Generalising $\Box_F$ and dmap_F for non-endofunctors

Fredrik Nordvall Forsberg

December 7, 2010

1 The standard situation

In the ordinary situation, we have an endofunctor $F : \mathrm{Set} \to \mathrm{Set}$ representing an inductive definition, i.e. we have a set μF and a constructor $c : F(\mu F) \to \mu F$. The eliminator has type

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

and computation rule

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

Here,

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

gives the induction hypothesis (it is called 'IH' in the induction-recursion papers) and

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

takes care of the recursive calls (in the IR papers, it is called mapIH). Both $\Box_F$ and dmap_F can be explicitly defined:

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

where $\widehat{g} : A \to \Sigma\ A\ P$ is defined by $\widehat{g}(z) = \langle z, g(z)\rangle$. Note that $\pi_0 \circ \widehat{g} = \mathrm{id}$, so that

$$F(\pi_0)(F(\widehat{g})(x)) = F(\pi_0 \circ \widehat{g})(x)) = F(\mathrm{id})(x) = \mathrm{id}(x) = x$$

which shows that indeed $\mathrm{dmap}_F(A, P, g, x) : \Box_F(A, P, x)$. One can show that this definition of $\Box_F$ is naturally isomorphic to the definition in the IR papers (see Problem 2 in `problems.pdf`).

However, we prefer to just use some abstract properties of $\square_F$ and dmap_F, namely

$$F(\Sigma\ A\ B) \cong \Sigma\ F(A)\ \square_F(A, B) \tag{$*_\square$}$$

and this isomorphism φ satisfies

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

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

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

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

2 Generalising $\square_F$ and dmap_F for non-endofunctors

Let us now try to generalise things, first by moving away from the category Set to an arbitrary category $\mathbb{C}$ (with some structure, of course), and then by replacing the endofunctor $F : \mathbb{C} \to \mathbb{C}$ with just a functor $F : \mathbb{C} \to \mathbb{D}$.

2.1 The category $[\![\mathbf{Fam}]\!](\mathbb{C})$ of interpretations of families in $\mathbb{C}$

Let us for the moment assume that the category $\mathbb{C}$ is a model of type theory. I want to define a category $[\![\mathrm{Fam}]\!](\mathbb{C})$ of "interpretations of families in $\mathbb{C}$". It has as objects an interpretation of '$A : \mathrm{Set}, B : A \to \mathrm{Set}$' in $\mathbb{C}$, and a morphism from $[\![A, B]\!]$ to $[\![A', B']\!]$ is an interpretation of '$f : A \to A', g : (x : A) \to B(x) \to B'(f(x))$'.

Example 1 ($\mathbb{C} = \mathrm{Set}$)**.** For $\mathbb{C} = \mathrm{Set}$, we recover the usual category Fam(Set) of families of sets, i.e. we have objects pairs (A, B) where $A : \mathrm{Set}$ and $B : A \to \mathrm{Set}$, and a morphism $(A, B) \to (A', B')$ is a pair $f : A \to A'$, $g : (x : A) \to B(x) \to B'(f(x))$.

Example 2 ($\mathbb{C} = \mathrm{Fam(Set)}$)**.** Unless I am mistaken, the objects in $[\![\mathrm{Fam}]\!](\mathrm{Fam(Set)})$ are tuples $((A_0, A_1), (B_0, B_1))$ where

$$\begin{aligned} &A_0 : \mathrm{Set}, \\ &A_1 : A_0 \to \mathrm{Set}, \\ &B_0 : A_0 \to \mathrm{Set}, \\ &B_1 : (x : A_0) \to A_1(x) \to B_0(x) \to \mathrm{Set}. \end{aligned}$$

A morphism from $((A_0, A_1), (B_0, B_1))$ to $((A'_0, A'_1), (B'_0, B'_1))$ consists of

$$
\begin{aligned}
&f_0 : A_0 \to A'_0, \\
&f_1 : (x : A_0) \to A_1(x) \to A'_1(f_0(x)), \\
&g_0 : (x : A_0) \to B_0(x) \to B'_0(f_0(x)) \\
&g_1 : (x : A_0) \to (y : A_1(x)) \to (z : B_0(x)) \to B_1(x, y, z) \\
&\qquad\qquad\qquad\qquad\qquad\qquad \to B'_1(f_0(x), f_1(x, y), g_0(x, z)).
\end{aligned}
$$

Question 1. Is there a better way to present $[\![\mathrm{Fam}]\!](\mathbb{C})$?

2.2 Σ as a functor from $[\![\mathbf{Fam}]\!](\mathbb{C})$ to $\mathbb{C}$

Now, I notice that the sigma constructor takes a family (A, B) of sets and gives me a set $\Sigma\ A\ B$ back. It is also functorial, since if I have a morphism $(f, g) : (A, B) \to (A', B')$, then I can construct a morphism $[f, g] : \Sigma\ A\ B \to \Sigma\ A'\ B'$ by defining

$$[f, g]\langle a, b\rangle = \langle f(a), g(a, b)\rangle.$$

In the general case, I would thus like to generalise Σ to a functor

$$\Sigma_{\mathbb{C}} : [\![\mathrm{Fam}]\!](\mathbb{C}) \to \mathbb{C}$$

together with some kind of (interpretation of) projection morphisms $\pi_0 : \Sigma_{\mathbb{C}}(A, B) \to A$, $\pi_1 : (x : \Sigma_{\mathbb{C}}(A, B)) \to B(\pi_0(x))$.

Question 2. I probably would like $\Sigma_{\mathbb{C}}$ to be the interpretation of Σ in $\mathbb{C}$. For morphisms, this would mean that I want $\Sigma_{\mathbb{C}}(f, g)$ to be the interpretation of $[f, g]$?

