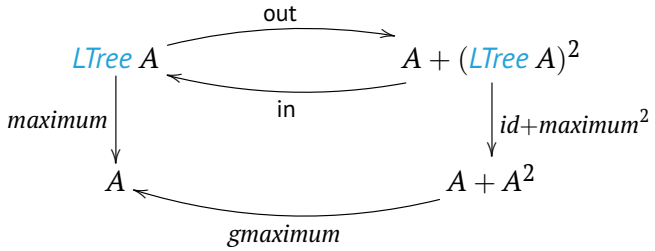
Cálculo de Programas

Resolução - Ficha 09

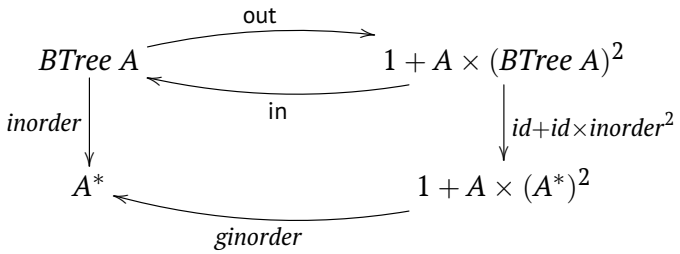
Eduardo Freitas Fernandes

2026

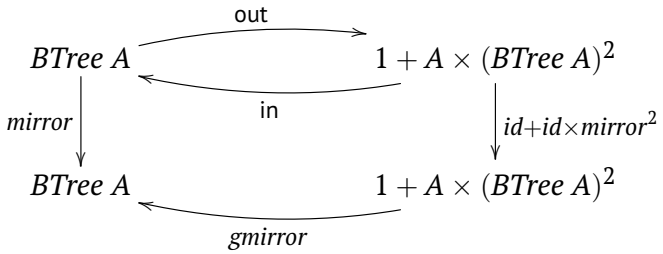
Exercício 1



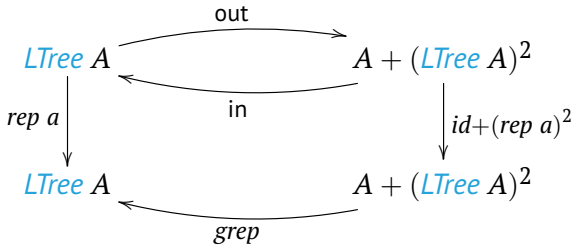
$$gmaximum = [id, umax]$$



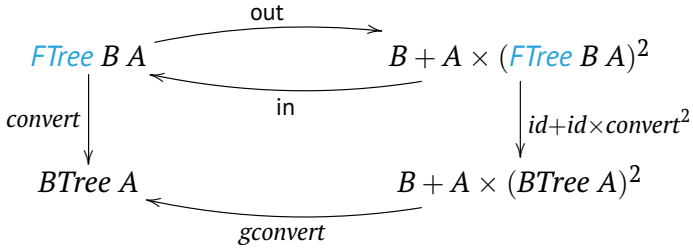
$$ginorder = [nil, aux]$$
$$\textbf{where } aux\ (h, (l, r)) = l +\!\!+ [h] +\!\!+ r$$



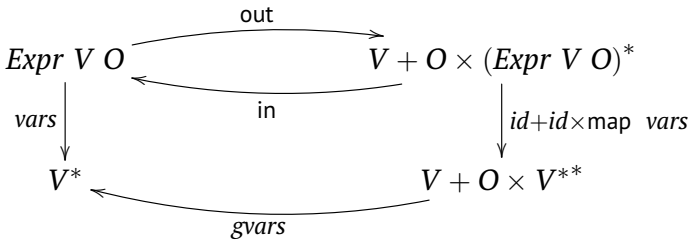
$$gmirror = [\underline{Empty}, Node \cdot (id \times swap)]$$



