

Resolução dos exercícios propostos - Ficha 2

Ex. 1.

(a) Demonstração da Lei da Reflexão - + :

$$[i_1, i_2] = id$$

Método 1:

Como observamos, na lei acima, temos um "either" igual a "qualquer coisa" (neste caso, a função id). Então, aplicando a propriedade do com binador do enunciado temos:

$$\begin{aligned} [i_1, i_2] = id &\Leftrightarrow \{ \text{universal} - + \} \\ \Leftrightarrow \begin{cases} id \cdot i_1 = i_1 \\ id \cdot i_2 = i_2 \end{cases} &\Leftrightarrow \{ \text{natural} - id \} \end{aligned}$$

$$\Leftrightarrow \begin{cases} i_1 = i_1 \\ i_2 = i_2 \end{cases} \Leftrightarrow \{ \text{Prop reflexiva da igualdade} \}$$

$$\Leftrightarrow \begin{cases} \text{True} \\ \text{True} \end{cases} \Leftrightarrow \text{True}$$

Método 2:

Aplicar o mesmo método, mas utilizando funções genéricas f e g :
Seja $K = id$. Assim, temos:

$$id = [f, g] \Leftrightarrow \{ \text{universal} - + \}$$

$$\Leftrightarrow \begin{cases} id \cdot i_1 = f \\ id \cdot i_2 = g \end{cases} \Leftrightarrow \{ \text{natural} - id \}$$

$$\Leftrightarrow \begin{cases} i_1 = f \\ i_2 = g \end{cases} \Leftrightarrow id = [f, g] = [i_1, i_2]$$

Para os próximos exercícios, usa-se o método 1, que será mais intuitivo.

(b)

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

Note-se que estamos a querer definir o split à custa de uma composta!

$$\underbrace{\langle h, k \rangle \cdot f}_k = \langle h \cdot f, k \cdot f \rangle \Leftrightarrow \{ \text{universal} - x \}$$

$$\Leftrightarrow \begin{cases} \pi_1 \cdot (\langle h, k \rangle \cdot f) = h \cdot f \\ \pi_2 \cdot (\langle h, k \rangle \cdot f) = k \cdot f \end{cases} \Leftrightarrow \{ \text{Prop. assoc. comp} \}$$

$$\Leftrightarrow \begin{cases} (\pi_1 \cdot \langle h, k \rangle) \cdot f = h \cdot f \\ (\pi_2 \cdot \langle h, k \rangle) \cdot f = k \cdot f \end{cases} \Leftrightarrow \{ \text{cancelamento} - x \}$$

$$\Leftrightarrow \begin{cases} h \cdot f = h \cdot f \\ k \cdot f = k \cdot f \end{cases} \Leftrightarrow \{ \text{Prop. reflexiva da igualdade} \}$$

$$\Leftrightarrow \begin{cases} \text{True} \\ \text{True} \end{cases} \Leftrightarrow \text{True.}$$

Alguns aspetos sobre este exercício:

1. em (a) queríamos definir a função $[i_1, i_2]$ à custa de um "k", que no nosso caso era a função id , daí fazer sentido recorrer à lei universal.
2. em (b) queríamos definir o combinador $\langle h \cdot f, k \cdot f \rangle$ à custa de um "k", que no nosso caso era a função composta $\langle h, k \rangle \cdot f$, daí fazer sentido recorrer de igual modo à lei universal desse mesmo combinador.

Ex. 2

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

Na verdade, sabemos que se um certo tipo for "submetido" a uma função constante, então, ele perde toda a sua informação e passa apenas a ser julgado o output dessa mesma função.

$$[\underline{k}, \underline{k}] = \underline{k} \Leftrightarrow \{ \text{universal} - + \}$$

$$\Leftrightarrow \begin{cases} \underline{k} \cdot i_1 = \underline{k} \\ \underline{k} \cdot i_2 = \underline{k} \end{cases} \Leftrightarrow \{ \text{natural} - \text{const} \} \text{ ou } \{ \text{def. } \underline{k} \cdot f = \underline{k} \}$$

$$\Leftrightarrow \begin{cases} \underline{k} = \underline{k} \\ \underline{k} = \underline{k} \end{cases} \Leftrightarrow \{ \text{Prop. reflexiva da igualdade} \} \begin{cases} \text{True} \\ \text{True} \end{cases} \Leftrightarrow \text{True.}$$