2.3 $\Box$ is a functor $\Box' : \mathbf{Set}^{\mathbf{Set}} \to \mathbf{Fam}(\mathbf{Set})^{\mathbf{Fam}(\mathbf{Set})}$

Looking at the type

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

of $\Box_F$ again, we see that we can write this as

$$\Box' : \mathrm{Set}^{\mathrm{Set}} \to \mathrm{Fam}(\mathrm{Set})^{\mathrm{Fam}(\mathrm{Set})}$$

if we define $\Box'(F) = \lambda(A, B).\ (F(A), \Box_F(A, B))$. For $\Box'(F)$ to be a functor Fam(Set) to Fam(Set), it must also act on morphisms. Suppose $(f, g) : (A, B) \to (A', B')$. We must find a morphism from $\Box'(F)(A, B)$ to $\Box'(F)(A', B')$, i.e. from

$$(F(A), \{y : F(\Sigma\ A\ B) \mid F(\pi_0)(y) = x\})$$

to

$$(F(A'), \{y : F(\Sigma\ A'\ B') \mid F(\pi_0)(y) = x\}).$$

But $f : A \to A'$, so $m = F(f) : F(A) \to F(A')$ can be our first component. Now we need to define

$$\begin{aligned} n : (x : F(A)) \to \{y : F(\Sigma\ A\ B) \mid F(\pi_0)(y) = x\} \\ \to \{z : F(\Sigma\ A'\ B') \mid F(\pi_0)(z) = F(f)(x)\} \end{aligned}$$

Let $n(x) = F([f, g])$. We need to prove that $F(\pi_0)(F([f, g])(y)) = F(f)(x)$ for $y : F(\Sigma\ A\ B)$ such that $F(\pi_0)(y) = x$. But $\pi_0 \circ [f, g] = f \circ \pi_0$, so

$$F(\pi_0)(F([f, g])(y)) = F(\pi_0 \circ [f, g])(y) = F(f \circ \pi_0)(y) = F(f)(F(\pi_0)(y)) = F(f)(x).$$

Thus we can take $\Box'(F)(f, g) = (n, m) = (F(f), \lambda x.\ F([f, g]))$. This shows that $\Box'(F) : \mathrm{Fam}(\mathrm{Set})^{\mathrm{Fam}(\mathrm{Set})}$ if $F : \mathrm{Set} \to \mathrm{Set}$ is a functor.

However, we want more! We want $\Box'$ to be a functor $\Box' : \mathrm{Set}^{\mathrm{Set}} \to \mathrm{Fam}(\mathrm{Set})^{\mathrm{Fam}(\mathrm{Set})}$. We have defined the object part of $\Box'$, what is the morphism part? Given a natural transformation $\eta : F \dot{\to} G$, we have to construct a natural transformation $\Box'(\eta) : \Box'(F) \dot{\to} \Box'(G)$. The component of $\Box'(\eta)$ at (A, B) consists of a function $f : F(A) \to G(A)$ and a function $g : (x : F(A)) \to \Box_F(A, B, x) \to \Box_G(A, B, f(x))$. For f, we can choose $f = \eta_A$. Choose $g(x) = \eta_{\Sigma\ A\ B}$. Certainly $\eta_{\Sigma AB} : F(\Sigma\ A\ B) \to G(\Sigma\ A\ B)$, but do we have $G(\pi_0)(\eta_{\Sigma AB}(y)) = \eta_a(x)$, given that $F(\pi_0)(y) = x$? Yes, because substituting $x = F(\pi_0)(y)$, we end up with the equation $G(\pi_0)(\eta_{\Sigma AB}(y)) = \eta_a(F(\pi_0)(y))$, which is exactly the naturality condition!

$$\begin{tikzcd} F(\Sigma\ A\ B) \arrow[r, "\eta_{\Sigma AB}"] \arrow[d, "F(\pi_0)"'] & G(\Sigma\ A\ B) \arrow[d, "G(\pi_0)"] \\ F(A) \arrow[r, "\eta_A"'] & G(A) \end{tikzcd}$$

Hence we can define $\Box'(\eta)_{(A,B)} = (\eta_A, \lambda x.\ \eta_{\Sigma AB})$. The naturality of $\Box'(\eta)$ follows directly from the naturality of η.

2.4 A generalised $\Box_F$ for F not an endofunctor

Assume that $\mathbb{C}$ and $\mathbb{D}$ are categories such that $[\![\mathrm{Fam}]\!](\mathbb{C})$ and $[\![\mathrm{Fam}]\!](\mathbb{D})$ make sense. (Then $\Sigma_{\mathbb{C}}$ and $\Sigma_{\mathbb{D}}$ should make sense as well.) We can then generalise $\Box_F$ to a functor

$$\Box : \mathbb{D}^{\mathbb{C}} \to [\![\mathrm{Fam}]\!](\mathbb{C}) \to [\![\mathrm{Fam}]\!](\mathbb{C})$$

such that there is a natural isomorphism

$$\eta : F \circ \Sigma \dot{\to} \Sigma\ \circ \Box \qquad (*_\Box)$$

satisfying

$$\begin{tikzcd} F(\Sigma\ A\ B) \arrow[r, "\eta_{(A,B)}"] \arrow[d, "F(\pi_0)"'] & \Sigma\ F(A)\ \Box_F(A, B) \arrow[dl, "\pi_0"] \\ F(A) & \end{tikzcd} \qquad (**_\Box)$$

i.e. $\pi_0 \circ \varphi = F(\pi_0)$.

...

3 Comparision with Fibrational Induction Rules for Initial Algebras