$$grep\ a = [\underline{a}, Fork]$$



$$gconvert = [\underline{Empty}, Node]$$



$$gvars = [singl, concat \cdot \pi_2]$$

Exercício 2

$$
\begin{array}{ll}
 & tar = (\!|\, [singl \cdot nil, g] \,|\!) \\
\equiv & \quad \{\ \text{universal-cata}\ \} \\
 & tar \cdot [\underline{Empty}, Node] = [singl \cdot nil, g] \cdot (id + id \times (tar \times tar)) \\
\equiv & \quad \{\ \text{fusão-+, absorção-+}\ \} \\
 & [tar \cdot \underline{Empty}, tar \cdot Node] = [singl \cdot nil, g \cdot (id \times (tar \times tar))] \\
\equiv & \quad \{\ \text{eq-+}\ \} \\
 & \left\{ \begin{array}{l} tar \cdot \underline{Empty} = singl \cdot nil \\ tar \cdot \overline{Node} = g \cdot (id \times (tar \times tar)) \end{array} \right. \\
\equiv & \quad \{\ \text{pointwise, def. comp}\ \} \\
 & \left\{ \begin{array}{l} tar\ Empty = [\,] \\ tar\ (Node\ (x,(l,r))) = g\ (x,(tar\ l, tar\ r)) \end{array} \right. \\
\equiv & \quad \{\ \text{def. g}\ \} \\
 & \left\{ \begin{array}{l} tar\ Empty = [\,] \\ tar\ (Node\ (x,(l,r))) = (\mathsf{map}\ \ cons \cdot lstr)\ (x, tar\ l +\!\!+ tar\ r) \end{array} \right. \\
\equiv & \quad \{\ \text{def. comp, def. } lstr\ \} \\
 & \left\{ \begin{array}{l} tar\ Empty = [\,] \\ tar\ (Node\ (x,(l,r))) = \mathsf{map}\ \ cons\ [(x,a) \mid a \leftarrow tar\ l +\!\!+ tar\ r] \end{array} \right. \\
\equiv & \quad \{\ \text{def. map cons}\ \} \\
 & \left\{ \begin{array}{l} tar\ Empty = [\,] \\ tar\ (Node\ (x,(l,r))) = [h : t \mid (h,t) \leftarrow [(x,a) \mid a \leftarrow tar\ l +\!\!+ tar\ r]] \end{array} \right.
\end{array}
$$

Exercício 3

$$
\begin{array}{ll}
 & vars = (\!|\, [singl, concat \cdot \pi_2] \,|\!) \\
\equiv & \quad \{\ \text{universal-cata}\ \} \\
 & vars \cdot [Var, Term] = [singl, concat \cdot \pi_2] \cdot (id + id \times \mathsf{map}\ \ vars) \\
\equiv & \quad \{\ \text{fusão-+, absorção-+, eq-+}\ \} \\
 & \left\{ \begin{array}{l} vars \cdot Var = singl \\ vars \cdot Term = concat \cdot \pi_2 \cdot (id \times \mathsf{map}\ \ vars) \end{array} \right. \\
\equiv & \quad \{\ \text{natural-}\pi_2\text{, pointwise}\ \} \\
 & \left\{ \begin{array}{l} vars\ (Var\ v) = [v] \\ vars\ (Term\ (o,l)) = concat\ (\mathsf{map}\ \ vars\ l) \end{array} \right.
\end{array}
$$

Exercício 4

$$
\begin{array}{ll}
& k = [\!((id + \langle f, id \rangle) \cdot \mathsf{out})\!] \\
\equiv & \quad \{\ \text{universal-ana}\ \} \\
& \mathsf{out} \cdot k = (id + id \times k) \cdot (id + \langle f, id \rangle) \cdot \mathsf{out} \\
\equiv & \quad \{\ \text{shunt-left, shunt-right}\ \} \\
& k \cdot \mathsf{in} = \mathsf{in} \cdot (id + id \times k) \cdot (id + \langle f, id \rangle) \\
\equiv & \quad \{\ \text{functor-+}\ \} \\
& k \cdot [\underline{0}, \mathsf{succ}] = [nil, cons] \cdot (id + (id \times k) \cdot \langle f, id \rangle) \\
\equiv & \quad \{\ \text{absorção-}\times\text{, absorção-+}\ \} \\
& k \cdot [\underline{0}, \mathsf{succ}] = [nil, cons \cdot \langle f, k \rangle] \\
\equiv & \quad \{\ \text{fusão-+, eq-+, pointwise}\ \} \\
& \begin{cases} k\ 0 = [\,] \\ k\ (n+1) = f\ n : k\ n \end{cases}
\end{array}
$$

Exercício 5

$$
\begin{array}{ll}
& suffixes = [\!((id + \langle cons, \pi_2 \rangle) \cdot \mathsf{out})\!] \\
\equiv & \quad \{\ \text{universal-ana}\ \} \\
& \mathsf{out} \cdot suffixes = (id + id \times suffixes) \cdot (id + \langle cons, \pi_2 \rangle) \cdot \mathsf{out} \\
\equiv & \quad \{\ \text{shunt-left, shunt-right}\ \} \\
& suffixes \cdot \mathsf{in} = \mathsf{in} \cdot (id + id \times suffixes) \cdot (id + \langle cons, \pi_2 \rangle) \\
\equiv & \quad \{\ \text{functor-+, fusão-+}\ \} \\
& [suffixes \cdot nil, suffixes \cdot cons] = [nil, cons] \cdot (id + ((id \times suffixes) \cdot \langle cons, \pi_2 \rangle)) \\
\equiv & \quad \{\ \text{absorção-+}\ \} \\
& [suffixes \cdot nil, suffixes \cdot cons] = [nil, cons \cdot ((id \times suffixes) \cdot \langle cons, \pi_2 \rangle)] \\
\equiv & \quad \{\ \text{absorção-}\times\ \} \\
& [suffixes \cdot nil, suffixes \cdot cons] = [nil, cons \cdot \langle cons, suffixes \cdot \pi_2 \rangle] \\
\equiv & \quad \{\ \text{eq-+, pointwise}\ \} \\
& \begin{cases} suffixes\ [\,] = [\,] \\ suffixes\ (h : t) = (h : t) : suffixes\ t \end{cases}
\end{array}
$$