Ex. 3

$$i) \quad id + id = id \quad \{ \text{functor} - id - + \}$$

$$id + id = \{ \text{def. } f + g = [i_1 \cdot f, i_2 \cdot g] \} \text{ ou } \{ \text{Def. } + \}$$

$$= [i_1 \cdot id, i_2 \cdot id] = \{ \text{natural} - id \}$$

$$= [i_1, i_2]$$

Então, temos $id + id = [i_1, i_2]$, MAS É $[i_1, i_2] = id$?

$$\textcircled{2} \quad \begin{array}{ccc} \text{universal} & \text{natural-id} & \text{reflexiva} \\ [i_1, i_2] = id \Leftrightarrow \begin{cases} id \cdot i_1 = i_1 \\ id \cdot i_2 = i_2 \end{cases} \Leftrightarrow \begin{cases} i_1 = i_1 \\ i_2 = i_2 \end{cases} \Leftrightarrow \text{TRUE} \end{array}$$

$$\text{Logo, } id + id = [i_1, i_2] = id \Leftrightarrow id + id = id.$$

$$ii) \quad (f + g) \cdot i_1 = i_1 \cdot f \quad \{ \text{Natural} - i_1 \}$$

$$\Leftrightarrow (f + g) \cdot i_1 = i_1 \cdot f \Leftrightarrow \{ \text{Def. } + \}$$

$$\Leftrightarrow [i_1 \cdot f, i_2 \cdot g] \cdot i_1 = i_1 \cdot f \Leftrightarrow \{ \text{Cancelamento} - + \}$$

$$\Leftrightarrow i_1 \cdot f = i_1 \cdot f \Leftrightarrow \text{TRUE}$$

$$iii) \quad (f + g) \cdot i_2 = i_2 \cdot g \quad \{ \text{Natural} - i_2 \}$$

$$(f + g) \cdot i_2 = i_2 \cdot g \Leftrightarrow \{ \text{Def. } + \}$$

$$\Leftrightarrow [i_1 \cdot f, i_2 \cdot g] \cdot i_2 = i_2 \cdot g \Leftrightarrow \{ \text{Cancelamento} - + \}$$

$$\Leftrightarrow i_2 \cdot g = i_2 \cdot g \Leftrightarrow \text{TRUE}$$

Notas sobre este exercício:

1. No passo ②, poderíamos ter aplicado diretamente a prop. reflexiva - +.

Assim:

$$[i_1, i_2] = id \quad \{ \text{reflexiva} - + \}$$

Obs. Estas demonstrações são, naturalmente, suportadas por diagramas.

Ex. 4

$$\text{coswap} = [i_2, i_1]$$

Objetos

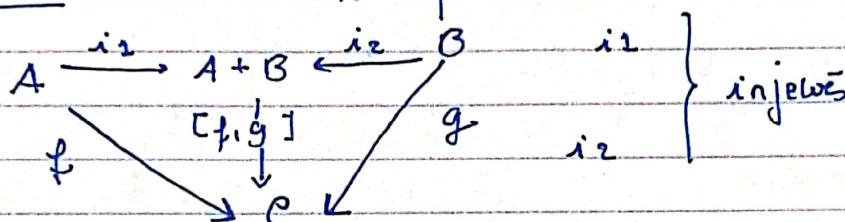
1) Diagrama de Coswap;

2) Demonstrar: $\text{coswap} \cdot \text{coswap} = \text{id}$

Revisar do Coproduto!

