Type Theory with Paige North 7/22

Jason Schuchardt

July 22, 2019

Last time we talked about contractibility (level 0), and we defined

$$\mathrm{isContr}(T) := \sum_{t:T} \prod_{s:T} s = t.$$

1 Propositions (h level 1)

This time we will talk about propositions.

Definition 1.1.

$$T : U \vdash \mathrm{isProp}(T) : U,$$

where

$$\mathrm{isProp}(T) := \prod_{s,t:T} \mathrm{isContr}(s = t).$$

This roughly means that T is a proposition if and only if $T \simeq \bot$ or $T \simeq \top$. (This isn't strictly true. We would need law of excluded middle to prove it.)

Proposition 1.1. *If $T \simeq \bot$, then T is a proposition. If $t : T$ and T is a proposition, then $T \simeq \top$.*

Proof. We begin by showing $\bot$ is a proposition.

$$\mathrm{isProp}(\bot) = \prod_{x,y:\bot} \mathrm{isContr}(x = y).$$

Then by $\bot$-elimination, there is always a dependent function out of bottom into any type. Thus isProp($\bot$) is true.

Suppose we have $t : T, p : \mathrm{isProp}(T)$. We want to construct a pair of a term $t : T$, and an element of $\prod_{s:T} s = t$. We already have a $t : T$, and we have

$$p : \mathrm{isProp}(T) := \prod_{s,t:T} \mathrm{isContr}(s = t).$$

Then

$$pt : \prod_{s:T} \mathrm{isContr}(s = t).$$

Then

$$\lambda s.\pi_1(pts) : \prod_{s:T} s = t$$

is our desired term. ∎

Why do we consider propositions?

Consider a dependent type, $b : B \vdash E(b)$. For example, $n : \mathbb{N} \vdash \text{isEven}(n) : U$. Then we might want to define the 'subtype' of B consisting of all $b : B$ such that $E(b)$ is inhabited. In set theory, we can just form

$$\{b \in B \mid E(b)\}.$$

In type theory on the other hand, the closest thing we can produce is the Σ-type:

$$\sum_{b:B} E(b).$$

Then we might define the type of even natural numbers to be

$$E := \sum_{n:\mathbb{N}} \text{isEven}(n).$$

Now we might run into a problem if our predicate is very complicated, for example, we might have many proofs of $E(b)$, then

$$\sum_{b:B} E(b)$$

might be very complicated, and not behave like a subtype.

However if each of $E(b)$ is a proposition, then it will behave like a subtype should behave. For example,

$$n : \mathbb{N} \vdash \text{isEven}(n) := \sum_{m:\mathbb{N}} 2m = n.$$

is a proposition.

Proposition 1.2. *Consider $b : B \vdash E(b)$, together with*

$$b : B \vdash p(b) : \text{isProp}(E(b)).$$

If we have

$$s, t : \sum_{b:B} E(b)$$

such that $\pi_1 s = \pi_1 t$, then $s = t$.

Proof. Consider $q : \pi_1 s = \pi_1 t$. We know

$$s = t \simeq \sum_{a:\pi_1 s = \pi_1 t} a_* \pi_2 s = \pi_2 t$$

We already have $q : \pi_1 s = \pi_1 t$. We now just need to find something of type $q_* \pi_2 s = \pi_2 t$. This is an identity type between terms of $E(\pi_1 t)$. However, we know that $E(\pi_1 t)$ is a proposition, so $q_* \pi_2 s = \pi_2 t$ is contractible. It is therefore inhabited by some term c. Then (q, c) is a term in our sum type. ■

Side note, once again, recall that all English statements are just translations of type theoretical statements. The proposition above is the following statement

$$B : U, E : B \to U, p : \prod_{b:B} \text{isProp}(E(b)), s, t : \sum_{b:B} E(b), q : \pi_1 s = \pi_1 t \vdash ? : s = t : U.$$

Proposition 1.3. isEquiv(f)*,* isContr(T)*,* isProp(T) *are always propositions.* isQEquiv *is not a proposition generally. (This is why we say that equivalences are better behaved than quasiequivalences.)*

Example 1.1.

$$\sum_{f:A\to B} \text{isEquiv}(f) \quad \text{“} \subseteq \text{”} \quad A \to B$$

2 Sets (h level 2)

Definition 2.1.

$$T : U \vdash \mathrm{isSet}(T) : U,$$

where

$$\mathrm{isSet}(T) := \prod_{s,t:T} \mathrm{isProp}(s = t)$$

Picture: We think of T as a space, with terms r, s, t being points of the space. Then we might have $p, q : s \to t$ paths. The space of paths is either empty or contractible. So if p and q are both paths, then we have a term of the identity type $p = q$.

