Cálculo de Programas

2.º Ano de LCC+MiEI (Universidade do Minho)
Ano Lectivo de 2019/20

Exame de Recurso — 18 de Julho de 2020
10h00–13h00
Prova realizada on-line via BBC

- *Esta prova consta de **10** questões que valem, cada uma, 2.0 valores. O tempo médio estimado para resolução de cada questão é de 15 min.*
- *Ao submeterem o seu teste no BB os alunos estão a declarar implicitamente que **subscrevem** a seguinte declaração:*

  Tendo presente o Código de Conduta Ética da Universidade do Minho e o Regulamento Disciplinar dos Estudantes da Universidade do Minho (Despacho RT-80/2019) e tendo conhecimento das sanções aplicáveis a atos de infração disciplinar, declaro por minha honra que pautarei a minha conduta na resolução desta prova de avaliação pelos valores de ética e integridade académica vigentes na Universidade do Minho. Declaro que realizarei a prova autonomamente e recorrendo exclusivamente aos elementos de consulta autorizada. Confirmo ainda que não incorrerei em qualquer ato de desonestidade, nomeadamente, os que violam os princípios éticos inerentes a processos de avaliação, como a prática de plágio ou qualquer outra forma indevida de utilização de informações. Mais declaro que não me envolverei em situações de prestação de informação ou apoio indevidos no decurso das provas que venha a realizar.
- *Os alunos devem escrever o seu **número mecanográfico** nas folhas cujas imagens vão submeter electronicamente.*

PROVA COM CONSULTA (3h)

Questão 1 Considere a função $\alpha = i_1 \cdot \mathsf{swap} \cdot \pi_2$. Infira o tipo mais geral de α e deduza a respectiva propriedade grátis, que deverá verificar **analiticamente**.

RESOLUÇÃO: O tipo obtém-se de imediato via `ghci Cp`:

```
i1.swap.p2 :: (c, (a, b)) -> Either (b, a) b
```

cuja propriedade grátis se obtém pelo diagrama do costume (omitido, fazer ao estudar):

$$i_1 \cdot \mathsf{swap} \cdot \pi_2 \cdot (h \times (f \times g)) = ((g \times f) + g) \cdot i_1 \cdot \mathsf{swap} \cdot \pi_2$$

Segue-se a verificação (completar com as justificações):

$$\begin{array}{ll}
& i_1 \cdot \mathsf{swap} \cdot \pi_2 \cdot (h \times (f \times g)) = ((g \times f) + g) \cdot i_1 \cdot \mathsf{swap} \cdot \pi_2 \\
\equiv & \{ \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \} \\
& i_1 \cdot \mathsf{swap} \cdot (f \times g) \cdot \pi_2 = (i_1 \cdot (g \times f)) \cdot \mathsf{swap} \cdot \pi_2 \\
\equiv & \{ \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \} \\
& i_1 \cdot (g \times f) \cdot \mathsf{swap} \cdot \pi_2 = i_1 \cdot (g \times f) \cdot \mathsf{swap} \cdot \pi_2 \\
\equiv & \{ \text{ trivial } \} \\
& \mathit{true}
\end{array}$$

□

□

Questão 2 Nas aulas desta disciplina foi definido e usado o isomorfismo

$$\mathsf{undistl} = [i_1 \times id, i_2 \times id] \tag{E1}$$

sem se ter definido logo o seu inverso $\mathsf{distl} : (A + B) \times C \to A \times C + B \times C$. Pretendemos mostrar que

$$\mathsf{distl} = \widehat{[\overline{i_1}, \overline{i_2}]} \tag{E2}$$

Complete a prova seguinte dessa definição, deduzida a partir de $\mathsf{distl} \cdot \mathsf{undistl} = id$:

$$\begin{array}{ll}
& \mathsf{distl} \cdot [i_1 \times id, i_2 \times id] = id \\
\equiv & \quad \{\ \boxed{1}\ \} \\
& \left\{ \begin{array}{l} \mathsf{distl} \cdot (i_1 \times id) = i_1 \\ \mathsf{distl} \cdot (i_2 \times id) = i_2 \end{array} \right. \\
\equiv & \quad \{\ \text{currying é um isomorfismo}\ \} \\
& \left\{ \begin{array}{l} \boxed{2} = \overline{i_1} \\ \overline{\mathsf{distl}\ (i_2 \times id)} = \boxed{3} \end{array} \right. \\
\equiv & \quad \{\ \text{fusão-exp}\ \} \\
& \left\{ \begin{array}{l} \overline{\mathsf{distl}} \cdot i_1 = \overline{i_1} \\ \boxed{4} \end{array} \right. \\
\equiv & \quad \{\ \boxed{5}\ \} \\
& \overline{\mathsf{distl}} = [\overline{i_1}, \overline{i_2}] \\
\equiv & \quad \{\ \boxed{6}\ \} \\
& \mathsf{distl} = \widehat{[\overline{i_1}, \overline{i_2}]} \\
\Box &
\end{array}$$