- Formado por funções que partilham o códomínio;
- Os elementos do coproduto são de A ou B, tendo uma identificação;
- $A+B = \{ \underbrace{i_1 a \mid a \in A}_{(i_1, a)} \cup \{ \underbrace{i_2 b \mid b \in B}_{(i_2, b)} \}$

Diagrama: sempre que há um coproduto/alternativa, devem sair duas setas que entram num mesmo tipo:



onde:

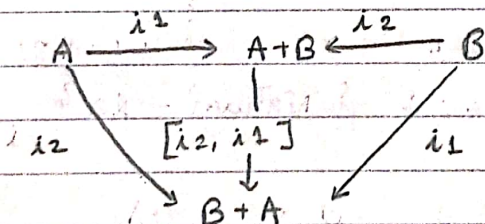
- i_1 : pega num elemento de um conjunto e injeta-o num $A+B$
- i_2 : pega num elemento de um conjunto e injeta-o num $A+B$

REAS: i_1 tem "etiqueta" A, logo: $i_1 : A \rightarrow A+B$
 i_2 tem "etiqueta" B, logo: $i_2 : B \rightarrow A+B$

1)

$$i_2 : B \rightarrow A+B$$

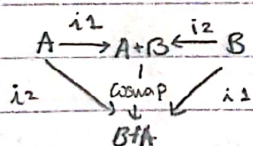
$$i_1 : A \rightarrow A+B$$



2) Demonstrar q diagrama que $\text{coswap} \cdot \text{coswap} = \text{id}$

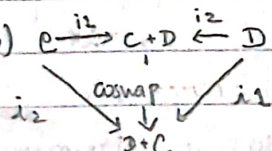
diagrama a) diagrama b)

a)

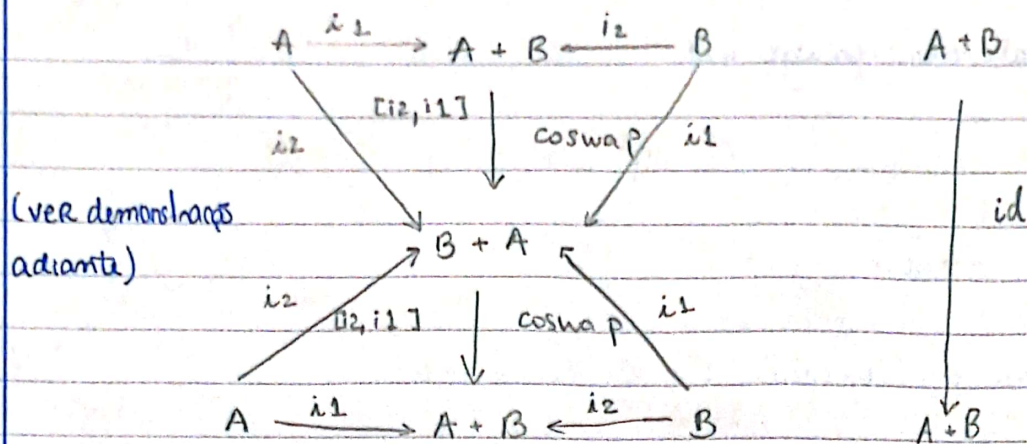


obs: de a) para b) houve apenas mudança de tipos!

b)



$\text{coswap} \cdot \text{coswap} : A+B \rightarrow B+A$, mas como as duas
 $C+D \rightarrow D+C$ funções são coswap:
 $B+A = C+D \Rightarrow \text{coswap} \cdot \text{coswap} : A+B \rightarrow B+A \rightarrow A+B$
 Assim, temos o seguinte diagrama genérico:



Então, note-se que podemos seguir o caminho de $A+B$ para $A+B$, tendo-se $\text{coswap} \cdot \text{coswap}$, mantendo-se o tipo de entrada e saída, logo, estamos perante a função identidade, provando-se: $\text{coswap} \cdot \text{coswap} = \text{id}$.

Ex.5

$$d = [\langle \text{false}, \text{id} \rangle, \langle \text{true}, \text{id} \rangle]$$

