Problem 1: $F(\Sigma\ A\ B) \cong \Sigma\ (FA)\ (\Box_F B))$

Here is a useful little construction: let $\widehat{\cdot} : ((x : A) \to B(a)) \to A \to \Sigma\ A\ B$ be defined by $\widehat{g}(x) = \langle x, g(x)\rangle$. Notice that $\pi_0 \circ \widehat{g} = \mathrm{id}$ and $\pi_1 \circ \widehat{g} = g$ for any g.

Definition 1. Given functor $F : \mathrm{Set} \to \mathrm{Set}$, we define

$$\begin{aligned} \Box_F &: \{A : \mathrm{Set}\} \to (P : A \to \mathrm{Set}) \to F(A) \to \mathrm{Set} \\ \mathrm{dmap}_F &: \{A : \mathrm{Set}\} \to \{P : A \to \mathrm{Set}\} \to ((x : A) \to P(x)) \to (x : F(A)) \to \Box_F(P, x) \end{aligned}$$

by

$$\begin{aligned} \Box_F(B, x) &= \{y : F\ (\Sigma\ A\ B) \mid F(\pi_0)(y) = x\} \\ \mathrm{dmap}_F(g, x) &= \langle F(\widehat{g})(x), \mathrm{refl}\rangle \end{aligned}$$

Notice that we are using the fact that $\pi_0 \circ \widehat{g} = \mathrm{id}$, so that $F(\pi_0)(F(\widehat{g})(x)) = F(\pi_0 \circ \widehat{g})(x) = F(\mathrm{id})(x) = x$, and refl has the right type.

Problem 1. (i) There is $\varphi : F(\Sigma\ A\ B) \xrightarrow{\cong} \Sigma\ (FA)\ (\Box_F B))$.

(ii) $\pi_0 \circ \varphi = F(\pi_0)$.

(iii) For $g : (x : A) \to B(x)$, we have $\widehat{\mathrm{dmap}_F}(g) = \varphi \circ F(\widehat{g})$.

Proof. (i) Define

$$\varphi : F(\Sigma\ A\ B) \to \Sigma\ (FA)\ (\Box_F B))$$

and

$$\psi : \Sigma\ (FA)\ (\Box_F B)) \to F(\Sigma\ A\ B)$$

by

$$\begin{aligned} \varphi(y) &= \langle F(\pi_0)(y), \langle y, \mathrm{refl}\rangle\rangle \\ \psi(\langle x, \langle y, p\rangle\rangle) &= y. \end{aligned}$$

Then $\psi(\varphi(y)) = \psi(\langle F(\pi_0)\ y, \langle y, \mathrm{refl}\rangle\rangle) = y$ and for every $\langle x, \langle y, p\rangle\rangle : \Sigma\ (FA)\ (\Box_F B))$, we have $x = F(\pi_0)(y)$ by p and $p = \mathrm{refl}$ by proof irrelevance, so that $\varphi(\psi(\langle x, \langle y, p\rangle\rangle)) = \varphi(y) = \langle F(\pi_0)(y), \langle y, \mathrm{refl}\rangle\rangle = \langle x, \langle y, p\rangle\rangle$. Hence $F(\Sigma\ A\ B) \cong \Sigma\ (FA)\ (\Box_F B))$.

(ii) $\pi_0(\varphi(y)) = \pi_0(\langle F(\pi_0)(y), \langle y, \mathrm{refl}\rangle\rangle) = F(\pi_0)(y)$.

(iii) $\widehat{\mathrm{dmap}_F}(g)(x) = \langle x, \mathrm{dmap}_F(g, x)\rangle = \langle x, \langle F(\widehat{g})(x), \mathrm{refl}\rangle\rangle$, and also $\varphi(F(\widehat{g})(x)) = \langle F(\pi_0)(F(\widehat{g})(x)), \langle (F(\widehat{g})(x), \mathrm{refl}\rangle\rangle = \langle F(\widehat{\pi_0 \circ g})(x)), \langle (F(\widehat{g})(x), \mathrm{refl}\rangle\rangle = \langle x, \langle (F(\widehat{g})(x), \mathrm{refl}\rangle\rangle$ since $\pi_0 \circ \widehat{g} = \mathrm{id}$. Hence $\widehat{\mathrm{dmap}_F}(g) = \varphi \circ F(\widehat{g})$.

