Computational Higher Type Theory (CHTT)

Robert Harper

Lecture Notes of Week 11 by Di Wang

1 Setting the Scene

Last week, we further explored some properties of the semantics of computational dependent type theory, such as functionality, P.E.R valued, symmetry, and transitivity. Then we revisited Intensional Type Theory (ITT) and claimed there is no notion of function extensionality, and we even can not prove $0 \neq 1$ in ITT. We enriched ITT by introducing *universes*, i.e., types of types, and studied higher identification $\mathsf{Id}_{\mathcal{U}}$ in a universe $\mathcal{U}$. We explored the *groupoid* structure of identification. Intuitively, for two types $A, B : \mathcal{U}$, $\mathsf{Id}_{\mathcal{U}}(A, B)$ is inhabited if there is an "isomorphism" $f : A \to B$ between A and B. The question remains how to express the idea in the type theory.

This week, we are going to study Homotopy Type Theory (HoTT), which is an extension of ITT with the following two ideas (Univalent Foundations Program [2013]):

1. **Univalence:** types are identified up to *equivalence*, which is defined based on the idea of isomorphism.
2. **Higher inductive types:** types can specify *points*, *paths*, and higher paths.

2 Univalence

The question we are going to address in this section is the following:

> How do we characterize $\mathsf{Id}_{\mathcal{U}}(A, B)$ for types $A, B : \mathcal{U}$ in ITT enriched with universes?

Before proceeding, we review some notions of ITT:

$\mathsf{Id}_A(M, N), M =_A N$	identity type, identification type, path type
$\mathsf{J}_{a,b,c.C}(a.Q)(P)$	elimination form of the identity type, path induction
$\mathsf{refl}_A(M)$	introduction form of the identity type
$\mathsf{transport}[a.B](P)$	transport

Intuitively, two types A and B are equal iff they are isomorphic up to identification. First, we observe that any function $f : A \to B$ preserves identification, i.e., f behaves functionally on paths.

Lemma 1. *If $f : A \to B$ and $P : M =_A M'$, then there is an operation $\mathsf{ap}_f(P) : f(M) =_B f(M')$, such that $\mathsf{ap}_f(\mathsf{refl}_A(M)) \equiv \mathsf{refl}_B(f(M))$.*

Proof. We give a definition of $\mathsf{ap}_f(P)$ by path induction on P:

$$\mathsf{ap}_f(P) := \mathsf{J}_{a,b,-.f(a)=_B f(b)}(a.\mathsf{refl}_B(f(a)))(P).$$

□

Then we want to extend this result to Π-types $f : \Pi a : A.\ B$. At the first attempt, we might write down the following formulation.

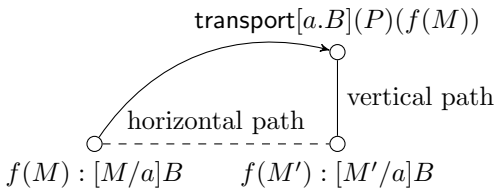
Attempt 2. *If $f : \Pi a : A.\ B$ and $P : M =_A M'$, then there is an operation $\mathsf{apd}_f(P) : f(M) =_B f(M')$.*

However, this statement will not typecheck—B could depend on a, thus $f(M)$ and $f(M')$ could have different types $[M/a]B$ and $[M'/a]B$, respectively. In other words, we need to come up with a notion of *heterogeneous paths* between $f(M)$ and $f(M')$.

Recall that *transport* has the following property: If $a : A \vdash B\ \mathsf{type}$ and $P : M =_A M'$, then $\mathsf{transport}[a.B](P) : [M/a]B \to [M'/a]B$, as well as $\mathsf{transport}[a.B](\mathsf{refl}_A(M))(Q) \equiv Q$ for any $Q : [M/a]B$. Then we can define heterogeneous paths between $f(M)$ and $f(M')$ as:

$$f(M) =_P^{a.B} f(M') := \mathsf{transport}[a.B](P)(f(M)) =_{[M'/a]B} f(M')$$

which is illustrated by the following figure.