$$\langle \text{false}, \text{id} \rangle : A \longrightarrow A \times B$$

$$\langle \text{true}, \text{id} \rangle : C \longrightarrow C \times D$$

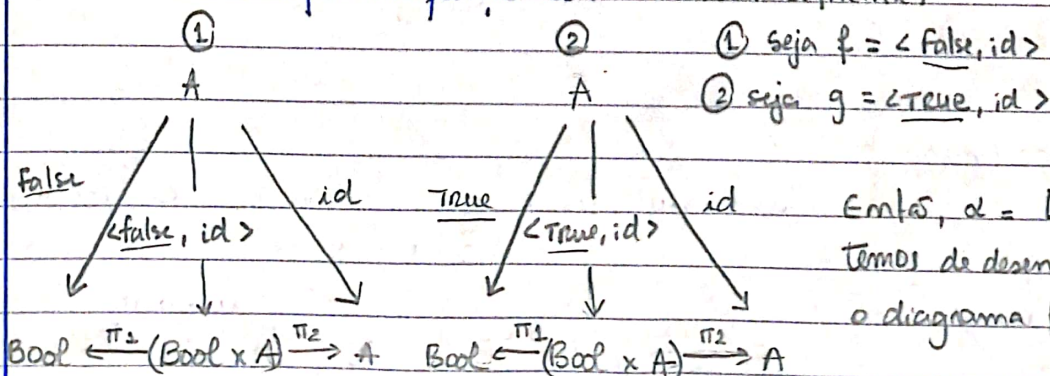
$$d = [\langle \text{false}, \text{id} \rangle, \langle \text{true}, \text{id} \rangle] \quad (\equiv) \quad \{ \text{universal} - + \}$$

$$(\equiv) \quad \begin{cases} d \cdot i_1 = \langle \text{false}, \text{id} \rangle \\ d \cdot i_2 = \langle \text{true}, \text{id} \rangle \end{cases} \quad (\equiv) \quad \begin{cases} d \cdot i_1 \cdot \pi_1 = \langle \text{false}, \text{id} \rangle \cdot \pi_1 \\ d \cdot i_2 \cdot \pi_1 = \langle \text{true}, \text{id} \rangle \cdot \pi_1 \end{cases} \quad (\equiv) \quad \{ \text{Def "split"} \}$$

$$(\equiv) \quad \begin{cases} d \cdot i_1 \cdot \pi_1 = (\text{false} \cdot \pi_1, \text{id} \cdot \pi_1) \\ d \cdot i_2 \cdot \pi_1 = (\text{true} \cdot \pi_1, \text{id} \cdot \pi_1) \end{cases} \quad (\equiv) \quad \begin{cases} d(i_1 \cdot \pi_1) = (\text{false} \cdot \pi_1, \text{id} \cdot \pi_1) \\ d(i_2 \cdot \pi_1) = (\text{true} \cdot \pi_1, \text{id} \cdot \pi_1) \end{cases} \quad (\equiv) \quad \{ \text{Def-comp} \}$$

$$(\equiv) \quad \begin{cases} d(i_1 \cdot \pi_1) = (\text{false}, \pi_1) \\ d(i_2 \cdot \pi_1) = (\text{true}, \pi_1) \end{cases} \quad \text{c.g. } d \quad \begin{cases} \{ \text{Def-id} \} \\ \{ \text{Def-const} \} \end{cases}$$

Determinar o tipo da função d : (continua a frente)



Então, $d = [f, g]$,
 temos de desenhar
 o diagrama final

Demonstrasi $\cos \omega \cdot \cos \omega = \text{id} :$

$$\text{Coswap} \cdot \text{Coswap} = \text{Id} \quad \Rightarrow \quad \{ \text{Id}, \text{Coswap} \}$$

$$= \text{swap} \cdot [12, 11] = \{ \text{FUSO} + \}$$

$$= [\coswap \cdot i2, \coswap \cdot i1] \} \text{ def } \coswap \}$$

$$= [[i_2, i_1] \cdot i_2, [i_2, i_1] \cdot i_1] \quad \{ \text{cancelamento} - + \}$$

$$= [i_1, i_2] = \{ \text{Reflexões} + \}$$

$$= id$$

Ex. 6

Temha-se a definição de fatorial:

$$\begin{cases} \text{fac } 0 = 1 \\ \text{fac } (m+1) = (m+1) * \text{fac } m \end{cases}$$

Preende-se mostrar a equação:

$$fac. [0, succ] = [1, mul. < succ, fac >]$$

where

$$\text{succ } m = m+1$$

$$\text{mul}(a, b) = a * b$$

A estratégia mais simples é começar pela igualdade sem variáveis e chegar a outra com variáveis (com zero = 0 e one = 1).

