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. 6

1. 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:

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

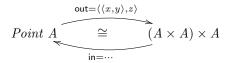
Pelo GHCi apura-se:

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:



2. Mostre que a propriedade de cancelamento da exponenciação

Show that the cancellation property

$$ap \cdot (\overline{f} \times id) = f \tag{F1}$$

corresponde à definição

is nothing but the definition

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

quando se escreve curry f em lugar de \overline{f} .

once curry f is written instead of \overline{f} .

3. Prove a igualdade

Prove the equality

$$\overline{f \cdot (g \times h)} = \overline{\mathsf{ap} \cdot (id \times h)} \cdot \overline{f} \cdot g \tag{F2}$$

usando as leis das exponenciais e dos produtos.

using the laws of products and exponentials.

4. Mostre que

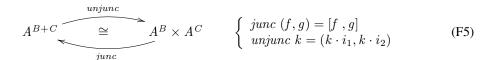
Show that

$$junc \cdot unjunc = id$$
 (F3)

$$unjunc \cdot junc = id$$
 (F4)

se verificam, onde

hold for



5. É dada a definição

Let flip be defined by

$$\mathsf{flip}\,f = \overline{\widehat{f} \cdot \mathsf{swap}} \tag{F6}$$

de acordo com:

according to:

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

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

$$\mathsf{flip}\,(\mathsf{flip}\,f) = f \tag{F7}$$

Mostre ainda que:

Furthermore show:

flip
$$f x y = f y x$$
.

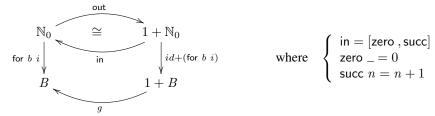
6. O código que se segue, escrito em Haskell, implementa a noção de ciclo-for , onde $\it b$ é o corpo ("body") do ciclo e i é a sua inicialização:

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

$$\begin{cases} \text{ for } b \ i \ 0 = i \\ \text{ for } b \ i \ (n+1) = b \ (\text{for } b \ i \ n) \end{cases}$$
 (F8)

Mostre que for b i = (g) para um dado g(descubra qual) de acordo com o diagrama seguinte:

Show that for b i = (g) for some g (find which) according to the following diagram:



$$\text{where} \quad \left\{ \begin{array}{l} \text{in} = [\mathsf{zero} \ , \mathsf{succ} \ \\ \mathsf{zero} \ _ = 0 \\ \mathsf{succ} \ n = n + 1 \end{array} \right.$$

- 7. Mostre que (a+) dada a seguir é o ciclo for b i para um dado b e um dado i descubra quais:
- Show that (a+) given next is a for-loop for b i for b and i to be calculated:

$$\begin{cases} a+0 = a \\ a+(n+1) = 1 + (a+n) \end{cases}$$
 (F9)

8. Apresente justificações para a seguinte dedução da lei de fusão-cata válida para ciclos-for:

Justify the following calculation of the catafusiona law valid for for-loops:

Em suma:

Summing up:

$$f \cdot (q) = (h) \iff f \cdot q = h \cdot (id + f) \tag{F10}$$

9. Sabendo que for f $i = ([\underline{i}, f])$ — que é o que deve ter apurado na questão 6 —, recorra à lei de fusão-cata para demonstrar a propriedade:

Knowing that for $f i = ([\underline{i}, f])$ — which is what you might have found in question 6 —, use the law of cata-fusion to prove the property:

$$f \cdot (\text{for } f \ i) = \text{for } f \ (f \ i) \tag{F11}$$

10. Mostre que as funções

Show that functions

$$f = \text{for } id \ i$$
$$g = \text{for } \underline{i} \ i$$

são a mesma função. (Qual?)

are the same function. (Which?)