HoTT reading notes

Sixuan Lou

August 9, 2017

Lemma 0.1. *We may inhabit*

$$transport : \prod_{(C:\prod_{(x:A)} \mathcal{U})} \prod_{(x,y:A)} \prod_{(p:x=_A y)} C(x) \to C(y)$$

.

Proof. Recall, we have

$$ind_{=_A} : \prod_{(C:\prod_{(x,y:A)}(x=_A y)\to\mathcal{U})} \left(\prod_{(x:A)} C(x,x,refl_x)\right) \to \prod_{(x,y:A)} \prod_{(p:x=_A y)} C(x,y,p)$$

hence,

$$transport(C,x,y,p) :\equiv ind_{=_A}(\lambda x.\lambda y.\lambda \_.C(x) \to C(y), \lambda x.id_{C(x)}, x, y, p)$$

$\square$

Problem 1.1. *Given function $f : A \to B$ and $g : B \to C$, define composite $g \circ f : A \to C$ and show $h \circ (g \circ f) \equiv (h \circ g) \circ f$.*

Proof.

$$\begin{aligned}
(g \circ f)(x) &:\equiv g(f(x)) \\
h \circ (g \circ f) &\equiv_\eta \lambda x.(h \circ (g \circ f))(x) \equiv_{\beta,\delta} \lambda x.(h((g \circ f)(x))) \\
&\equiv_{\beta,\delta} \lambda x.(h(g(f(x)))) \equiv_{\beta,\delta^{-1}} \lambda x.(h \circ g)(f(x)) \\
&\equiv_{\beta,\delta^{-1}} \lambda x.((h \circ g) \circ f)(x) \equiv_{\eta^{-1}} (h \circ g) \circ f
\end{aligned}$$

$\square$

Problem 1.4. *Define the recursor $rec_{\mathbb{N}}$ in turns of the iterator $iter_{\mathbb{N}}$ and show they are propositionally equal with the induction principle $ind_{\mathbb{N}}$.*

Proof. Put $f :\equiv \lambda pr.(s(\pi_1(pr)), c_s(\pi_1(pr), \pi_2(pr)))$

$$\begin{aligned}
rec_{\mathbb{N}} &: \prod_{C:\mathcal{U}} C \to (\mathbb{N} \to C \to C) \to \mathbb{N} \to C \\
rec_{\mathbb{N}}(C, c_0, c_s, n) &:\equiv \pi_2(iter_{\mathbb{N}}(\mathbb{N} \times C, (0, c_0), f, n))
\end{aligned}$$

Let $\widetilde{rec_{\mathbb{N}}}$ denote the regular recursor, let $D :\equiv \lambda n.rec_{\mathbb{N}}(C, c_0, c_s, n) =_{\mathbb{N}} \widetilde{rec_{\mathbb{N}}}(C, c_0, c_s, n)$. We want to construct a term of type

$$\prod_{(C:\mathcal{U})} \prod_{(c_0:C)} \prod_{(c_s:\mathbb{N}\to C\to C)} \prod_{(n:\mathbb{N})} D(n)$$

We first construct a term of type

$$\prod_{(C:\mathcal{U})}\prod_{(c_0:C)}\prod_{(c_s:\mathbb{N}\to C\to C)}\prod_{(n:\mathbb{N})} D'(n)$$

where, $D' :\equiv \lambda n.\pi_1(iter_{\mathbb{N}}(\mathbb{N}\times C,(0,c_0),f,n)) =_{\mathbb{N}} n$. This is simple, an inhabitant is simply:

$$\begin{aligned}
&ind_{\mathbb{N}}(D', refl_0,\\
&\quad \lambda n.\lambda(pf : D'(n)).\\
&\qquad ind_{=_{\mathbb{N}}}(E, refl_{s(n)}, s(\pi_1(iter_{\mathbb{N}}(\mathbb{N}\times C, c_0, f, n))), s(n), pf))
\end{aligned}$$

where $E :\equiv \lambda x.\lambda y.s(x) =_{\mathbb{N}} s(y)$. That term is kind of long, we name it as foo. Now, with foo in hand, we are able to complete our task. The inhabitant is too long to write down, but there is the skeleton:

$$ind_{\mathbb{N}}(D, refl_{c_0}, \lambda n.\lambda(pf : D(n)).g(n, pf))$$

Where g has type:

$$\begin{aligned}
&\prod_{n:\mathbb{N}} D(n) \to D(s(n))\\
&\equiv \prod_{n:\mathbb{N}} rec_{\mathbb{N}}(C,c_0,c_s,n) =_{\mathbb{N}} \widetilde{rec_{\mathbb{N}}}(C,c_0,c_s,n)\\
&\qquad \to rec_{\mathbb{N}}(C,c_0,c_s,s(n)) =_{\mathbb{N}} \widetilde{rec_{\mathbb{N}}}(C,c_0,c_s,s(n))\\
&\equiv \prod_{n:\mathbb{N}} rec_{\mathbb{N}}(C,c_0,c_s,n) =_{\mathbb{N}} \widetilde{rec_{\mathbb{N}}}(C,c_0,c_s,n)\\
&\qquad \to \pi_2(iter_{\mathbb{N}}(\mathbb{N}\times C,(0,c_0),f,s(n))) =_{\mathbb{N}} \widetilde{rec_{\mathbb{N}}}(C,c_0,c_s,s(n))\\
&\equiv \prod_{n:\mathbb{N}} rec_{\mathbb{N}}(C,c_0,c_s,n) =_{\mathbb{N}} \widetilde{rec_{\mathbb{N}}}(C,c_0,c_s,n)\\
&\qquad \to \pi_2(f(iter_{\mathbb{N}}(\mathbb{N}\times C,(0,c_0),f,n))) =_{\mathbb{N}} \widetilde{rec_{\mathbb{N}}}(C,c_0,c_s,s(n))\\
&\equiv \prod_{n:\mathbb{N}} rec_{\mathbb{N}}(C,c_0,c_s,n) =_{\mathbb{N}} \widetilde{rec_{\mathbb{N}}}(C,c_0,c_s,n)\\
&\qquad \to c_s(\pi_1(iter_{\mathbb{N}}(\mathbb{N}\times C,(0,c_0),f,n)), \pi_2(iter_{\mathbb{N}}(\mathbb{N}\times C,(0,c_0),f,n))) =_{\mathbb{N}} c_s(n, \widetilde{rec_{\mathbb{N}}}(C,c_0,c_s,n))
\end{aligned}$$

Applying identity elimination twice using pf and foo will replace $\pi_1(iter_{\mathbb{N}}(\mathbb{N}\times C,(0,c_0),f,n))$ with n and $\pi_2(iter_{\mathbb{N}}(\mathbb{N}\times C,(0,c_0),f,n))$ with $\widetilde{rec_{\mathbb{N}}}(C,c_0,c_s,n)$. We have the result. $\square$

Problem 1.5. *Show that if we define $A+B :\equiv \Sigma_{x:2} rec_2(\mathcal{U},A,B,x)$, then we can give a definition of ind_{A+B} such that the definitional equalities stated in 1.7 hold.*

Proof. We define $inl(a) :\equiv (0,a)$ and $inr(b) :\equiv (1,b)$. By the definition of the induction principle of Σ-type,

$$\begin{aligned}
&ind_{A+B} \equiv ind_{\Sigma_{(x:2)} rec_2(\mathcal{U},A,B,x)}\\
&\quad : \prod_{(C:\Sigma_{(x:2)} rec_2(\mathcal{U},A,B,x)\to\mathcal{U})}\left(\prod_{(x:2)}\prod_{(e:rec_2(\mathcal{U},A,B,x))} C(x,e)\right) \to \prod_{(pr:\Sigma_{(x:2)} rec_2(\mathcal{U},A,B,x))} C(pr)\\
&ind_{A+B}(C,f,(a,b)) :\equiv f(a)(b)
\end{aligned}$$

Hence, if we define the regular induction principle in turns of this weird definition by:

$$\begin{aligned}
\widetilde{ind_{A+B}}(C,g_0,g_1,inl(a)) &:\equiv ind_{A+B}(C, \lambda(_ : 2).g_0, (0,a))\\
\widetilde{ind_{A+B}}(C,g_0,g_1,inl(b)) &:\equiv ind_{A+B}(C, \lambda(_ : 2).g_0, (1,b))
\end{aligned}$$

It's easy to verify everything works out. $\square$

Problem 1.6. *Show that if we define* $A \times B :\equiv \prod_{(x:2)} rec_2(\mathcal{U}, A, B, x)$*, then we can give a definition of* $ind_{A\times B}$ *for which the definitional equalities stated in 1.5 hold propositionally.*

Proof. We define $(a, b) :\equiv \lambda x.rec_2(rec_2(\mathcal{U}, A, B, x), a, b, x)$. By the definition of the induction principle of the $\prod$-type, $ind_{A\times B}$ is simply the application of a lambda with type $\prod_{(x:2)} rec_2(\mathcal{U}, A, B, x)$ to an argument of type 2. More precisely,

$$ap : \left(\prod_{(x:2)} rec_2(\mathcal{U}, A, B, x)\right) \to 2 \to rec_2(\mathcal{U}, A, B, x)$$

To recover the usual induction principle, we first attempt to define the following term:

$$\widetilde{ind_{A\times B}}' : \prod_{C:A\times B\to\mathcal{U}} \left(\prod_{(x:A)} \prod_{(y:B)} C(x, y)\right) \to \prod_{(pr:A\times B)} C(pr(0), pr(1))$$
$$\widetilde{ind_{A\times B}}'(C, f, g) :\equiv f(g(0))(g(1))$$

We however, want the body to have type $C(pr)$, not $C(pr(0), pr(1))$, to do that, we need to inhabit the propositional uniqueness principle for our new product:

$$uniq_{A\times B} : \prod_{p:A\times B} (p(0), p(1)) =_{A\times B} p$$
$$uniq_{A\times B} :\equiv \lambda p.funext(\lambda(x : 2).refl_{rec_2(rec_2(\mathcal{U},A,B,x),p(0),p(1),x)})$$

Finally,

$$\widetilde{ind_{A\times B}} : \prod_{C:A\times B\to\mathcal{U}} \left(\prod_{(x:A)} \prod_{(y:B)} C(x, y)\right) \to \prod_{(pr:A\times B)} C(pr)$$
$$\widetilde{ind_{A\times B}}(C, f, g) :\equiv transport(C, (g(0), g(1)), g, uniq_{A\times B}(g))(\widetilde{ind_{A\times B}}'(C, f, g))$$

$\square$