RESOLUÇÃO:

$$\begin{array}{ll}
& \mathsf{distl} \cdot [i_1 \times id, i_2 \times id] = id \\
\equiv & \quad \{\ \boxed{1} = \text{fusão-+ (20); universal-+ (17)}\ \} \\
& \left\{ \begin{array}{l} \mathsf{distl} \cdot (i_1 \times id) = i_1 \\ \mathsf{distl} \cdot (i_2 \times id) = i_2 \end{array} \right. \\
\equiv & \quad \{\ \text{currying}\ \} \\
& \left\{ \begin{array}{l} \underbrace{\overline{\mathsf{distl} \cdot (i_1 \times id)}}_{\boxed{2}} = \overline{i_1} \\ \overline{\mathsf{distl}\ (i_2 \times id)} = \underbrace{\overline{i_2}}_{\boxed{3}} \end{array} \right. \\
\equiv & \quad \{\ \text{fusão-exp}\ \} \\
& \left\{ \begin{array}{l} \overline{\mathsf{distl}} \cdot i_1 = \overline{i_1} \\ \underbrace{\overline{\mathsf{distl}} \cdot i_2 = \overline{i_2}}_{\boxed{4}} \end{array} \right.
\end{array}$$

$$\equiv \quad \{\ \boxed{5} = \text{universal-+ (17)}\ \}$$

$$\overline{\mathsf{distl}} = [\overline{i_1}, \overline{i_2}]$$

$$\equiv \quad \{\ \boxed{6} = \text{isomorfismo uncurry}\ \}$$

$$\mathsf{distl} = \widehat{[\overline{i_1}, \overline{i_2}]}$$

□

□

Questão 3 Considere a função $\mathsf{mirror} = (\!|\, \mathsf{in} \cdot (id + \mathsf{swap}) \,|\!)$ que espelha uma árvore de tipo LTree e as duas funções seguintes:

- $lm = (\!|\, [id, \pi_1] \,|\!)$ — vai buscar a folha mais à esquerda (lm abrevia 'leftmost')
- $rm = (\!|\, [id, \pi_2] \,|\!)$ — vai buscar a folha mais à direita (rm abrevia 'rightmost')

Mostre, por fusão-cata (em LTree) que:

$$lm \cdot \mathsf{mirror} = rm \tag{E3}$$

RESOLUÇÃO: Completar com as justificações:

$$lm \cdot m = rm$$

$$\Leftarrow \quad \{\ \ldots\ \}$$

$$lm \cdot \mathsf{in} \cdot (id + \mathsf{swap}) = [id, \pi_2] \cdot (id + lm^2)$$

$$\equiv \quad \{\ \ldots\ \}$$

$$[id, \pi_1] \cdot (id + lm^2) \cdot (id + \mathsf{swap}) = [id, \pi_2] \cdot (id + lm^2)$$

$$\equiv \quad \{\ \ldots\ \}$$

$$[id, \pi_1 \cdot lm^2 \cdot \mathsf{swap}] = [id, \pi_2 \cdot lm^2]$$

$$\equiv \quad \{\ \ldots\ \}$$

$$[id, \pi_1 \cdot \mathsf{swap} \cdot lm^2] = [id, \pi_2 \cdot lm^2]$$

$$\equiv \quad \{\ \ldots\ \}$$

$$[id, \pi_2 \cdot lm^2] = [id, \pi_2 \cdot lm^2]$$

□

□

Questão 4 Suponha que tem um catamorfismo $(\!|\, \alpha \,|\!)$ em que $\alpha : \mathsf{F}\ A \to A$ é um isomorfismo, isto é, $\alpha^\circ : A \to \mathsf{F}\ A$ é um isomorfismo também. Mostre, por fusão-cata, que

$$[\![\alpha^\circ]\!] \cdot (\!|\, \alpha \,|\!) = id. \tag{E4}$$

se verifica.

