15-819 Homotopy Type Theory Lecture Notes

Joseph Lee and Kristina Sojakova

October 28 and 30, 2013

1 The Path Spaces of Coproducts

Recall from last week that we intend to characterize paths in coproducts by showing

$$\prod_{x:A+B} \prod_{x':A+B} \mathsf{Id}_{A+B}(x, x') \simeq F(x, x')$$

where $F : (A + B) \to (A + B) \to \mathcal{U}$ is defined by nested case-analysis so that

$$\begin{aligned} F(\mathsf{inl}(a), \mathsf{inl}(a')) &\equiv \mathsf{Id}_A(a, a') \\ F(\mathsf{inr}(a), \mathsf{inr}(a')) &\equiv \mathsf{Id}_B(b, b') \\ F(\mathsf{inl}(a), \mathsf{inr}(b)) &\equiv 0 \\ F(\mathsf{inr}(a), \mathsf{inl}(b)) &\equiv 0 \end{aligned}$$

To this end we define a function $f : \prod_{x:A+B} \prod_{x':A+B} \mathsf{Id}_{A+B}(x, x') \to F(x, x')$ by

$$f := \lambda x.\lambda x'.\lambda p.\ \mathsf{J}[F](p; z.\, \mathsf{case}(z; a.\mathsf{refl}_A(a); b.\mathsf{refl}_B(b)))$$

Next we need to define a function $g : \prod_{x:A+B} \prod_{x':A+B} F(x, x') \to \mathsf{Id}_{A+B}(x, x')$ such that $g(x, x')$ is a quasi-inverse of $f(x, x')$. We put

$$\begin{aligned} g := \lambda x.\lambda x'.\, \mathsf{case}(x; a.\, \mathsf{case}(x'; a'.\lambda z : F(\mathsf{inl}(a), \mathsf{inl}(a')).\mathsf{ap}_{\mathsf{inl}}(z); b'.\lambda z : F(\mathsf{inl}(a), \mathsf{inr}(b')).\mathsf{abort}_{(}z)); \\ b.\, \mathsf{case}(x'; a'.\lambda z : F(\mathsf{inr}(b), \mathsf{inl}(a')).\mathsf{abort}_{(}z); b'.\lambda z : F(\mathsf{inr}(b), \mathsf{inr}(b')).\mathsf{ap}_{\mathsf{inr}}(z))) \end{aligned}$$

We then have to exhibit terms

$$\begin{aligned} \alpha &: \prod_{x:A+B} \prod_{x':A+B} \prod_{u:F(x,x')} f(g(u)) =_{F(x,x')} u \\ \beta &: \prod_{x:A+B} \prod_{x':A+B} \prod_{v:\mathsf{Id}_{A+B}(x,x')} g(f(v)) =_{\mathsf{Id}_{A+B}(x,x')} v \end{aligned}$$

These terms are left as homework exercises.

Exercise. *Characterize the path space of the empty type* $\mathbf{0}$.

2 The Path Spaces of Identity Types

For a given type A, we would like to characterize the types $\mathsf{Id}_A(\_,\_)$, $\mathsf{Id}_{\mathsf{Id}_A(\_,\_)}(\_,\_)$, $\mathsf{Id}_{\mathsf{Id}_{\mathsf{Id}_A(\_,\_)}(\_,\_)}(\_,\_)$, and so on. While this is possible for certain specific types such as $\mathbf{0}, \mathbf{1}, A \times B, A \to B, A + B$ and Nat, it can be very difficult for other types, even seemingly simple ones. For example, determining the loop-spaces of n-spheres - i.e., path spaces based at a single point, denoted by $\Omega(S^n)$ - is a famous open problem in algebraic topology.