Lemma 3. *If $f : \Pi a : A.\ B$ and $P : M =_A M'$, then there is an operation $\mathsf{apd}_f(P) : f(M) =_P^{a.B} f(M')$.*

Proof. We give a definition of $\mathsf{apd}_f(P)$ by path induction on P:

$$\mathsf{apd}_f(P) := \mathsf{J}_{a,b,p.f(a)=_p^{a.B} f(b)}(a.\mathsf{refl}_B(f(M)))(P).$$

□

Then we need to reason about when a function $f : A \to B$ is a bijection up to identification. Intuitively, f is a bijection if for any $b : B$, $f^{-1}(b)$ contains exactly one element. To proceed, we give a formal account of $f^{-1}(b)$.

Definition 4 (Fibers)**.** *For any $b : B$, its fiber is defined as $f^{-1}(b) := \Sigma a : A.\ f(a) =_B b$.*

Rather than containing a single element, we want to express the property as containing a *center* such that every element has a path to the center.

Definition 5 (Contractibility)**.** *A type A is contractible iff there exists a center $c : A$ such that every element $a : A$ is identical to c. Formally, the following type expresses contractibility:*

$$\mathsf{isContractible}(A) := \Sigma c : A.\ \Pi a : A.\ a =_A c.$$

A contractible type is also called a singleton.

Now we are able to express that $f : A \to B$ is a bijection up to identification.

Definition 6 (Equivalences)**.** *$f : A \to B$ is an equivalence between types A and B, iff for any $b : B$, its fiber is contractible. Formally, the following type expresses this property:*

$$\mathsf{isEquiv}(f) := \Pi b : B.\ \mathsf{isContractible}(f^{-1}(b)).$$

Further, we say that two types $A, B : \mathcal{U}$ are equivalent*, iff there is an equivalence $f : A \to B$. Formally, the following type characterizes this property:*

$$A \simeq B := \Sigma f : A \to B.\ \mathsf{isEquiv}(f).$$

Recall that our intuition is that two types A and B are equal iff they are isomorphic up to identification. We might come up with the following claim:

Proposition 7 (Pre-univalence)**.** *$A =_{\mathcal{U}} B$ has inhabitants if and only if $A \simeq B$ does.*

We can strengthen this idea by claiming the two types $A =_{\mathcal{U}} B$ and $A \simeq B$ are indeed equivalent. In fact, we are able to define a map from $A =_{\mathcal{U}} B$ to $A \simeq B$.

Lemma 8. *There is a function* $\mathsf{idtoequiv}_{A,B} : (A =_{\mathcal{U}} B) \to (A \simeq B)$.

Proof. For any path $p : A =_{\mathcal{U}} B$, we define $\mathsf{idtoequiv}_{A,B}(p)$ by path induction on p:

$$\mathsf{idtoequiv}_{A,B} := \lambda p : A =_{\mathcal{U}} B.\mathsf{J}_{a,b,-.a\simeq b}(a.\langle \mathsf{id}_a, \lambda x : a.\mathsf{pf\_contr}\rangle)(p)$$

where $a : \mathcal{U}, x : a \vdash \mathsf{pf\_contr} : \mathsf{isContractible}((\mathsf{id}_a)^{-1}(x))$. Let $T := (\mathsf{id}_a)^{-1}(x)$. $\mathsf{pf\_contr}$ is defined as

$$\mathsf{pf\_contr} := \langle\langle x, \mathsf{refl}_a(x)\rangle, \lambda y : T.\mathsf{refl}_T(\langle y.1, y.2\rangle)\rangle.$$

□

However, we could not *prove* that $\mathsf{idtoequiv}_{A,B}$ is an equivalence. Instead, we introduce an *axiom* to ensure it is an equivalence: Voevodsky's *univalence axiom.*