Example 2.1 (Groups)**.**

$$\mathbb{G}\mathrm{roup} := \sum_{G:U} \sum_{p:\mathrm{isSet}(G)} \sum_{m:G\times G\to G} \sum_{e:G} \sum_{i:G\to G}$$
$$\left(\prod_{a,b,c:G} m(a, m(b,c)) = m(m(a,b),c) \right)$$
$$\times \left(\prod_{g:G} m(g,e) = g \times m(e,g) = g \right)$$
$$\times \left(\prod_{g:G} m(g,ig) = e \times m(ig,g) = e \right)$$

Now we need G to be a set. All identity types are propositions. This tells us that the equalities we impose are forming a nice subtype.

If G were not a set, then we might need equalities between our equalities and equalities between all the equality equalities and so on. We'd need infinitely many equalities, which would be bad.

Proposition 2.1. *$\perp$, $\top$, $\mathbb{B}$, $\mathbb{N}$ are sets.*

3 h-levels

Definition 3.1. $T : U, n : \mathbb{N} \vdash \mathrm{isnType}(n,T) : U$, where

$$\mathrm{isnType}(n,T) := \begin{cases} \mathrm{isnType}(0,T) := \mathrm{isContr}(T) \\ \mathrm{isnType}(sn,T) := \prod_{s,t:T} \mathrm{isnType}(n, s = t) \end{cases}$$

3.1 Topology

$\pi_0(X)$ is the set of path-connected components. If $X = \mathbb{T}^2 \sqcup \mathbb{T}^2$, then $\pi_0(\mathbb{T}^2 \sqcup \mathbb{T}^2) = 2$, and $\pi_1(\mathbb{T}^2) = \mathbb{Z} \times \mathbb{Z}$. We also have $\pi_2(X)$, $\pi_3(X)$, and so on.

Being contractible is $\pi_n(X) = *$. Being proposition is $\pi_0(X) \in \{0,1\}$ and $\pi_n(X) = *$ for $n > 0$. Being a set means π_0 can be anything, and $\pi_n(X) = *$ for $n > 0$. Being a 3-type means $\pi_0(X), \pi_1(X)$ can be anything $\pi_n(X) = *$ for $n > 1$.

This gives us a stratification of types and a way to measure their complexity.

Proposition 3.1. *If T is an n-type, then T is an $n+1$-type.*

Proof. Induct on n. $n = 0$. Suppose T is a 0-type, so we have

$$c : \text{isContr}(T) := \sum_{t:T} \prod_{s:T} (t = s).$$

We need to show

$$\text{isProp}(T) := \prod_{s,s':T} \text{isContr}(s = s').$$

Fix s, s'. Since we have $c = (t, p)$, we get $p(s) : t = s$ and $p(s') : t = s'$. Then we have $p(s)^{-1} \cdot p(s') : s = s'$. Now we have

$$\prod_{s,s':T} \prod_{q:s=s'} p(s)^{-1} \cdot p(s') = q.$$

Then we induct on q. Thus we just need to show

$$\prod_{s:T} p(s)^{-1} \cdot p(s) = \text{refl}_s.$$

Let's quickly prove $p^{-1} \cdot p = \text{refl}_s$ for all paths p. Induct on p, then we want to show

$$\text{refl}_s^{-1} \cdot \text{refl}_s = \text{refl}_s,$$

but this is true by definition of inverses and path composition for reflexivities.

So we have a term in

$$\prod_{s,s':T} \text{isContr}(s = s') =: \text{isProp}(T).$$

Now the inductive step. Want to show that

$$\text{isnType}(n, T) \to \text{isnType}(sn, T).$$

By definition,

$$\text{isnType}(sn, T) := \prod_{x,y:T} \text{isnType}(n, x = y).$$

Then by the induction, we have $\ell_{n,x=y} : \text{isnType}(n, x = y) \to \text{isnType}(sn, x = y)$. Then composing this with $p : \text{isnType}(n, T)$, we obtain a term of $\text{isnType}(sn, T)$. ■

Corollary. *$\top$ is a proposition, set. $\bot$ is a set.*

Proposition 3.2. *$\mathbb{B}$ is a set.*

Proposition 3.3. *If we have $\mathbb{B}$ and the univalence axiom, then U is not a set. Indeed, it is not an n-type for any n.*