$\square$

Problem 2: $\Box$ for containers $\cong$ $\Box$ for functors

Definition 2. For a container $S \lhd P$, we define

$$\Box_{S\lhd P} : \{A : \mathrm{Set}\} \to (B : A \to \mathrm{Set}) \to [\![S \lhd P]\!]\ A \to \mathrm{Set}$$

by $\Box_{S\lhd P}(B, \langle s, f\rangle) = (p : P(s)) \to B(f(p))$.

Problem 2. For all $A : \mathrm{Set}$, $B : A \to \mathrm{Set}$ and $\langle s, f\rangle : [\![S \lhd P]\!](A)$,

$$\Box_{S\lhd P}(B, \langle s, f\rangle) \cong \Box_{[\![S\lhd P]\!]}(B, \langle s, f\rangle).$$

Proof. Define

$$\varphi : ((p : P(s)) \to B(f(p))) \to \{\langle s', f'\rangle : \Sigma s' : S\ .(P(s') \to \Sigma\ A\ B) \mid \langle s', \pi_0 \circ f'\rangle = \langle s, f\rangle\}$$

and

$$\psi : \{\langle s', f'\rangle : \Sigma s' : S\ .(P(s') \to \Sigma\ A\ B) \mid \langle s', \pi_0 \circ f'\rangle = \langle s, f\rangle\} \to ((p : P(s)) \to B(f(p)))$$

by

$$\varphi(g) = \langle\langle s, (\lambda p\ .\ \langle f(p), g(p)\rangle)\rangle, \langle \mathrm{refl}, \mathrm{ext}(\mathrm{refl})\rangle\rangle$$
$$\psi(\langle\langle s', f'\rangle, r\rangle) = \pi_1 \circ f'$$

(there's lots of invented notation for equality proofs and hidden substitutions going on here). Now $\psi(\varphi(g)) = \psi(\langle\langle s, (\lambda p\ .\ \langle f(p), g(p)\rangle)\rangle, \langle \mathrm{refl}, \mathrm{ext}(\mathrm{refl})\rangle\rangle) = \pi_1 \circ (\lambda p\ .\ \langle f(p), g(p)\rangle) = g$ and

$$\begin{aligned}
\varphi(\psi(\langle\langle s', f'\rangle, r\rangle)) &= \varphi(\pi_1 \circ f') \\
&= \langle\langle s, (\lambda p\ .\ \langle f(p), \pi_1(f'(p))\rangle)\rangle, \langle \mathrm{refl}, \mathrm{ext}(\mathrm{refl})\rangle\rangle \\
&= \langle\langle s', (\lambda p\ .\ \langle \pi_0(f'(p)), \pi_1(f'(p))\rangle)\rangle, r\rangle \\
&= \langle\langle s', f'\rangle, r\rangle
\end{aligned}$$

where we have used eta for Σ and Π in the last equality. $\Box$

We can make $\Box_F$ into a functor $\boxdot_F$: Fam(Set) $\to$ Fam(Set) [in general Fam($\mathbb{C}$) $\to$ Fam($\mathbb{D}$)?] by defining $\boxdot_F(A, B) = (F(A), \Box_F(A, B))$ for objects and $\boxdot(f, g) = (F(f), \lambda\ x\ \langle y, \mathrm{refl}\rangle\ .\langle F([f, g])(y), \mathrm{refl}\rangle)$ where $[f, g] : \Sigma\ A\ B \to \Sigma\ A'\ B'$ is the obvious map, and the last refl is of the right type since $\pi_0 \circ [f, g] = f \circ \pi_0$.

Similarly, we can define $\boxdot_{S\lhd P}(A, B) = ([\![S \lhd P]\!](A), \Box_{S\lhd P})$ for objects and and $\boxdot_{S\lhd P}(f, g) = ([\![S \lhd P]\!](f), \lambda(s, h)\ j\ p\ .\ g(h(p), j(p)))$ for morphisms $(f, g) : (A, B) \to (A', B')$. The result can now be strengthened to:

Proposition 3. *There is a natural isomorphism* $\eta : \Box_{S\lhd P} \Rightarrow \Box_{[\![S\lhd P]\!]}$.