Axiom 9 (Univalence)**.** $\mathsf{idtoequiv}_{A,B}$ *is an equivalence. For $A, B : \mathcal{U}$ and $E : A \simeq B$, the axiom gives us a term $\mathsf{ua}(A, B, E) : A =_{\mathcal{U}} B$.*

Remark 10. *The introduction of univalence axiom breaks computationality. Suppose we are evaluating $\mathsf{J}_{a,b,p.C}(a.Q)(\mathsf{ua}(A, B, E))$. The computation is stuck because $\mathsf{ua}(A, B, E)$ is given by an axiom and thus irreducible. The insight is that we should define entailment* before *implication, i.e., first express identity judgmentally, and then internalize the identity into the language. We will explore this idea in future lectures.*

We then turn to show that ITT extended with univalence has function extensionality.

Definition 11 (Total path spaces)**.** *The total path space in a type A is characterized by the type $\mathsf{Paths}(A) := \Sigma a : A.\ \Sigma b : A.\ a =_A b$. In addition, $\Sigma b : A.\ a =_A b$ is called the* based path space *at a. Based path spaces are contractible by definition.*

Lemma 12. *A is equivalent to $\mathsf{Paths}(A)$, i.e., $A \simeq \mathsf{Paths}(A)$ has inhabitants.*

Proof. We prove this by showing there are a map f from A to $\mathsf{Paths}(A)$ and a map g from $\mathsf{Paths}(A)$ to A, as well as f and g are mutually inverse.

$$\begin{array}{lcl} f & := & \lambda a : A.\langle a, a, \mathsf{refl}_A(a)\rangle \\ g & := & \lambda t : \mathsf{Paths}(A).t.1. \end{array}$$

It is straightforward to show that $g(f(a)) =_A a$ for all $a : A$. We need to show that $f(g(t)) =_{\mathsf{Paths}(A)} t$ for all $t : \mathsf{Paths}(A)$, i.e., $\langle t.1, t.1, \mathsf{refl}_A(t.1)\rangle =_{\mathsf{Paths}(A)} t$. This is proved by the fact that $\langle t.1, \mathsf{refl}_A(t.1)\rangle$ belongs to the based path space at $t.1$, which is contractible. □

By univalence axiom we obtain the following identification.

Corollary 13. *A is identical to $\mathsf{Paths}(A)$, i.e., $A =_{\mathcal{U}} \mathsf{Paths}(A)$ has inhabitants.*

Definition 14 (Homotopies)**.** *The homotopy between types A and B is characterized by the type $\mathsf{Htpy}(A, B) := \Sigma f, g : A \to B.\ A \sim B$, where $A \sim B := \Pi a : A.\ f(a) =_B g(a)$.*

Lemma 15. *$A \to \mathsf{Paths}(B)$ is equivalent to $\mathsf{Htpy}(A, B)$, i.e., $(A \to \mathsf{Paths}(B)) \simeq \mathsf{Htpy}(A, B)$ has inhabitants.*

Proof. We prove this by showing there are a map f from $A \to \mathsf{Paths}(B)$ to $\mathsf{Htpy}(A, B)$ and a map g from $\mathsf{Htpy}(A, B)$ to $A \to \mathsf{Paths}(B)$, as well as f and g are mutually inverse.

$$\begin{array}{rcl} f & := & \lambda F : A \to \mathsf{Paths}(B).\langle \lambda a : A.F(a).1, \lambda a : A.F(a).2, \lambda a : A.F(a).3\rangle \\ g & := & \lambda H : \mathsf{Htpy}(A, B).\lambda a : A.\langle H.1(a), H.2(a), H.3(a)\rangle. \end{array}$$

It is straightforward to show they are mutually inverse. □

By univalence axiom we obtain the following identification.

