

Cálculo de Programas

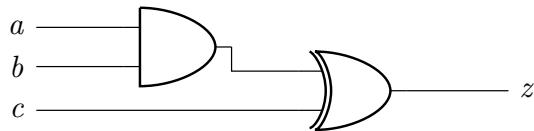
Algebra of Programming

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

2024/25 - Ficha nr.º 2

1. O circuito booleano

The following Boolean circuit



pode descrever-se pela função f que se segue,

can be described by the following function f

$$\begin{cases} f : (\mathbb{B} \times \mathbb{B}) \times \mathbb{B} \rightarrow \mathbb{B} \\ f = xor \cdot (and \times id) \end{cases} \quad (F1)$$

onde $and(a, b) = a \wedge b$ e $xor(x, y) = x \oplus y$.

where $and(a, b) = a \wedge b$ and $xor(x, y) = x \oplus y$.

(a) Mostre que f se pode também definir como se segue:

(a) Show that f can also be defined as follows:

$$f((a, b), c) = (a \wedge b) \oplus c$$

(b) Qual é o tipo da função $g = \langle \pi_1, f \rangle$?

(b) What is the type of function $g = \langle \pi_1, f \rangle$?

2. Implemente e teste f (F1) no GHCI, após carregar a biblioteca *Cp.hs* disponível no material pedagógico.

NB: recorde $x \oplus y = x \neq y$.

*Encode and test f (F1) on GHCI, after loading the library *Cp.hs* available from the course website.*

NB: recall $x \oplus y = x \neq y$.

3. Defina no GHCI o seguinte tipo de dados:

Define the following data type on GHCI:

```
data X = B Bool | P (Bool, Int)
```

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 ?

NB: em Haskell a alternativa $[f, g]$ escreve-se either f g .

*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: $[f, g]$ is written either f g in Haskell.

4. O combinador $\langle f, g \rangle$ — isto é, “ f em paralelo com g ” — satisfaz a seguinte propriedade, dita **universal**:

$$k = \langle f, g \rangle \equiv \begin{cases} \pi_1 \cdot k = f \\ \pi_2 \cdot k = g \end{cases} \quad (\text{F2})$$

Identifique-a no formulário. Que outra propriedade desse formulário obtém fazendo $k = id$ e simplificando?

The $\langle f, g \rangle$ combinator — that is, “ f in parallel with g ” — satisfies the following property, called **universal**:

Find it in the reference sheet. Which other property of this form do you get by making $k = id$ and simplifying?

5. Derive a partir de (F2) a lei

From (F2) derive the law

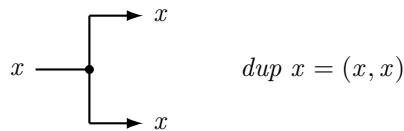
$$\langle h, k \rangle \cdot f = \langle h \cdot f, k \cdot f \rangle$$

que também consta desse formulário sob a designação *fusão-* \times .

that also appears in the reference sheet under the name \times -fusion.

6. Uma das operações essenciais em processamento da informação é a sua *duplicação*:

A core operation in information processing is data duplication:



Recorra à lei de fusão- \times para demonstrar a seguinte propriedade da duplicação de informação:

Derive from the \times -fusion law the following property of data duplication:

$$dup \cdot f = \langle f, f \rangle$$

7. Considere a função:

Consider function *xr* defined as follows:

$$xr = \langle \pi_1 \times id, \pi_2 \cdot \pi_1 \rangle \quad (\text{F3})$$

Mostre que *xr* satisfaz a propriedade

Show that property

$$xr \cdot \langle \langle f, g \rangle, h \rangle = \langle \langle f, h \rangle, g \rangle \quad (\text{F4})$$

para todo o f, g e h .

holds for all f, g and h .

8. O combinador

The following combinator

$$\begin{aligned} \text{const} &:: a \rightarrow b \rightarrow a \\ \text{const } a\ b &= a \end{aligned}$$

está disponível em Haskell para construir funções constantes, sendo habitual designarmos *const k* por *k*. Demonstre a igualdade

is available in Haskell to build constant functions. (One usually abbreviates *const k* to *k*.) Prove the equality

$$\underline{(b, a)} = \langle \underline{b}, \underline{a} \rangle \quad (\text{F5})$$

a partir da propriedade universal do produto e das propriedades das funções constantes que constam do formulário.

based on the \times -universal property and the properties of constant functions that appear in the reference sheet.

9. Determine o tipo da função α que se segue:

Infer the most general type of the following function α :

$$\alpha = [\langle \text{FALSE}, \text{id} \rangle, \langle \text{TRUE}, \text{id} \rangle] \quad (\text{F6})$$

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-cp** 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, if so, publish their solutions in the **#geral-cp** Slack channel, so as to trigger discussion among other colleagues. The requirements of the problem are given below.

Problem requirements: Given a name, for instance "Jose Nuno Oliveira" we wish to obtain its acronym and its short version, as suggested below:

```
*Cp> acronym "Jose Nuno Oliveira"
"JNO"
*Cp> short "Jose Nuno Oliveira"
"Jose Oliveira"
*Cp>
```

Define

```
acronym = ...
short = ...
```

subject to the following restrictions:

- you cannot use argument variables (x, y, \dots)
- you can use function composition $f \cdot g$ and the parallel combinator $\langle f, g \rangle$ as well as any function available from module `Cp.hs`
- you can resort to Haskell standard functions such as e.g. `map`, `filter` and so on.