RESOLUÇÃO: Tem-se (completar com as justificações):

$$
\begin{array}{ll}
& [\![\alpha^\circ]\!] \cdot (\![\alpha]\!) = (\![\mathsf{in}]\!) \\
\Leftarrow & \{ \ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \ \} \\
& [\![\alpha^\circ]\!] \cdot \alpha = \mathsf{in} \cdot \mathsf{F}\ [\![\alpha^\circ]\!] \\
\equiv & \{ \ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \ \} \\
& \mathsf{out} \cdot [\![\alpha^\circ]\!] = \mathsf{F}\ [\![\alpha^\circ]\!] \cdot \alpha^\circ \\
\equiv & \{ \ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \ \} \\
& \mathit{true} \\
\Box &
\end{array}
$$

□

Questão 5 Considere a seguinte série definida por recorrência da seguinte forma:

$$
\begin{array}{l}
s_0 = 1 \\
s_{n+1} = 2 * s_n + n + 2
\end{array}
$$

Assim, a lista $[1, 4, 11, 26, 57, 120, 247, 502, \ldots]$ mostra os primeiros termos da série. Mostre, por aplicação da lei de recursividade mútua, que a seguinte função

$$
\begin{array}{l}
s = \pi_1 \cdot \mathsf{for}\ \mathit{loop}\ \mathsf{init}\ \mathbf{where} \\
\quad \mathit{loop}\ (x, y) = (2 * x + y, y + 1) \\
\quad \mathsf{init} = (1, 2)
\end{array}
$$

calcula o n-ésimo termo da série.

RESOLUÇÃO: Há várias maneiras de resolver esta questão. A que se apresenta a seguir faz "reverse engineering" do ciclo:

$$
\begin{array}{ll}
& \mathit{loop}\ (x, y) = (2 * x + y, y + 1) \\
\equiv & \{ \ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \ \} \\
& \mathit{loop} = \langle h, \mathsf{succ} \cdot \pi_2 \rangle\ \mathbf{where}\ h\ (x, y) = 2 * x + y
\end{array}
$$

Então (completar as justificações):

$$
\begin{array}{ll}
& s = \pi_1 \cdot \mathsf{for}\ \mathit{loop}\ \mathsf{init} \\
\equiv & \{ \text{ para algum } r \ \} \\
& \langle s, r \rangle = \mathsf{for}\ \langle h, \mathsf{succ} \cdot \pi_2 \rangle\ (1, 2) \\
\equiv & \{ \ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \ \} \\
& \langle s, r \rangle = (\![[\underline{(1, 2)}, \langle h, \mathsf{succ} \cdot \pi_2 \rangle]]\!) \\
\equiv & \{ \ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \ \} \\
& \langle s, r \rangle = (\![\langle [\underline{1}, h], [\underline{2}, \mathsf{succ} \cdot \pi_2] \rangle]\!) \\
\equiv & \{ \ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \ \} \\
& \left\{ \begin{array}{l} s \cdot \mathsf{in}_{\mathbb{N}_0} = [\underline{1}, h] \cdot (id + \langle s, r \rangle) \\ r \cdot \mathsf{in}_{\mathbb{N}_0} = [\underline{2}, \mathsf{succ} \cdot \pi_2] \cdot (id + \langle s, r \rangle) \end{array} \right.
\end{array}
$$

$$\equiv \quad \{ \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad \}$$

$$\begin{cases} s \cdot \mathsf{in}_{\mathbb{N}_0} = [\underline{1}, h \cdot \langle s, r \rangle] \\ r \cdot \mathsf{in}_{\mathbb{N}_0} = [\underline{2}, \mathsf{succ} \cdot r] \end{cases}$$

$$\equiv \quad \{ \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad \}$$

$$\begin{cases} \begin{cases} s\ 0 = 1 \\ s\ (n+1) = 2 * (s\ n) + r\ n \end{cases} \\ \begin{cases} r\ 0 = 2 \\ r\ (n+1) = 1 + r\ n \end{cases} \end{cases}$$

$$\equiv \quad \{ \ r\ n = n + 2, \text{ pois } r\ 0 = 0 + 2 = 2 \text{ e } r\ (n+1) = (n+1) + 2 = 1 + (n+2) \ \}$$

$$\begin{cases} s\ 0 = 1 \\ s\ (n+1) = 2 * (s\ n) + n + 2 \end{cases}$$

□

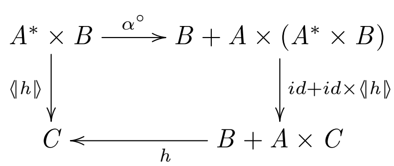
Questão 6 Considere o isomorfismo

$$\alpha = [\langle \mathsf{nil}, id \rangle, (\mathsf{cons} \times id) \cdot \mathsf{assocl}] \qquad \text{(E5)}$$

onde, como sabe,

$$\mathsf{assocl} = \langle id \times \pi_1, \pi_2 \cdot \pi_2 \rangle \qquad \text{(E6)}$$