Corollary 16. *$A \to \mathsf{Paths}(B)$ is identical to $\mathsf{Htpy}(A, B)$, i.e., $A \to \mathsf{Paths}(B) =_{\mathcal{U}} \mathsf{Htpy}(A, B)$ has inhabitants.*

Lemma 17. *$\mathsf{Paths}(A \to B)$ is identical to $\mathsf{Htpy}(A, B)$, i.e., $\mathsf{Paths}(A \to B) =_{\mathcal{U}} \mathsf{Htpy}(A, B)$ has inhabitants.*

Proof. By transitivity of identification and lemmas above we can derive the following:

$$\begin{array}{rl} & \mathsf{Paths}(A \to B) \\ =_{\mathcal{U}} & A \to B \\ =_{\mathcal{U}} & A \to \mathsf{Paths}(B) \\ =_{\mathcal{U}} & \mathsf{Htpy}(A, B). \end{array}$$

□

Corollary 18. *Univalence entails function extensionality.*

Proof. Function extensionality states that identical functions map the same arguments to identical results. Identical functions in $A \to B$ are captured by $\mathsf{Paths}(A \to B) = \Sigma f, g : A \to B.\ f =_{A \to B} g$. Functions that map the same arguments to identical results are captured by $\mathsf{Htpy}(A, B) = \Sigma f, g : A \to B.\ \Pi a : A.\ f(a) =_A g(a)$. Because $\mathsf{Paths}(A \to B)$ is identical to $\mathsf{Htpy}(A, B)$, we conclude function extensionality in ITT extended with univalence. □

3 Higher Inductive Types

Inductive types are essentially characterized by a "mapping out" property. Recall the formalization of natural numbers. We define two introduction forms (for zero and successors, respectively), and one elimination form (i.e., natural recursion). Then we claim the natural number type is the least thing that satisfies the specification.

$$\frac{}{\Gamma \vdash \mathsf{z} : \mathsf{Nat}}\ (\text{I-ZERO}) \qquad \frac{\Gamma \vdash M : \mathsf{Nat}}{\Gamma \vdash \mathsf{s}(M) : \mathsf{Nat}}\ (\text{I-SUCC})$$

$$\frac{\Gamma \vdash M : \mathsf{Nat} \qquad \Gamma \vdash M_0 : [\mathsf{z}/n]A \qquad \Gamma, n : \mathsf{Nat} \vdash A\ \mathsf{type} \qquad \Gamma, n : \mathsf{Nat}, a : A \vdash M_1 : [\mathsf{s}(n)/n]A}{\Gamma \vdash \mathsf{natrec}_{n.A}\{M_0; n.a.M_1\}(M) : [M/n]A}\ (\text{E})$$

As a generalization of the idea, in defining a higher inductive type, we might have constructors for not only *points*, but also *paths* as well as higher paths in that type. In other words, an inductively defined type A is somehow coupled to $\mathsf{Id}_A(-,-)$, $\mathsf{Id}_A(-,-)$ is somehow coupled to $\mathsf{Id}_{\mathsf{Id}_A(-,-)}(*,*)$, and so on.

As an example, we explore the *interval* type, denoted by $\mathbb{I}$. Following are the introduction forms.

$$\frac{}{\Gamma \vdash 0 : \mathbb{I}} \qquad \frac{}{\Gamma \vdash 1 : \mathbb{I}} \qquad \frac{}{\Gamma \vdash \mathsf{seg} : \mathsf{Id}_{\mathbb{I}}(0, 1)}$$

Fact 19. $\mathbb{I}$ *is contractible.*

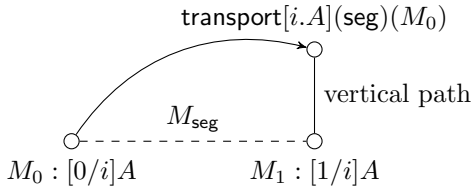
The elimination form is trickier than introduction forms—we should keep in mind that the type is the least thing that satisfies our specification. At the first attempt, we assume the result type of the elimination does not depend on the value to be eliminated.

$$\frac{\Gamma \vdash A\ \mathsf{type} \qquad \Gamma \vdash M : \mathbb{I} \qquad \Gamma \vdash M_0 : A \qquad \Gamma \vdash M_1 : A \qquad \Gamma \vdash M_{\mathsf{seg}} : \mathsf{Id}_A(M_0, M_1)}{\Gamma \vdash \mathsf{Irec}(M_0; M_1; M_{\mathsf{seg}})(M) : A}$$