Proof. We have already constructed the components at each (A, B) and shown them to be isomorphisms in the proof of Problem 2. All that is left to do is to check the naturality condition, but this follows from a straightforward verification:

$$\begin{tikzcd} j \arrow[r, mapsto] \arrow[d, mapsto] & \langle\langle s, (\lambda\ p\ .\ \langle h(p), j(p)\rangle)\rangle, \mathrm{refl}\rangle \arrow[d, mapsto] \\ \lambda\ p\ .\ g(h(p), j(p)) \arrow[r, mapsto] & \langle\langle s, (\lambda\ p\ .\ \langle f(h(p)), g(h(p), j(p))\rangle)\rangle, \mathrm{refl}\rangle \end{tikzcd}$$

$\square$

Problem 3: Init $\implies$ elim (in Set)

Let $F : \mathrm{Set} \to \mathrm{Set}$ be a functor and $(\mu F, \mathrm{in}_F)$ a F-algebra, i.e.

$$\mu F : \mathrm{Set} \qquad\qquad \mathrm{in}_F : F(\mu F) \to \mu F.$$

$\square_F$ and dmap_F

Let us first record what we need and expect from $\square_F$ and dmap_F. They should have types

$$\begin{aligned} \square_F &: \{A : \mathrm{Set}\} \to (P : A \to \mathrm{Set}) \to F(A) \to \mathrm{Set} \\ \mathrm{dmap}_F &: \{A : \mathrm{Set}\} \to \{P : A \to \mathrm{Set}\} \to ((x : A) \to P(x)) \to (x : F(A)) \to \square_F(P, x) \end{aligned}$$

and satisfy

$$F(\Sigma\ A\ P) \cong \Sigma\ (FA)\ (\square_F P)) \tag{$*_\square$}$$

and this isomorphism φ must satisfy

$$\begin{tikzcd} F(\Sigma\ A\ P) \arrow[r, "\varphi"] \arrow[d, "F(\pi_0)"'] & \Sigma\ (F\ A)\ (\square_F\ P) \arrow[dl, "\pi_0"] \\ F\ A & \end{tikzcd} \tag{$**_\square$}$$

i.e. $\pi_0 \circ \varphi = F(\pi_0)$. For dmap_F, we must for all $f : (x : A) \to B(x)$ have

$$\begin{tikzcd} F\ A \arrow[r, "F(\widehat{f})"] \arrow[d, "\widehat{\mathrm{dmap}_F(f)}"'] & F(\Sigma\ A\ B) \arrow[dl, "\varphi"] \\ \Sigma\ (F\ A)\ (\square_F\ B) & \end{tikzcd} \tag{$*_{\mathrm{dmap}}$}$$

i.e. $\widehat{\mathrm{dmap}_F(f)} = \varphi \circ F(\widehat{f})$.

For $\square_F(P, x) = \{y : F\ (\Sigma\ A\ P) \mid F(\pi_0)(y) = x\}$ and $\mathrm{dmap}_F(g, x) = \langle F(\widehat{g})(x), \mathrm{refl}\rangle$, this holds, as proved in Problem 1.

The equivalence of elim and init

Principle (Elim)**.** The elimination principle for F says that we have

$$\begin{aligned}\mathrm{elim}_F : (P : \mu F \to \mathrm{Set}) \to \big((x : F(\mu F)) \to \Box_F(P, x) \to P(\mathrm{in}_F(x))\big) \\ \to (x : \mu F) \to P(x)\end{aligned}$$

and computation rule

$$\mathrm{elim}_F(P, g, \mathrm{in}_F(x)) = g(x, \mathrm{dmap}_F(\mathrm{elim}_F(P, g), x))$$

Principle (Init)**.** The initial algebra principle for F says that $(\mu F, \mathrm{in}_F)$ is the *initial* algebra for F, i.e. for any other F-algebra (X, f), there is $\mathrm{fold}(f) : \mu F \to X$ such that

$$\begin{tikzcd} F(\mu F) \arrow[r, "\mathrm{in}_F"] \arrow[d, "F(\mathrm{fold}(f))"'] & \mu F \arrow[d, "\mathrm{fold}(f)"] \\ F(X) \arrow[r, "f"] & X \end{tikzcd}$$

commutes.

Problem 3. Init $\implies$ elim.

