Type Theory with Paige North 7/19

Jason Schuchardt

July 22, 2019

1 Identity types

Last time we talked about identity terms in Σ-types, and we gave an explicit description. Can we do this for other types? Unfortunately, in general this isn't possible for the other types. There are models of type theory, in which the things we'd want to be the same aren't the same. So we postulate this for our type theory, since we don't care about those models.

1.1 Π types

$$\mathrm{Id}_{\prod_{x:B} E(x)}(f, f') \simeq ?$$

We already talked about homotopy,

$$f \sim f' := \prod_{x:B} \mathrm{Id}(fx, f'x).$$

We have

$$\mathrm{idToHo} : \prod_{f,f' : \prod_{x:B} E(x)} \mathrm{Id}(f, f') \to (f \sim f').$$

How does this work? By induction, we need to produce

$$\prod_{f : \prod_{x:B} E(x)} (f \sim f) = \prod_{f} \prod_{x:B} \mathrm{Id}(fx, fx),$$

and we already have a term in this last type,

$$\lambda f, x.\mathrm{refl}_{fx}.$$

We would hope that there is a function in the other direction. I.e., we want a term of type

$$\mathrm{isEquiv}(\mathrm{idToHo}).$$

This doesn't exist in general, so we postulate the existence of such a term,

$$\mathrm{FunExt}_{f,f'} : \mathrm{isEquiv}(\mathrm{idToHo}_{f,f'}).$$

This is called function extensionality.

If you are interested in bad models, which don't satisfy this, you can check out: von Glehn. Dialectica Models of TT.

1.2 Id types

We could postulate the following:

$$\mathrm{UIP} : \mathrm{Id}_{\mathrm{Id}(s,t)}(p,q) \simeq \top$$

UIP stands for uniqueness of identity proofs. However, we definitely don't assume this one. It's not equivalent to assuming axiom K that we talked about before, but it's morally the same, so we don't want to assume this either.

1.3 U types

$$\mathrm{Id}_U(S,T) \simeq ?$$

Should ? be $S \simeq T$?

Just like with Π-types, we can produce a map

$$\mathrm{idToEquiv} : \prod_{S,T:U} \mathrm{Id}_U(S,t) \to (S \simeq T).$$

We can postulate

$$\mathrm{UA} : \mathrm{isEquiv}(\mathrm{idToEquiv}),$$

and this axiom is called the univalence axiom.

Univalence implies function extensionality. However, univalence and uniqueness of identity proofs are incompatible. If you assume both, you can prove false.

2 Programs

Mathematicians and theoretical CS people are interested in the following two programs.

2.1 Univalent foundations

1. It's a new foundation of mathematics.
2. It can be formalized/computer-checked.
3. Dependent type theory with Σ, Π, Id, $\top$, $\bot$, $\mathbb{B}$, $\mathbb{N}$, UA, propositional truncation, and propositional resizing. (These last two we haven't seen, and we won't talk about the last, since it's not clear that it's consistent).
4. Invented by Vladimir Voevodsky. He won a Fields medal for work in motivic homotopy theory. His proofs were found to have a lot of holes, though he managed to fix the ones found in his Fields medal work.
5. UniMath (GitHub)

2.2 Homotopy Type Theory

1. Basically the same, but not purporting to be a new foundation for mathematics.
2. DTT with Σ, Π, Id, certain (higher) inductive types, and univalence.
3. Emphasizes the connections between the type theory and classical homotopy theory.

3 h-levels

h stands for homotopy. Consider a type T. It has the terms $r, s, t : T$. Think of T as a space, with r, s, t points in the space. It has the paths $p, q : \mathrm{Id}(s, t)$. I.e., terms of the identity type represent paths between the points in the space. It also has paths $\alpha, \beta : \mathrm{Id}(p, q)$. We can think of these as deformations of paths in the space. I.e., these are homotopies of paths.

Write $\mathrm{Id}(s, t)$ as $s = t$, and $s = t$ as $s \equiv t$ now.

Stratify types into their homotopy levels.

Level 0: Types $\simeq \top$.

Level 1: Types $\simeq \perp$ or $\top$.

Level 2: Types that look like sets. (Types with terms)

Level 3: Types that look like graphs. (groupoids) (Types with terms and paths, but no paths between paths)

Level 4: 2-groupoids.

Now let's be rigorous.

3.1 h-level 0 (contractible)

Some people say that this is level -2 to make the numbering work out with the way we number groupoids.

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

where

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

Proposition 3.1. *$\top$ is contractible.*

Proof. Take $* : \top$, then we want $? : \prod_{x:\top} * = x$. By induction, we can assume $x \equiv *$ by the elimination rule for $\top$. Then we have $\mathrm{refl}_* : * = *$. This gives us $j_{\mathrm{refl}_*} : \prod_{x:\top} * = x$, as desired. ∎

Thus up to homotopy, $\top$ is a one point space.

Proposition 3.2. *We have a map $\mathrm{isContr}(S) \to (S \simeq \top)$ (actually this is an equivalence).*

Proof. Consider $(t, p) : \mathrm{isContr}(S)$, where $t : S$, and $p : \prod_{x:S} t = x$. There is a function $g : S \to \top$, given by $\lambda s.*$. There is a function $f : \top \to S$ by $* \mapsto t$.

We have $g \circ f \sim \mathrm{id}_\top := \prod_{x:\top} gfx = x$, which by induction, we can prove by proving on the canonical term, $*$. On the canonical term, we have $gf* \equiv gt \equiv *$, and so we have $\mathrm{refl}_* : gf* = *$.

Now we need to prove $f \circ g \sim \mathrm{id}_S := \prod_{x:S} fgx = x$. However, $gx \equiv *$, and $f* \equiv t$, so $fgx \equiv t$. Then p is already of this type. I.e., by definition, we have $p : \prod_{x:S} fgx = x$.

This completes the proof. ∎

HW: Consider a dependent type, $b : B \vdash E(b) : U$ such that $b : B \vdash c(b) : \mathrm{isContr}(E(b))$, then

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

This generalizes a problem on HW3, where we prove

$$\sum_{b:B} \top \simeq B.$$

Proposition 3.3. *Given $f : A \to B$,*

$$\text{isEquiv}(f) \simeq \prod_{b:B} \text{isContr}\left(\sum_{a:A} fa = b\right)$$

I.e., f being an equivalence is equivalent to all the fibers of f being contractible.

Proposition 3.4. *For any type T, and $t : T$, the type*

$$\sum_{u:T} t = u$$

is contractible.

Note that this is an English translation of the type theory statement:

$$T : U, t : T \vdash ? : \text{isContr}\left(\sum_{u:T} t = u\right)$$

Topologically (roughly), this is the statement that the universal cover of a path connected space is simply connected. Note that we have a point in this space for every path out of t.

Proof. Center of construction, (t, refl_t).

$$\prod_{s:\sum_{u:T} t=u} ((t, \text{refl}_t) = s) \simeq \sum_{p:t=\pi_1 s} (p_* \text{refl}_t = \pi_2 s)$$

HW: prove that we have $\text{refl}_{\text{refl}_t} : \text{refl}_{t,*}\text{refl}_t = \text{refl}_t$ by showing $\text{refl}_{t,*}\text{refl}_t \equiv \text{refl}_t$. ∎