The following properties should hold:

$$\begin{aligned}&\mathsf{Irec}(M_0; M_1; M_{\mathsf{seg}})(0) \equiv M_0\\ &\mathsf{Irec}(M_0; M_1; M_{\mathsf{seg}})(1) \equiv M_1\\ &\mathsf{ap}_{\mathsf{Irec}(M_0;M_1;M_{\mathsf{seg}})(\cdot)}(M_{\mathsf{seg}}) =_{\mathsf{Id}_A(M_0,M_1)} M_{\mathsf{seg}}\end{aligned}$$

Then we consider the general case where A could depend on the value of M. Similar to previous discussion on ap and apd, we make use of transport to establish heterogeneous paths, which is illustrated by the following figure.



The general elimination form is the following:

$$\frac{\begin{array}{c}\Gamma, i : \mathbb{I} \vdash A\ \mathsf{type}\\ \Gamma \vdash M : \mathbb{I} \qquad \Gamma \vdash M_0 : [0/i]A \qquad \Gamma \vdash M_1 : [1/i]A \qquad \Gamma \vdash M_{\mathsf{seg}} : M_0 =^{i.A}_{\mathsf{seg}} M_1\end{array}}{\Gamma \vdash \mathsf{Irec}(M_0; M_1; M_{\mathsf{seg}})(M) : [M/i]A}$$

Lemma 20. $\mathsf{Paths}(A)$ *is identical to* $\mathbb{I} \to A$.

Proof. By univalence axiom, it suffices to show that $\mathsf{Paths}(A)$ is equivalent to $\mathbb{I} \to A$. We then show there are a map f from $\mathsf{Paths}(A)$ to $\mathbb{I} \to A$ and a map g from $\mathbb{I} \to A$ to $\mathsf{Paths}(A)$, as well as f and g are mutually inverse.

$$\begin{aligned}f &:= \lambda t : \mathsf{Paths}(A).\lambda i : \mathbb{I}.\mathsf{Irec}(t.1; t.2; t.3)(i)\\ g &:= \lambda h : \mathbb{I} \to A.\langle h(0), h(1), \mathsf{ap}_h(\mathsf{seg})\rangle.\end{aligned}$$

It is straightforward to show they are mutually inverse. □

Although the interval type is not very interesting, it has some interesting features, such as it could lead to another proof of function extensionality.

Lemma 21. $\mathsf{Paths}(A \to B)$ *is identical to* $\mathsf{Htpy}(A, B)$*, i.e.,* $\mathsf{Paths}(A \to B) =_{\mathcal{U}} \mathsf{Htpy}(A, B)$ *has inhabitants.*

Proof. By transitivity of identification and lemmas above we can derive the following:

$$\begin{aligned}&\mathsf{Paths}(A \to B)\\ =_{\mathcal{U}}\ &\mathbb{I} \to (A \to B)\\ =_{\mathcal{U}}\ &(\mathbb{I} \times A) \to B\\ =_{\mathcal{U}}\ &(A \times \mathbb{I}) \to B\\ =_{\mathcal{U}}\ &A \to (\mathbb{I} \to B)\\ =_{\mathcal{U}}\ &A \to \mathsf{Paths}(B)\\ =_{\mathcal{U}}\ &\mathsf{Htpy}(A, B).\end{aligned}$$

□

4 Outlook

We will finish exploration of HoTT by investigating another example of higher inductive types, the *circle* type, which contains a base (as a point) and a loop (as a path). Then we will redevelop the idea of *dimensionality* directly and turn to discuss cubical type theory, which takes care of computationality and univalence becomes definable in that framework.

References

The Univalent Foundations Program. *Homotopy Type Theory: Univalent Foundations of Mathematics*. `https://homotopytypetheory.org/book`, Institute for Advanced Study, 2013.