Proof. Assume that $(\mu F, \mathrm{in}_F)$ is initial. We must construct

$$\begin{aligned}\mathrm{elim}_F : (P : \mu F \to \mathrm{Set}) \to \big((x : F(\mu F)) \to \Box_F(P, x) \to P(\mathrm{in}_F(x))\big) \\ \to (x : \mu F) \to P(x)\end{aligned}$$

such that

$$\mathrm{elim}_F(P, g, \mathrm{in}_F(x)) = g(x, \mathrm{dmap}_F(\mathrm{elim}_F(P, g), x)).$$

Let $P : \mu F \to \mathrm{Set}$ and $g : (x : F(\mu F)) \to \Box_F(P, x) \to P(\mathrm{in}_F(x))$ be given. The plan is to make $\Sigma\ \mu F\ P$ into a F-algebra $(\Sigma\ \mu F\ P, h)$ and then show that $\pi_0 \circ \mathrm{fold}(h) = \mathrm{id}$, so that $\pi_1 \circ \mathrm{fold}(h) : (x : \mu F) \to P(x)$. We can then define $\mathrm{elim}_F(P, g, x) = \pi_1(\mathrm{fold}(f)(x))$ and must show that the computation rule holds.

First, we need to define $h : F(\Sigma\ \mu F\ P) \to \Sigma\ \mu F\ P$. Let $\varphi : F(\Sigma\ \mu F\ P) \xrightarrow{\cong} \Sigma\ (F\ A)\ (\Box_F\ P)$ be the witness of $(*_\Box)$. We can then define

$$h(x) = \langle \mathrm{in}_F(\pi_0(\varphi(x))), g(\pi_0(\varphi(x)), \pi_1(\varphi(x)))\rangle,$$

so that $\mathrm{fold}(h) : \mu F \to \Sigma\ \mu F\ P$. We also know that

$$\begin{tikzcd} F(\mu F) \arrow[r, "\mathrm{in}_F"] \arrow[d, "F(\mathrm{fold}(h))"'] & \mu F \arrow[d, "\mathrm{fold}(h)"] \\ F(\Sigma\ \mu F\ P) \arrow[r, "h"] & \Sigma\ \mu F\ P \end{tikzcd} \tag{1}$$

commutes. Now $\pi_0(h(x)) = \mathrm{in}_F(\pi_0(\varphi(x))) \overset{(**_\square)}{=} \mathrm{in}_F(F(\pi_0)(x))$, so that the diagram

$$\begin{tikzcd}
F(\Sigma\ \mu F\ P) \arrow[r, "h"] \arrow[d, "F(\pi_0)"'] & \Sigma\ \mu F\ P \arrow[d, "\pi_0"] \\
F(\mu F) \arrow[r, "\mathrm{in}_F"] & \mu F
\end{tikzcd} \tag{2}$$

commutes. Pasting Diagram (1) and (2) together, we get the commuting diagram

$$\begin{tikzcd}
F(\mu F) \arrow[r, "\mathrm{in}_F"] \arrow[d, "F(\mathrm{fold}(h))"'] & \mu F \arrow[d, "\mathrm{fold}(h)"] \\
F(\Sigma\ \mu F\ P) \arrow[r, "h"] \arrow[d, "F(\pi_0)"'] & \Sigma\ \mu F\ P \arrow[d, "\pi_0"] \\
F(\mu F) \arrow[r, "\mathrm{in}_F"] & \mu F
\end{tikzcd}$$

which shows that $\pi_0 \circ \mathrm{fold}(h) : (\mu F, \mathrm{in}_F) \to (\mu F, \mathrm{in}_F)$ is a morphism in the category of F-algebras. But also $\mathrm{id}_{\mu F}$ is such a morphism, and since $(\mu F, \mathrm{in}_F)$ is initial, we must have $\pi_0 \circ \mathrm{fold}(h) = \mathrm{id}$. Hence we can define $\mathrm{elim}_F(P, g) := \pi_1 \circ \mathrm{fold}(h) : (x : \mu F) \to P(x)$.