$$fac \cdot [0, succ] = [\perp, mul \cdot \langle succ, fac \rangle] \quad (\equiv) \quad \{ fus\tilde{a} - + ?$$

$$(\Rightarrow) [\text{fac} \cdot 0, \text{fac} \cdot \text{succ}] = [1, \text{mul} \cdot \langle \text{succ}, \text{fac} \rangle] \Leftrightarrow \{E_7 - +\}$$

$$\Rightarrow \begin{cases} fac \cdot 0 = \underline{1} \\ fac \cdot succ = mul \cdot \langle succ, fac \rangle \end{cases} \quad \Rightarrow \begin{cases} fac \cdot zero = one \\ fac \cdot succ = mul \cdot \langle succ, fac \rangle \end{cases} \quad \Rightarrow$$

$$(E) \begin{cases} \text{fac } \text{zero } x = \text{one} \\ \text{fac} \cdot \text{succ } x = \text{mul} \cdot \langle \text{succ}, \text{fac} \rangle x \end{cases} \quad (E') \begin{cases} \text{fac } \text{zero} = \text{one} \\ \text{fac} \cdot \text{succ } x = \text{mul} \cdot \langle \text{succ}, \text{fac} \rangle x \end{cases}$$

$\{ \text{Def-comp} \}$

$$(\Rightarrow) \int f_{AC}(z) dz = \text{one}$$

$$\begin{aligned} \text{② } \left\{ \begin{array}{l} \text{fac}(zero) = one \\ \text{fac}(succ\ x) = mul(\langle succ, fac \rangle x) \end{array} \right. & \quad \text{③ } \left\{ \begin{array}{l} \text{fac}\ zero = one \\ \text{fac}(succ\ x) = mul(succ\ x, fac\ x) \end{array} \right. \quad \text{④} \end{aligned}$$

{ def split }

fac zero = one

$$\text{fac}(\text{succ } n) = \text{mul}(\text{succ } n, \text{fac } n)$$

$$[zero = 0 \wedge one = 1]$$

{ Def-succ \wedge Def-mult }

$$\Leftrightarrow \begin{cases} \text{fac } 0 = 1 \\ \text{fac } (n+1) = \text{mul } (n+1, \text{fac } n) \end{cases}$$

$$\text{fac } (n+1) = \text{mul } (n+1, \text{fac } n)$$

{ Def-mult }

$$\Leftrightarrow \begin{cases} \text{fac } 0 = 1 \\ \text{fac } (n+1) = n+1 \cdot \text{fac } n \end{cases}$$

$$\text{fac } (n+1) = n+1 \cdot \text{fac } n$$

Notas sobre o exercício:

1. A estratégia é pegar sempre na notação point-free e chegar à notação point-wise.
2. A lei universal - + e x das muito eficazes mask-tips de demonstrações.
3. Pode ajudar inferir os tipos.

Ex. 7

data LTree a = Leaf a | Fork (LTree a, LTree a)

in = [Leaf, Fork]

out = ?

out · in = id

$$\text{out} \cdot \text{in} = \text{id} \Leftrightarrow \{ \text{Def-in} \}$$

$$\Leftrightarrow \text{out} \cdot [\text{Leaf}, \text{Fork}] = \text{id} \Leftrightarrow \{ \text{Fusion} - + \}$$

$$\Leftrightarrow [\text{out} \cdot \text{Leaf}, \text{out} \cdot \text{Fork}] = \text{id} \Leftrightarrow \{ \text{Universal} - + \}$$

$$\Leftrightarrow \begin{cases} \text{id} \cdot i_1 = \text{out} \cdot \text{Leaf} \\ \text{id} \cdot i_2 = \text{out} \cdot \text{Fork} \end{cases} \Leftrightarrow \{ \text{natural-id} \}$$

