*

$$parallel\ p\ f\ g = \langle\langle f, g\rangle, p\rangle$$

e depois fazer a escolha do resultado:

*and the choose the outcome:*

$$choose = \pi_2 \rightarrow \pi_1 \cdot \pi_1, \pi_2 \cdot \pi_1$$

Mostre que, de facto:

*Show that, indeed:*

$$choose \cdot parallel\ p\ f\ g \quad = \quad p \rightarrow f, g$$

---

3. Sabendo que as igualdades — *Assuming*

$$p \to k, k \;=\; k \tag{F3}$$
$$(p? + p?) \cdot p? \;=\; (i_1 + i_2) \cdot p? \tag{F4}$$

se verificam, demonstre as seguintes propriedades do mesmo combinador: — *prove the following laws of the McCarthy conditional:*

$$\langle (p \to f, h), (p \to g, i) \rangle \;=\; p \to \langle f, g \rangle, \langle h, i \rangle \tag{F5}$$
$$\langle f, (p \to g, h) \rangle \;=\; p \to \langle f, g \rangle, \langle f, h \rangle \tag{F6}$$
$$p \to (p \to a, b), (p \to c, d) \;=\; p \to a, d \tag{F7}$$

---

4. Mostre que a propriedade de cancelamento da exponenciação — *Show that the cancellation property*

$$\mathsf{ap} \cdot (\overline{f} \times id) = f \tag{F8}$$

corresponde à definição — *is nothing but the definition*

$$\mathsf{curry}\, f \;\; a \;\; b = f \;\; (a, b)$$

quando se escreve $\mathsf{curry}\, f$ em lugar de $\overline{f}$. — *once* $\mathsf{curry}\, f$ *is written instead of* $\overline{f}$.

---

5. Mostre que a definição de $\mathsf{uncurry}$ se pode obter também de (F8) fazendo $f := \mathsf{uncurry}\;\; g$, introduzindo variáveis e simplificando. — *Show that the definition of* $\mathsf{uncurry}$ *can also be obtained from (F8) by instantiating* $f := \mathsf{uncurry}\;\; g$*, introducing variables and simplifying.*

---

6. Prove a igualdade — *Prove the equality*

$$\overline{f \cdot (g \times h)} \;=\; \overline{\mathsf{ap} \cdot (id \times h)} \cdot \overline{f} \cdot g \tag{F9}$$

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

---

7. É dada a definição — *Let* $\mathsf{flip}$ *be defined by*

$$\mathsf{flip}\, f = \overline{\widehat{f} \cdot \mathsf{swap}} \tag{F10}$$

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 $\mathsf{flip}$ é um isomorfismo por ser a sua própria inversa: — *Show that it is an isomorphism because it is its own inverse:*

$$\mathsf{flip}\; (\mathsf{flip}\, f) = f \tag{F11}$$

Mostre ainda que:

*Furthermore show:*

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

---

8. Mostre que

*Show that*

$$junc \cdot unjunc = id \tag{F12}$$
$$unjunc \cdot junc = id \tag{F13}$$

se verificam, onde

*hold for*

$$\begin{tikzcd} A^{B+C} \arrow[rr, bend left, "unjunc"] \arrow[rr, phantom, "\cong"] & & A^B \times A^C \arrow[ll, bend left, "junc"] \end{tikzcd} \qquad \left\{ \begin{array}{l} junc\ (f, g) = [f\ , g] \\ unjunc\ k = (k \cdot i_1, k \cdot i_2) \end{array} \right. \tag{F14}$$

---

9. Considere a seguinte sintaxe concreta em Haskell para um tipo que descreve pontos no espaço tridimensional:

*Consider the following concrete syntax in Haskell for a type that describes 3D-points:*

$$\mathbf{data}\ Point\ a = Point\ \{x :: a, y :: a, z :: a\}\ \mathbf{deriving}\ (Eq, Show)$$

Pelo GHCi apura-se:

*GHCi tells:*

$$Point :: a \rightarrow a \rightarrow a \rightarrow Point\ a$$

Raciocinando apenas em termos de tipos, conjecture a definição de in na seguinte conversão dessa sintaxe concreta para abstracta:

*Reasoning only in terms of types, conjecture the definition of* in *in the following conversion from concrete to abstract syntax:*

$$\begin{tikzcd} Point\ A \arrow[rr, bend left, "\mathsf{out} = \langle\langle x, y\rangle, z\rangle"] \arrow[rr, phantom, "\cong"] & & (A \times A) \times A \arrow[ll, bend left, "\mathsf{in} = \cdots"] \end{tikzcd}$$

---

10. **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**: The solution given for a previous problem,*

$$store\ c = \mathsf{take}\ 10 \cdot nub \cdot (c{:}) \tag{F15}$$

*calls the standard function*

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

*available from the Data.List library in Haskell.*

*After inspecting the standard implementation of this function, define* $f$ *so that*

$$nub = [\mathsf{nil}\ , \mathsf{cons}] \cdot f.$$

*is an alternative to the standard definition, where* $\mathsf{nil}\ \_ = [\,]$ *and* $\mathsf{cons}\ (h, t) = h : t$. *Check that* $store\ c$ *(F15) works properly once the standard* $nub$ *is replaced by yours.*

***Important****: Structure your solution across the* $f \cdot g$, $\langle f, g \rangle$, $f \times g$, $[f\ , g]$ *and* $f + g$ *combinators available from library Cp.hs. Use* ***diagrams*** *to plan your solution, in which you should avoid re-inventing functions over lists already available in the Haskell standard libraries.*

□