It remains to be shown that $\mathrm{elim}_F(P, g, \mathrm{in}_F(x)) = g(x, \mathrm{dmap}_F(\mathrm{elim}_F(P, g), x))$. Unfolding the definition of elim_F, we see that we must prove $\pi_1(\mathrm{fold}(h)(\mathrm{in}_F(x))) = g(x, \mathrm{dmap}_F(\pi_1 \circ \mathrm{fold}(h), x))$. Since $\mathrm{fold}(h) \circ \mathrm{in}_F = h \circ F(\mathrm{fold}(h))$ by (1), the left hand side reduces to $\pi_1(h(F(\mathrm{fold}(h)(x))))$ which is equal to $g(\pi_0(\varphi(F(\mathrm{fold}(h)(x)))),$ $\pi_1(\varphi(F(\mathrm{fold}(h)(x)))))$ by the definition of h. It is thus enough to show that

(i) $\pi_0 \circ \varphi \circ F(\mathrm{fold}(h)) = \mathrm{id}$

(ii) $\pi_1 \circ \varphi \circ F(\mathrm{fold}(h)) = \mathrm{dmap}_F(\pi_1 \circ \mathrm{fold}(h))$.

Identity (i) is easily taken care of: by $(**_\square)$, $\pi_0 \circ \varphi = F(\pi_0)$, so that we have $\pi_0(\varphi(F(\mathrm{fold}(h)(x)))) = F(\pi_0)(F(\mathrm{fold}(h)(x))) = F(\pi_0 \circ \mathrm{fold}(h))(x) = F(\mathrm{id})(x) = x$.

For (ii), note that eta for Σ implies $\mathrm{fold}(h)(x) = \langle \pi_0(\mathrm{fold}(h)(x)), \pi_1(\mathrm{fold}(h)(x))\rangle = \langle x, \pi_1(\mathrm{fold}(h)(x))\rangle$ so that $\mathrm{fold}(h) = \widehat{f}$ for $f := \pi_1 \circ \mathrm{fold}(h)$. Hence, using $\pi_1 \circ \widehat{g} = g$ several times for different functions g, we have

$$\begin{aligned}
\mathrm{dmap}_F(\pi_1 \circ \mathrm{fold}(h)) &= \mathrm{dmap}_F(\pi_1 \circ \widehat{f}) \\
&= \mathrm{dmap}_F(f) \\
&= \pi_1 \circ \widehat{\mathrm{dmap}_F(f)} \\
&\overset{(*_{\mathrm{dmap}})}{=} \pi_1 \circ \varphi \circ F(\widehat{f}) = \pi_1 \circ \varphi \circ F(\mathrm{fold}(h))
\end{aligned}$$

which takes care of (ii), and we are done. □

Problem 4: Elim $\implies$ init (in Set)

Problem 4. Elim $\implies$ init.

Proof. Let $(X.f)$ be a F-algebra. We must construct $\mathrm{fold}(f) : \mu F \to X$ such that $\mathrm{fold}(f) \circ \mathrm{in}_F = f \circ F(\mathrm{fold}(f))$. We have

$$\begin{aligned}\mathrm{elim}_F : (P : \mu F \to \mathrm{Set}) \to \big(g : (x : F(\mu F)) \to \Box_F(P, x) \to P(\mathrm{in}_F(x))\big) \\ \to (x : \mu F) \to P(x)\end{aligned}$$

such that $\mathrm{elim}_F(P, g, \mathrm{in}_F(x)) = g(x, \mathrm{dmap}_F(\mathrm{elim}_F(P, g), x))$, so let us choose $P = K\ X$ to be constantly X and $g(x, y) = f(F(\pi_1)(\varphi^{-1}(\langle x, y\rangle))))$, and define $\mathrm{fold}(f) = \mathrm{elim}_F(K\ X, g)$. Then $\mathrm{fold}(f) : \mu F \to X$, and

$$\begin{aligned}\mathrm{fold}(f)(\mathrm{in}_F(x)) &= \mathrm{elim}_F(P, g, \mathrm{in}_F(x)) \\ &= g(x, \mathrm{dmap}_F(\mathrm{elim}_F(P, g), x)) \\ &= f(F(\pi_1)(\varphi^{-1}(\langle x, \mathrm{dmap}_F(\mathrm{elim}_F(P, g), x)\rangle)))) \\ &= f \circ F(\pi_1) \circ \varphi^{-1} \circ (\lambda x\ .\ \langle x, \mathrm{dmap}_F(\mathrm{fold}(f), x)\rangle)(x) \\ &= f \circ F(\pi_1) \circ \varphi^{-1} \circ \widehat{\mathrm{dmap}_F(\mathrm{fold}}(f))(x) \\ &\overset{(*_{\mathrm{dmap}})}{=} f \circ F(\pi_1) \circ \varphi^{-1} \circ \varphi \circ F(\widehat{\mathrm{fold}(f)})(x) \\ &= f \circ F(\pi_1 \circ \widehat{\mathrm{fold}(f)})(x) \\ &= f \circ F(\mathrm{fold}(f))(x),\end{aligned}$$

so $\mathrm{fold}(f) \circ \mathrm{in}_F = f \circ F(\mathrm{fold}(f))$ as required. ◻