Exercício 6

$$
\begin{array}{ll}
& (\!|\, [\underline{0}, \mathsf{succ} \cdot \pi_2] \,|\!) = anaNat\ ((id + \pi_2) \cdot \mathsf{out}) \\
\equiv & \quad \{\ \text{universal-ana}\ \} \\
& \mathsf{out} \cdot (\!|\, [\underline{0}, \mathsf{succ} \cdot \pi_2] \,|\!) = \mathsf{F}\ (\!|\, \cdot \,|\!)\ [\underline{0}, \mathsf{succ} \cdot \pi_2] \cdot (id + \pi_2) \cdot \mathsf{out} \\
\equiv & \quad \{\ \text{shunt-left, shunt-right, def. functor dos naturais}\ \} \\
& (\!|\, [\underline{0}, \mathsf{succ} \cdot \pi_2] \,|\!) \cdot \mathsf{in} = \mathsf{in} \cdot (id + (\!|\, [\underline{0}, \mathsf{succ} \cdot \pi_2] \,|\!)) \cdot (id + \pi_2) \\
\equiv & \quad \{\ \text{functor-+, def. in, cancelamento-cata}\ \} \\
& [\underline{0}, \mathsf{succ} \cdot \pi_2] \cdot (id + id \times (\!|\, [\underline{0}, \mathsf{succ} \cdot \pi_2] \,|\!)) = [\underline{0}, \mathsf{succ}] \cdot (id + (\!|\, [\underline{0}, \mathsf{succ} \cdot \pi_2] \,|\!) \cdot \pi_2)
\end{array}
$$

$$
\begin{aligned}
&\equiv \quad \{ \text{ absorção-}+ \ \} \\
&\quad [\underline{0}, \mathsf{succ} \cdot \pi_2 \cdot (id \times (\!| [\underline{0}, \mathsf{succ} \cdot \pi_2] |\!))] = [\underline{0}, \mathsf{succ} \cdot (\!| [\underline{0}, \mathsf{succ} \cdot \pi_2] |\!) \cdot \pi_2] \\
&\equiv \quad \{ \text{ eq-}+\text{, natural-}\pi_2 \ \} \\
&\quad \begin{cases} \underline{0} = \underline{0} \\ \mathsf{succ} \cdot (\!| [\underline{0}, \mathsf{succ} \cdot \pi_2] |\!) \cdot \pi_2 = \mathsf{succ} \cdot (\!| [\underline{0}, \mathsf{succ} \cdot \pi_2] |\!) \cdot \pi_2 \end{cases} \\
&\square
\end{aligned}
$$

Exercício 7

```
Data QTree a = Pixel a | Blocks ((QTree a, QTree a), (QTree a, QTree a))
  deriving (Show)
inQTree :: a + ((QTree a, QTree a), (QTree a, QTree a)) → QTree a
inQTree = [Pixel, Blocks]
outQTree :: QTree a → a + ((QTree a, QTree a), (QTree a, QTree a))
outQTree (Pixel x) = i1 x
outQTree (Blocks ((x, y), (z, w))) = i2 ((x, y), (z, w))
  -- Bi-Functor de QTree
baseQTree g f = g + ((f × f) × (f × f))
  -- Functor de QTree
recQTree f = baseQTree id f
  -- catamorfismo de QTree
cataQTree g = g · (recQTree (cataQTree g)) · outQTree
  -- anamorfismo de QTree
anaQTree g = inQTree · (recQTree (anaQTRee g)) · g
  -- hylomorfismo de QTree
hyloQTree f g = cataQTree f · anaQTree g
instance Functor QTree where
  fmap f = cataQTree (inQTree · baseQTree f id)
mirrorQTree = cataQTree [Pixel, Blocks · swap · (swap × swap)]
countQTree = cataQTree [one, add · (add × add)]
depthQTree = cataQTree [one, succ · umax · (umax × umax)]
rotate90 = cataQTree [Pixel, Blocks · f]
  where f ((x, y), (z, w)) = ((w, x), (y, z))
tips = cataQTree [singl, conc · (conc × conc)]
```