What we can say, however, is that if two types are equivalent then so are their path spaces:

Lemma. *If $f : A \to B$ is an equivalence, then so is $\mathsf{ap}_f : \mathsf{Id}_A(a, a') \to \mathsf{Id}_B(f(a), f(a'))$.*

Proof. Because f is an equivalence, it has a quasi-inverse $f^{-1} : B \to A$ and we have the following coherences:

- $\alpha : \prod_{a:A} f^{-1}(f(a)) =_A a$
- $\beta : \prod_{b:B} f(f^{-1}(b) =_B b$

In order to show that ap_f is an equivalence, it suffices to give a quasi-inverse $\mathsf{ap}_f^{-1} : \mathsf{Id}_B(f(a), f(a')) \to \mathsf{Id}_A(a, a')$, which we define by

$$\mathsf{ap}_f^{-1}(q) := \alpha(a)^{-1} \cdot \mathsf{ap}_{f^{-1}}(q) \cdot \alpha(a')$$

We now need to construct coherences

$$\gamma : \prod_{p:\mathsf{Id}_A(a,a')} \mathsf{ap}_f^{-1}(\mathsf{ap}_f(p)) =_{\mathsf{Id}_A(a,a')} p$$
$$\delta : \prod_{q:\mathsf{Id}_B(f(a),f(a'))} \mathsf{ap}_f(\mathsf{ap}_f^{-1}(q)) =_{\mathsf{Id}_B(f(a),f(a'))} q$$

This will imply that ap_f^{-1} is indeed a quasi-inverse of ap_f and thus both are equivalences We leave these as exercises. □

Exercise. *Define the coherence γ in the above proof by path induction.*

Exercise. *Define the coherence δ in the above proof as follows:*
1. Use the naturality of β to show that

$$\beta(f(a))^{-1} \cdot \mathsf{ap}_f(\mathsf{ap}_{f^{-1}}(q)) \cdot \beta(f(a')) =_{\mathsf{Id}_B(f(a),f(a'))} q$$

2. Use the naturality of α to show that

$$\alpha(f^{-1}(f(a)))^{-1} \cdot \mathsf{ap}_{f^{-1}}(\mathsf{ap}_f(\mathsf{ap}_{f^{-1}}(q))) \cdot \alpha(f^{-1}(f(a'))) =_{\mathsf{Id}_A(f^{-1}(f(a)),f^{-1}(f(a')))} \mathsf{ap}_{f^{-1}}(q)$$

3. Use the naturality of α to show that

$$\alpha(f^{-1}(f(a))) =_{\mathsf{Id}_A((f^{-1}(f(f^{-1}(f(a))))),f^{-1}(f(a)))} \mathsf{ap}_{f^{-1}}(\mathsf{ap}_f(\alpha(a)))$$

and similarly for a'.

4. Use 1), 2), 3) and the naturality of β to obtain the desired conclusion

$$\mathsf{ap}_f(\alpha(a)^{-1} \cdot \mathsf{ap}_{f^{-1}}(q) \cdot \alpha(a')) =_{\mathsf{Id}_B(f(a),f(a'))} q$$

3 Transport Properties of Identity

The identity type family on a type A can be thought of as a function $IdA(\_,\_) : A \to A \to \mathcal{U}$. Keeping the first argument $x : A$ fixed yields a type family $IdA(x,\_) : A \to \mathcal{U}$. For any path $q : y =_A y'$, the fibers $\mathsf{Id}_A(x,y)$ and $\mathsf{Id}_A(x,y')$ are related by the transport function

$$\mathsf{tr}[z.\mathsf{Id}_A(x,z)](q) : \mathsf{Id}_A(x,y) \to \mathsf{Id}_A(x,y')$$

Can we give an explicit description of this function? Clearly we can construct a function of the desired type "manually" by taking a $p : x = Ay$ and concatenating it with q to obtain $p \cdot q : x =_A y'$. Fortunately, it turns out this is precisely characterizes the behavior of the transport function, up to a propositional equality:

Lemma. *For any term $x : A$, and paths $q : y =_A y'$, $p : x =_A y$ we have*

$$\mathsf{tr}[z.\mathsf{Id}_A(x,z)](q)(p) =_{\mathsf{Id}_A(x,y')} p \cdot q$$

We have a similar characterization of transport in the case when the second argument is fixed:

Lemma. *For any term $y : A$, and paths $q : x =_A x'$, $p : x =_A y$ we have*

$$\mathsf{tr}[z.\mathsf{Id}_A(z,y)](q)(p) =_{\mathsf{Id}_A(x',y)} q^{-1} \cdot p$$

We notice that in the first case, we use the path q as-is whereas in the second case we first have to invert it. This can be described in category-theoretic terms as saying that the type family Id_A is *covariant in the second argument and contravariant in the first.*

Finally, we can consider the case of loops when the base point is allowed to vary:

Lemma. *For any paths $q : x =_A y$, $p : x =_A x$ we have*

$$\mathsf{tr}[z.\mathsf{Id}_A(z,z)](q)(p) =_{\mathsf{Id}_A(y,y)} q^{-1} \cdot p \cdot q$$

All these lemmas follow by straightforward path induction.

4 Justifying the Identity-Elimination Rule

Recall the identity elimination rule:

$$\frac{\Gamma \vdash P : \mathsf{Id}_A(M,N) \quad \Gamma, x{:}A, y{:}A, z{:}\mathsf{Id}_A(x,y) \vdash C\ \mathsf{type} \quad \Gamma, x{:}A \vdash Q : [x,x,\mathsf{refl}_A(x)/x,y,z]C}{\Gamma \vdash \mathsf{J}[x.y.z.C](P, x.Q) : [M,N,P/x,y,z]C}\ \mathsf{Id\text{-}E}$$

In Extensional Type Theory (ETT), this rule is a consequence of the identity reflection and UIP rules and thus has no special status.

In Intensional Type Theory (ITT), this rule can be understood as an induction principle: since the only way we can construct a proof of equality $P : M =_A N$ is by reflexivity in the case when $M \equiv N$, in order to prove $C[M,N,P]$ it is sufficient to prove $C[x,x,\mathsf{refl}_A(x)]$ for an arbitrary $x : A$. This intuition is justified by the following very important (and highly nontrivial) theorem:

Theorem 1. *In an empty context, two terms are propositionally equal if and only if they are definitionally equal and any identity proof is necessarily a reflexivity. In other words, if* $\vdash P : \mathsf{Id}_A(M,N)$*, then* $\vdash M \equiv N : A$ *and* $\vdash P \equiv \mathsf{refl}_A(M,M) : \mathsf{Id}_A(M,N)$*.*

In HoTT, it is no longer the case that every proof of equality is a reflexivity. For example, we have the following non-trivial identity proofs:

- $\mathsf{funext}(H) : f =_{A\to B} g$, where $H : \prod_{a:A} f(a) =_B g(a)$
- $\mathsf{ua}(E) : A =_{\mathcal{U}} B$, where $E : \sum_{f:A\to B} \mathsf{isequiv}(f)$
- $\mathsf{seq} : 0 =_{\mathbb{I}} 1$, where $\mathbb{I}$ is the interval type
- $\mathsf{loop} : b =_{S^1} b$, where S^1 is the circle type

This suggests that in HoTT, terms of an identity type should not be thought of purely as proofs of identity but rather as paths between terms. Since there can be potentially many distinct paths between two terms, the identity elimination rule should no longer be thought of as an induction principle.

The presence of nontrivial paths, however, poses a serious problem with the computational interpretation of HoTT: for example, what should $\mathsf{J}[\_](\mathsf{funext}(H); x.Q)$, $\mathsf{J}[\_](\mathsf{ua}(E))$, or $\mathsf{J}[\_](\mathsf{seq})$ compute to?

Even leaving aside univalence and higher inductive types, the addition of the function extensionality axiom to ITT poses a problem with computation. In ETT, we get function extensionality for free from the identity reflection rule. In Observational Type Theory (OTT), which combines intensional and extensional aspects, we get function extensionality with some special arrangements. Both ETT and OTT admit a computational interpretation, albeit by different means.