Problem 5: $[\![\mathrm{elim}_T]\!]_{\mathbf{Fam}} \cong$ elim for simple ind.-ind.

If we consider "simple" induction-induction where the constructors for B do not refer to the constructors for A, the code for an inductive-inductive set is given by two functors

$$F : (A : \mathrm{Set}) \to (B : A \to \mathrm{Set}) \to \mathrm{Set}$$

$$G : (A : \mathrm{Set}) \to (B : A \to \mathrm{Set}) \to F(A, B) \to \mathrm{Set},$$

i.e. an endofunctor FG on Fam(Set). The formation and introduction rules we expect now says that there is $(A, B) : \mathrm{Fam(Set)}$ and $(c, d) : FG(A, B) \to_{\mathrm{Fam(Set)}} (A, B)$, i.e.

$$A : \mathrm{Set} \qquad\qquad B : A \to \mathrm{Set}$$

$$c : F(A, B) \to A \qquad\qquad d : (x : F(A, B)) \to G(A, B, x) \to B(c(x)).$$

Here are the types of the eliminators I would expect (whatever $\Box_F$, $\Box_G$, dmap_F, dmap_G are):

$$\begin{aligned}
\mathrm{elim}_F : (P : A \to \mathrm{Set}) \to (Q : (x : A) \to B(x) \to P(x) \to \mathrm{Set}) \to \\
(\overline{c} : (x : F(A, B)) \to \Box_F(P, Q, x) \to P(c(x))) \to \\
(\overline{d} : (x : F(A, B)) \to (y : G(A, B, x)) \to (\overline{x} : \Box_F(P, Q, x)) \\
\to \Box_G(P, Q, x, y, \overline{x}) \to Q(c(x), d(x, y), \overline{c}(x, \overline{x}))) \to \\
(x : A) \to P(x)
\end{aligned}$$

$$\begin{aligned}
\mathrm{elim}_G : (P : A \to \mathrm{Set}) \to (Q : (x : A) \to B(x) \to P(x) \to \mathrm{Set}) \to \\
(\overline{c} : (x : F(A, B)) \to \Box_F(P, Q, x) \to P(c(x))) \to \\
(\overline{d} : (x : F(A, B)) \to (y : G(A, B, x)) \to (\overline{x} : \Box_F(P, Q, x)) \\
\to \Box_G(P, Q, x, y, \overline{x}) \to Q(c(x), d(x, y), \overline{c}(x, \overline{x}))) \to \\
(x : A) \to (y : B(x)) \to Q(x, y, \mathrm{elim}_F(P, Q, \overline{c}, \overline{d}, x))
\end{aligned}$$

with computation rules

$$\mathrm{elim}_F(P, Q, \overline{c}, \overline{d}, c(x)) = \overline{c}(x, \mathrm{dmap}_F(f, g, x))$$

$$\mathrm{elim}_G(P, Q, \overline{c}, \overline{d}, c(x), d(x, y)) = \overline{d}(x, y, \mathrm{dmap}_F(f, g, x), \mathrm{dmap}_G(f, g, x, y))$$

where $f = \mathrm{elim}_F(P, Q, \overline{c}, \overline{d})$, $g = \mathrm{elim}_G(P, Q, \overline{c}, \overline{d})$.

Problem 5. The interpretation of ordinary elim_T in Fam(Set) is elim_F and elim_G.