Vamos designar por $(\!| h |\!)$ o hilomorfismo $[\![h, \alpha^\circ]\!]$:



É possível mostrar que, por α ser um isomorfismo, se tem a propriedade *universal*

$$k = (\!| h |\!) \quad \Leftrightarrow \quad k \cdot \alpha = h \cdot \mathsf{F}\ k \qquad \text{(E7)}$$

onde $\mathsf{F}\ f = id + id \times f$ é o functor das listas. Demonstre a partir de (E7) a propriedade

$$\pi_1 = (\!| \mathsf{in} |\!)$$

onde $\mathsf{in} = [\mathsf{nil}, \mathsf{cons}]$.

RESOLUÇÃO: Completar com as justificações:

$$\pi_1 = (\!| \mathsf{in} |\!)$$

$$\equiv \quad \{ \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad \}$$

$$\pi_1 \cdot \alpha = [\mathsf{nil}, \mathsf{cons}] \cdot (id + id \times \pi_1))$$

$$\equiv \quad \{ \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad \}$$

$$\pi_1 \cdot \alpha = [\mathsf{nil}, \mathsf{cons} \cdot (id \times \pi_1)]$$

$$\equiv \quad \{ \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad \}$$

$$\pi_1 \cdot \alpha = [\mathsf{nil}, \mathsf{cons} \cdot \pi_1 \cdot \mathsf{assocl}]$$

□

□

Questão 7 Considere o hilomorfismo

$$h = \textbf{tailr}\ g\ \textbf{where}$$
$$g = (\pi_2 + \mathsf{swap}) \cdot z? \cdot \langle \widehat{mod}, \pi_2 \rangle$$
$$z\ (r, b) = r \equiv 0$$

onde mod é a função com esse nome em Haskell.

1. Faça um diagrama para h e converta-a numa função recursiva com variáveis que não recorra a nenhum combinador 'pointfree' que estudou nesta disciplina, justificando cada passo do seu raciocínio.

2. A função h implementa um algoritmo célebre — consegue identificá-lo?

RESOLUÇÃO: Alíneas:

1. Como **tailr** $g = [\![[id, id], g]\!]$ para $\mathsf{F} = id + f$:[1]

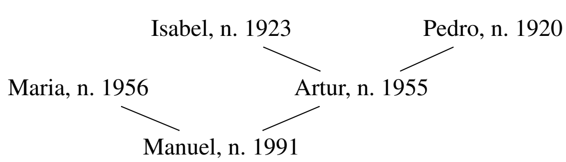
$$h = \textbf{tailr}\ g$$
$$\equiv \quad \{ \quad \dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots \quad \}$$
$$h = [id, id] \cdot (id + h) \cdot (\pi_2 + \mathsf{swap}) \cdot z? \cdot \langle \widehat{mod}, \pi_2 \rangle$$
$$\equiv \quad \{ \quad \dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots \quad \}$$
$$h = [\pi_2, h \cdot \mathsf{swap}] \cdot z? \cdot \langle \widehat{mod}, \pi_2 \rangle$$
$$\equiv \quad \{ \quad \dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots \quad \}$$
$$h = (z \rightarrow \pi_2\ ,\ h \cdot \mathsf{swap}) \cdot \langle \widehat{mod}, \pi_2 \rangle$$
$$\equiv \quad \{ \quad \dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots \quad \}$$
$$h = (z \cdot \langle \widehat{mod}, \pi_2 \rangle) \rightarrow \pi_2\ ,\ h \cdot \langle \pi_2, \widehat{mod} \rangle$$
$$\equiv \quad \{ \quad \dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots \quad \}$$
$$h\ (a, b) = \textbf{if}\ z\ (a\ \text{`}mod\text{`}\ b, b)\ \textbf{then}\ b\ \textbf{else}\ h\ (b, a\ \text{`}mod\text{`}\ b)$$
$$\equiv \quad \{ \quad \dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots \quad \}$$
$$h\ (a, b) = \textbf{if}\ a\ \text{`}mod\text{`}\ b = 0\ \textbf{then}\ b\ \textbf{else}\ h\ (b, a\ \text{`}mod\text{`}\ b)$$

2. Algoritmo de Euclides (encontra o máximo divisor comum entre dois números inteiros)

$$mdc\ a\ b = \textbf{let}\ r = mod\ a\ b\ \textbf{in}\ \textbf{if}\ r \equiv 0\ \textbf{then}\ b\ \textbf{else}\ mdc\ b\ r$$

□

Questão 8 Uma árvore genealógica ascendente indica, para cada indivíduo, a árvore da sua mãe e a do seu pai, quando conhecidas. Por exemplo:



(E8)

[1] Completar com as justificações.

