Type Theory with Paige North 7/10

Jason Schuchardt

July 17, 2019

1 Identity terms in Σ-types

Last time, we constructed

$$f : \prod_{s,s' : \sum_{x:B} E(x)} \left[\mathrm{Id}(s,s') \to \sum_{p:\mathrm{Id}(\pi_1 s, \pi_1 s')} \mathrm{Id}(p_* \pi_2 s, \pi_2 s') \right].$$

By Σ-elimination, we can assume that $s = (b,e), s' = (b',e')$ are canonical. By Id-elimination, it suffices to define the function on canonical terms of the identity type, $\mathrm{refl}_{(e,b)}$. Then constructing this function is equivalent to producing

$$\prod_{b:B, e:E(b)} \sum_{p:\mathrm{Id}(b,b)} \mathrm{Id}(p_* e, e).$$

Undoing the Π, we want to produce

$$b : B, e : E(b) \vdash \sum_{p:\mathrm{Id}(b,b)} \mathrm{Id}(p_* e, e).$$

I.e., we want to produce a pair (p,q), with $p : \mathrm{Id}(b,b)$, $q : \mathrm{Id}(p_* e, e)$. We can take $p = \mathrm{refl}_b$, then $p_* e = e$ by definition, so we can produce $q = \mathrm{refl}_e : \mathrm{Id}(e,e) = \mathrm{Id}(p_* e, e)$. Thus $(\mathrm{refl}_b, \mathrm{refl}_e)$ is in the sum type.

Assuming terms are canonical is often called induction, since in the natural numbers, defining a function out of $\mathbb{N}$ by specifying where the canonical terms $0 : \mathbb{N}$, $sn : \mathbb{N}$ go is induction.

Now we want to construct maps in the other direction:

$$g : \prod_{s,s' : \sum_{x:B} E(x)} \left[\left(\sum_{p:\mathrm{Id}(\pi_1 s, \pi_1 s')} \mathrm{Id}(p_* \pi_2 s, \pi_2 s') \right) \to \mathrm{Id}(s,s') \right]$$

Want

$$s, s' : \sum_{x:B} E(x), t : \sum_{p:\mathrm{Id}(\pi_1 s, \pi_1 s')} \mathrm{Id}(p_* \pi_2 s, \pi_2 s') \vdash ? : \mathrm{Id}(s,s').$$

Induct on t, so assume we have (p,q), with $p : \mathrm{Id}(\pi_1 s, \pi_1 s')$, $q : \mathrm{Id}(p_* \pi_2 s, \pi_2 s')$. It suffices to construct

$$s, s' : \sum_{x:B} E(x), p : \mathrm{Id}(\pi_1 s, \pi_1 s'), q : \mathrm{Id}(p_* \pi_2 s, \pi_2 s') \vdash ? : \mathrm{Id}(s,s')$$

Then inducting on s, s', it suffices to prove

$$b, b' : B, e : E(b), e' : E(b'), p : \mathrm{Id}(b,b'), q : \mathrm{Id}(p_* e, e') \vdash ? : \mathrm{Id}((b,e),(b',e')).$$

Can't induct on q, since p_*e is not free essentially. However, we can induct on p, so it suffices to prove

$$b : B, e, e' : E(b), q : \mathrm{Id}(e, e') \vdash ? : \mathrm{Id}((b, e), (b, e')).$$

Now we may induct on q. It now suffices to construct

$$b : B, e : E(b) \vdash ? : \mathrm{Id}((b, e), (b, e)).$$

However, we can now construct this term. It is $\mathrm{refl}_{(b,e)}$.

Question: If we were to write out the logic using the rules we've gotten, the flow of logic would go from bottom to top? Yes. But day to day, this is how we write proofs in type theory. But we could write out the term. If we wrote out the resulting term, we would get something like

$$\lambda s, s'.\lambda t.j_{j_{r_{b,e}}}(s, s', t)$$

To show that we have a quasiequivalence term in

$$\prod_{s,s':\sum_{x:B} E(x)} (g_{s,s'} \circ f_{s,s'} \sim 1)$$

we expand out the type, to get that we need to construct a term of type

$$\prod_{s,s':\sum_{x:B} E(x)} \prod_{u:\mathrm{Id}(s,s')} \mathrm{Id}(g(f(u)), u).$$

Induct on u. We need

$$\prod_{s:\sum_{x:B} E(x)} \mathrm{Id}(g(f(\mathrm{refl}_s)), \mathrm{refl}_s).$$

Recall that $f(\mathrm{refl}_s) = (\mathrm{refl}_{\pi_1 s}, \mathrm{refl}_{\pi_2 s})$, and $g(\mathrm{refl}_b, \mathrm{refl}_e) = \mathrm{refl}_s$. Thus $g(f(\mathrm{refl}_s)) = \mathrm{refl}_s$. Then

$$\lambda s.\mathrm{refl}_{\mathrm{refl}_s} : \prod_{s:\sum_{x:B} E(x)} \mathrm{Id}(\mathrm{refl}_s, \mathrm{refl}_s).$$

Now we want to go the other way, and construct

$$\prod_{t,t':\Sigma} (f_{t,t'} \circ g_{t,t'} \sim 1),$$

where Σ is the sum type in the domain of g. Unfold the homotopy ($\sim$) type

$$\prod_{t,t':\Sigma} \prod_{t:\Sigma} \mathrm{Id}(f(g(t)), t)$$

Induct on s, s', t to get that we need

$$\prod_{b,b':B} \prod_{e:E(b)} \prod_{e':E(b')} \prod_{p:\mathrm{Id}(b,b')} \prod_{q:\mathrm{Id}(p_*e,e')} \mathrm{Id}(f(g(p, q)), (p, q)).$$

Induct on p, so we now need

$$\prod_{b:B} \prod_{e,e':E(b)} \prod_{q:\mathrm{Id}(e,e')} \mathrm{Id}(f(g(\mathrm{refl}_b, q)), (\mathrm{refl}_b, q)).$$

Inducting on q, we now want

$$\prod_{b:B}\prod_{e:E(b)} \mathrm{Id}(f(g(\mathrm{refl}_b, \mathrm{refl}_e)), (\mathrm{refl}_b, \mathrm{refl}_e)).$$

As before $g(\mathrm{refl}_b, \mathrm{refl}_e) = \mathrm{refl}_{(b,e)}$, and $f(\mathrm{refl}_{(b,e)}) = (\mathrm{refl}_b, \mathrm{refl}_e)$. Thus this last type contains the term

$$\mathrm{refl}_{(\mathrm{refl}_b, \mathrm{refl}_e)}$$

We have now constructed a term in

$$\prod_{s,s':\sum_{x:B} E(x)} \mathrm{Id}(s, s') \simeq \prod_{p:\mathrm{Id}(\pi_1 s, \pi_1 s')} \mathrm{Id}(p_* \pi_2 s, \pi_2 s').$$

2 Identity terms in Π-types

Given $f, g : \prod_{x:B} E(x) \vdash \mathrm{Id}(f, g)$. We have both $\mathrm{Id}(f, g) : \mathcal{U}$ and $f \sim g : \mathcal{U}$. We'd like to say that these are equivalent types.

However, we can't prove this. There are models of type theory for which this is false. So we can't prove this. Thus we postulate this, since it's reasonable to assume. We assume that we have a term

$$\mathrm{FunExt} : \mathrm{Id}(f, g) \simeq (f \sim g).$$

This is an axiom.