Proof. Let us compile a small list of translations (of course, one should prove that this list is correct):

- $[\![A : \mathrm{Set}]\!]$ should be $A : \mathrm{Set}$, $B : A \to \mathrm{Set}$ – the new basic objects are families.
- $[\![x : A]\!]$ should be $x : A$ and $y : B(x)$.
- $[\![f : A \to A']\!]$ should be $f : A \to A'$, $g : (x : A) \to B(x) \to B'(f(x))$ – a family morphism.
- $[\![P : A \to \mathrm{Set}]\!]$ should be $P : A \to \mathrm{Set}$, $Q : (a : A) \to B(a) \to P(a) \to \mathrm{Set}$ since we want $[\![P]\!] : [\![A]\!][\![\to]\!][\![\mathrm{Set}]\!] = (A, B) \to_{\mathrm{Fam(Set)}} \mathrm{Fam(Set)}$.
- $[\![x : A \vdash p : P(x)]\!]$ should be $x : A \vdash p : P(x)$ (or should we allow $x : A, y : B(x) \vdash p : P(x)$ here?) and $x : A, y : B(x) \vdash q : Q(x, y, p)$.
- $[\![f : (x : A) \to P(x)]\!]$ should be $f : (x : A) \to P(x)$ and $g : (x : A) \to$ (y : B(x)) $\to Q(x, y, f(x))$.

- $[\![\Sigma\ A\ P]\!]$ should be $\Sigma\ A\ P : \text{Set}, (\lambda\ \langle a,p\rangle\ .\ \Sigma\ b{:}B(a).Q(a,b,p)))$ – this is a a set and a family over it.
- $[\![\widehat{f}]\!]$ should, given $f : (x : A) \to P(x)$ and $g : (x : A) \to (y : B(x)) \to Q(x, y, f(x))$ [i.e. $[\![f : (x : A) \to P(x)]\!]$] be a function $\widehat{f} : A \to \Sigma\ A\ P$, $\widehat{f}(a) = \langle a, f(a)\rangle$, and a function $\widehat{g} : (a : A) \to B(a) \to \Sigma\, b{:}B(a).Q(a, b, f(a))$, $\widehat{g}(a, b) = \langle b, g(a, b)\rangle$.

We do not need all of these yet, but they should be useful in the future.

Now, we can start to deconstruct elim_F and elim_G. We already noticed that F and G form an endofunctor FG on Fam(Set). We also see that the arguments $P : A \to \text{Set}$ and $Q : (a : A) \to B(a) \to P(a) \to \text{Set}$ are the interpretation of some $P : A \to \text{Set}$, i.e. $[\![P]\!] = (P, Q)$.

Moving on, we notice that $\Box_F(P, Q) : F(A, B) \to \text{Set}$ and $\Box_G(P, Q) : (x : F(A, B)) \to G(A, B, x) \to \Box_F(P, Q, x) \to \text{Set}$ is a family over $(F(A, B), G(A, B))$, i.e. the interpretation of some $\Box_{FG} : FG(AB) \to \text{Set}$.

With some currying, we see that $\overline{c} : (x : F(A, B)) \to \Box_F(P, Q, x) \to P(c(x))$ and $\overline{d} : (x : F(A, B)) \to (y : G(A, B, x)) \to (\overline{x} : \Box_F(P, Q, x)) \to \Box_G(P, Q, x, y, \overline{x}) \to Q(c(x), d(x, y), \overline{c}(x, \overline{x}))$ just form a family morphism from $xy : \Sigma\ (F(A, B), G(A, B))\ (\Box_F(P, Q), \Box_G(P, Q))$ to $(P, Q)((c, d)([\![\pi_0]\!](xy)))$, i.e. they are the interpretation of some $g : (x : \Sigma\ FG(AB)\ (\Box_{FG}\ P)) \to P(\text{in}_{FG}(\pi_0(x)))$.

Finally, we see that $(\text{elim}_F(P, Q, \overline{c}, \overline{d}), \text{elim}_G(P, Q, \overline{c}, \overline{d})$ is a dependent morphism from $(xy : (A, B))$ to $(P, Q)(xy)$.

All in all, $(\text{elim}_F, \text{elim}_G)$ is thus the translation of

$$\begin{aligned}\text{elim}_{FG} : (P : A \to \text{Set}) \to \big(g : (x : \Sigma\ (FG(A))\ \Box_{FG}(P)) \to P(\text{in}_{FG}(\pi_0(x)))\big) \\ \to (x : A) \to P(x)\end{aligned}$$

which is the ususal eliminator. The computation rules also seem to be exactly the interpretatino of the usual computation rules.

$\Box$