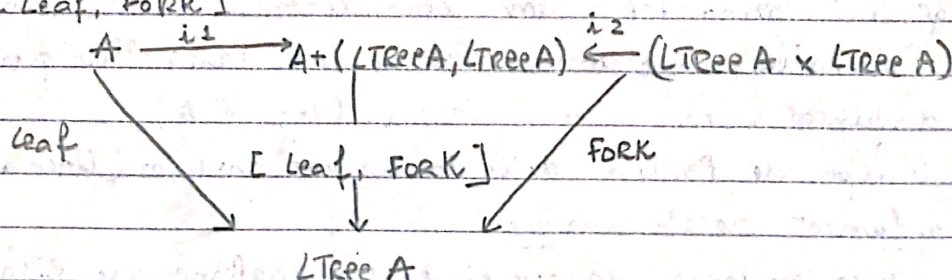
$$\Leftrightarrow \begin{cases} i_1 = \text{out} \cdot \text{Leaf} \\ i_2 = \text{out} \cdot \text{Fork} \end{cases} \Leftrightarrow \{ \text{assoc-comp?} \} \text{ e } \{ \text{Equality Extensional?} \}$$

{ Def-out }

$$\Leftrightarrow \begin{cases} i_1 a = \text{out} (\text{Leaf } a) \\ i_2 (e, d) = \text{out} (\text{Fork } (e, d)) \end{cases} = \text{out}$$

Diagrama que explica a função in:

in = [Leaf, Fork]



Ver observações do ex no verso!

Obs. [Ex. 5]

1. False e True representam funções constantes, já False e True (sem sublinhado) correspondem a elementos do tipo booleano;
2. Como temos "Splits" utilizamos projeções π_1 e π_2 em vez de injeções;
3. f e g , sendo funções de um "either", $\alpha = [f, g]$, partilham o mesmo codomínio; logo, no diagrama pode usar as mesmas letras;
4. Concluímos que o domínio de α é $A + A$, pois os domínios são: A e A ;
 pois se α é um "either"

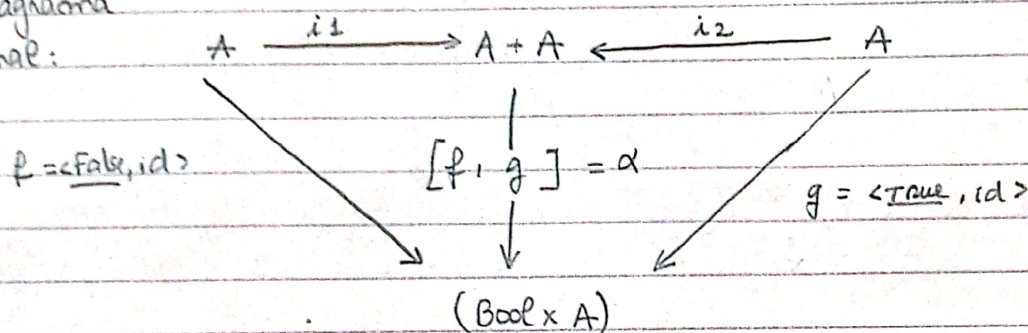
Assum:

$$f: A \rightarrow \text{Bool} \times A$$

$$g: A \rightarrow \text{Bool} \times A$$

$$\alpha = [f, g] \text{ e } \alpha: A + A \rightarrow \text{Bool} \times A$$

Diagrama
Final:



Obs. [Ex. 7]

1. Desde logo, sabemos que, sendo in a inversa de out, o seu domínio será o codomínio de out e o seu codomínio será o domínio de out. Logo: $\text{in}: A + (\text{LTree } A \times \text{LTree } A) \rightarrow \text{LTree } A$;
2. Para desenhar o diagrama:
 - 2.1. Começa-se pelas setas que entram no mesmo tipo, pois in é definido por um "either";
 - 2.2. podemos escolher a função Leaf à esquerda e fork à direita já que $\text{in} = [\text{leaf}, \text{fork}]$;
 - 2.3. o tipo para o qual as setas apontam é LTree A pois tanto a função leaf como fork têm como codomínio esse tipo.
 - 2.4. Para saber o tipo do domínio de leaf basta olhar para a definição da função: $\text{leaf} :: a \rightarrow \text{LTree } a$. (logo é A);
 - 2.5. O tipo de fork é definido por um produto $(\text{LTree } a \times \text{LTree } a)$, pois a função recebe um par!
 - 2.6. Tendo em conta as projeções e injeções, sabemos que either usa injeções: logo da esquerda vem a projeção i_1 e da direita i_2 , obtendo o tipo final do domínio: $A + (\text{LTree } A \times \text{LTree } A)$.