Em Haskell podemos definir o tipo genérico para estas árvores que se segue:

$$\mathbf{data}\ GenT\ a = GenT\ \{ind :: a, mother :: Maybe\ (GenT\ a), father :: Maybe\ (GenT\ a)\}$$

isto é:

$$GenT\ A \cong A \times (1 + GenT\ A)^2$$

1. Defina os combinadores ana-cata-hilo para este tipo de dados.
2. Seja dado, em Haskell, o tipo

   $$\mathbf{data}\ Ind\ \{name :: String, birth :: String\}$$

   para descrever pessoas. Escreva como um anamorfismo a função

   $$names :: GenT\ Ind \to GenT\ String$$

   que elimina da árvore argumento toda a informação que não seja o nome de cada pessoa.

RESOLUÇÃO: Comece-se por:

$$\mathsf{B}\ (f, g) = f \times (id + g)^2\ \mathbf{where}\ f^2 = f \times f$$

e, de seguida,

$$inGenT\ (a, (x, y)) = GenT\ a\ (inM\ x)\ (inM\ y)$$
$$outGenT\ (GenT\ a\ x\ y) = (a, (outM\ x, outM\ y))$$

onde

$$inM = [nothing, \mathsf{Just}]$$
$$outM\ \mathsf{Nothing} = i_1\ (); outM\ (\mathsf{Just}\ a) = i_2\ a$$

1. Daqui se deriva a trilogia ana-cata-hilo da formal habitual:

   $$\mathsf{F}\ f = \mathsf{B}\ (id, f)$$
   $$cataGenT\ f = f \cdot \mathsf{F}\ (cataGenT\ f) \cdot outGenT$$
   $$anaGenT\ f = inGenT \cdot \mathsf{F}\ (anaGenT\ f) \cdot f$$
   $$hyloGenT\ f\ g = (cataGenT\ f) \cdot (anaGenT\ g)$$

2. O selector $name : Ind \to String$ seleciona o nome de um indivíduo. O "fmap"

   $$GenT\ name$$

   é a função que se pede. Por **Def-map-ana** (57):

   $$GenT\ name = [\![\mathsf{B}\ (f, id) \cdot outGenT]\!]$$

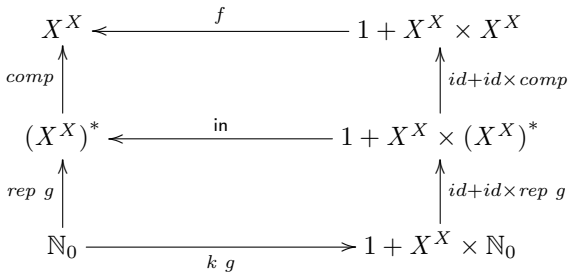
   onde B e $outGenT$ foram definidas acima.

□

Questão 9 Recorde a seguinte questão do teste de 13 de Junho: dada uma função $g{:}X \to X$, construir um catamorfismo de números naturais $many\ g : \mathbb{N}_0 \to X \to X$ tal que, dado um número natural n e um valor x, $many\ g\ n\ x$ retorna a aplicação de g ao valor x, n vezes:

$$many\ g\ 0\ x = x$$
$$many\ g\ (n+1)\ x = g\ (many\ g\ n\ x)$$

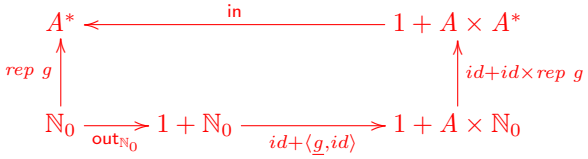
É possível mostrar que $many\ g$ é um hilomorfismo cuja primeira parte (anamorfismo) replica g tantas vezes quanto o segundo argumento, compondo depois essas réplicas no catamorfismo. Defina f e k por forma a que $many\ g = [\![f, k\ g]\!]$, cf. o seguinte diagrama:



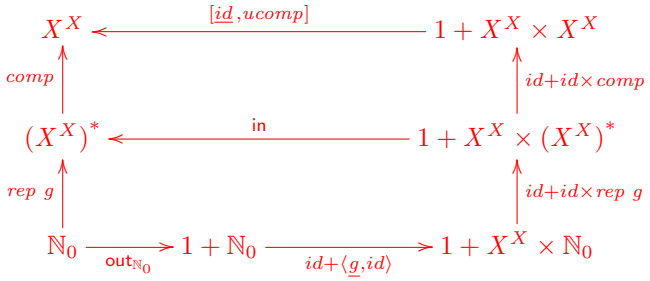
RESOLUÇÃO: O anamorfismo

$$rep\ a = [\![(k\ a)]\!]\ \mathbf{where}\ k\ a = (id + \langle \underline{a}, id\rangle) \cdot \mathsf{out}_{\mathbb{N}_0}$$

é polimórfico em a no sentido de que replica *qualquer* a tantas vezes quanto o argumento:



Já o catamorfismo $comp = (\!|\ [\underline{id}, ucomp]\ |\!)$ — para $ucomp\ (f, g) = f \cdot g$ — deve aceitar uma lista de funções, que irá compor entre si. Logo $A = X^X$ deverá ser o tipo dessas funções, paramétrico em X. Ter-se-á então, finalmente:



□

Questão 10 Relembrando o mónade de estado, complete a prova da seguinte iguadade de acções

$$modify\ f = \mathbf{do}\ \{a \leftarrow get; put\ (f\ a)\} \tag{E9}$$

preenchendo as caixas numeradas:

$$
\begin{array}{ll}
 & \mathbf{do}\ \{\, a \leftarrow get; put\ (f\ a)\,\} \\
= & \qquad \{\ \boxed{1}\ \} \\
 & get \ggg (put \cdot f) \\
= & \qquad \{\ \text{definição de } (\ggg)\ \} \\
 & (\mu \cdot \boxed{2})\ get \\
= & \qquad \{\ \boxed{3}\ \} \\
 & \mathsf{ap} \cdot (put \cdot f \times id) \cdot \boxed{4} \\
= & \qquad \{\ \boxed{5}\ \} \\
 & \widehat{put} \cdot (f \times id) \cdot get \\
= & \qquad \{\ \text{definições de } put \text{ e } get, \text{ e } \boxed{6}\ \} \\
 & \boxed{7} \cdot \langle f, id \rangle \\
= & \qquad \{\ \boxed{8}\ \} \\
 & \langle !, f \rangle \\
= & \qquad \{\ \text{definição de } modify\ \} \\
 & modify\ f \\
\Box &
\end{array}
$$

RESOLUÇÃO:

$$
\begin{array}{ll}
 & \mathbf{do}\ \{\, a \leftarrow get; put\ (f\ a)\,\} \\
= & \qquad \{\ \boxed{1} = (85);\ (73)\ \} \\
 & get \ggg (put \cdot f) \\
= & \qquad \{\ \text{definição de } (\ggg)\ \} \\
 & (\mu \cdot \underbrace{(put \cdot f \times id)^S}_{\boxed{2}})\ get \\
= & \qquad \{\ \boxed{3} = \mu = \mathsf{ap}^S\ (4.42);\ f^S\ g = f \cdot g\ (2.87)\ \} \\
 & \mathsf{ap} \cdot (put \cdot f \times id) \cdot \underbrace{get}_{\boxed{4}} \\
= & \qquad \{\ \boxed{5} = \text{functor-}\times\ (14)\ ;\ put = \widehat{put};\ \text{cancelamento-exp}\ (34)\ \} \\
 & \widehat{put} \cdot (f \times id) \cdot get \\
= & \qquad \{\ \text{definições de } put \text{ e } get, \text{ e } \underbrace{(11)}_{\boxed{6}}\ (\text{absoção-}\times)\ \} \\
 & \underbrace{\langle !, \pi_1 \rangle}_{\boxed{7}} \cdot \langle f, id \rangle \\
= & \qquad \{\ \underbrace{(9);\ (7);\ (3)}_{\boxed{8}} — \text{i.é fusão-}\times, \text{cancelamento-}\times, \text{natural-const}\ \}
\end{array}
$$

$$
\begin{array}{ll}
 & \langle !, f \rangle \\
= & \quad \{ \text{ definição de } modify \ \} \\
 & modify\ f \\
\square &
\end{array}
$$

$\square$