The computational interpretation of HoTT is currently a principal open problem. So how do we justify the J-rule as a suitable identity elimination rule? Given

- $C : \prod_{x:A} \prod y : A, \mathsf{Id}_A xy \to \mathcal{U}$
- $M, N : A$ and $P : M =_A N$
- $x : A \vdash Q : C[x, x, \mathsf{refl}_A(x)]$

why should there exist a term $\mathsf{J}[x.y.z.C](P; x.Q) : C[M, N, P]$?

Plugging M into Q, we obtain a term $Q[M] : C[M, M, \mathsf{refl}_A(M)]$. Similarly, plugging M into C yields a type family $C[M] : \prod_{y:A}(p : M =_A y) \to \mathcal{U}$, which can be equivalently understood as the function

$$\lambda z.C[M, \pi_1(z), \pi_2(z)] : \Big(\sum_{y:A} \mathsf{Id}_A(M, y)\Big) \to \mathcal{U}$$

Since functions are supposed to be functorial, constructing a path γ from $(M, \mathsf{refl}_A(M))$ to (N, P) in the type $\sum_{y:A} \mathsf{Id}_A(M, y)$ would give us a term

$$\mathsf{ap}_{\lambda z.C[M,\pi_1(z),\pi_2(z)]}(\gamma) : C[M, M, \mathsf{refl}_A(M)] =_{\mathcal{U}} C[M, N, P]$$

We could thus obtain our desired conclusion of the J-rule as

$$\mathsf{tr}[x : \mathcal{U}.x](\mathsf{ap}_{\lambda z.C[M,\pi_1(z),\pi_2(z)]}(\gamma))(Q[M]) : C[M, N, P]$$

In order to construct a path $\gamma : (M, \mathsf{refl}_A(M)) =_{\sum_{y:A} \mathsf{Id}_A(M,y)} (N, P)$, we need a characterization of path spaces of Σ-types, outlined in the following exercise:

Exercise. *Characterize the path space of the type $\Sigma_{x:A}B$ by constructing a term of type*

$$\prod_{p,p':\sum_{x:A} B(x)} (\mathsf{Id}_{\sum_{x:A} B(x)}(p, p') \simeq \sum_{q:\pi_1(p)=_A\pi_1(p')} \pi_2(p) =_q^{x.B} \pi_2(p'))$$

The above exercise tells us that in order to construct a path from $(M, \mathsf{refl}_A(M))$ to (N, P) in the type $\sum_{y:A} M =_A y$, it is sufficient to construct an element of the type $\sum_{q:M=_A N} \mathsf{refl}_A(M) =_q^{y.M=_A y} P$. The natural choice for the first component of the pair is the path $P : M =_A N$ itself. It thus remains to show that $\mathsf{refl}_A(M) =_P^{y.M=_A y} P$. In particular, this means showing that

$$\mathsf{tr}[y.M =_A y](P)(\mathsf{refl}_A(M)) =_{\mathsf{Id}_A(M,N)} P$$

By the first lemma in Section. 3, we have $\mathsf{tr}[y.M =_A y](P)(\mathsf{refl}_A(M)) =_{\mathsf{Id}_A(M,N)} \mathsf{refl}_A(M) \cdot P$. Since the left-hand evaluates to P, we are done.

5 Introduction to Homotopy Types

One way to see HoTT is that Homotopy *Type Theory* is *Homotopy Type* Theory. That is, HoTT can be thought of as the theory of homotopy types. We have already encountered several examples of the homotopy type *set*, also sometimes called an *h-set* or a *0-type*; however, we have not explicitly labeled them as such. We have the following definition:

Definition. *A type A is called a* set *(or a* 0-type*), if for all $x, y : A$ and $p, q : x =_A y$, we have that $p =_{\mathsf{Id}_A(x,y)} q$. In other words, the following type is inhabited:*

$$\mathsf{isSet}(A) :\equiv \prod_{x,y:A} \prod_{p,q:\mathsf{Id}_A(x,y)} p =_{\mathsf{Id}_A(x,y)} q$$

Intuitively, the type A can be viewed as "discrete up to homotopy". The familiar example of the type Nat of natural numbers (unsurisingly) turns out to be a set. More about this and other Homotopy Types next week!