Homotopy Type Theory

Lecture 3: Identity types and Univalence

Nicola Gambino   Dan Licata   Peter LeFanu Lumsdaine

Mathematical Structures of Computation

Lyon, January 2014

Propositions as types

Today: developing some maths within the type theory.

Propositions as types

- Every statement represents a type;
- Proving/assuming a statment means constructing/assuming an element of a type.

Translations from prose to syntax will be given explicitly, at first.

Functions act on equalities

Proposition

Given a function $f : A \to B$*, elements* $x, y : A$*, and an equality* $p : \mathsf{Id}_A(x, y)$*, can define an equality* $\mathsf{ap}(f, p) : \mathsf{Id}_B(f(x), f(y))$*.*

$$\mathsf{ap}_{A,B} : \Pi(f : A \to B,\ x, y : A)\ \mathsf{Id}_A(x, y) \to \mathsf{Id}_B(f(x), f(y))$$

Functions act on equalities

Proposition

Given a function $f : A \to B$, elements $x, y : A$, and an equality $p : \mathsf{Id}_A(x, y)$, can define an equality $\mathsf{ap}(f, p) : \mathsf{Id}_B(f(x), f(y))$.

$$\mathsf{ap}_{A,B} : \Pi(f : A \to B,\ x, y : A)\ \mathsf{Id}_A(x, y) \to \mathsf{Id}_B(f(x), f(y))$$

Proof.

Apply Id-elimination. Goal type, for (x, y, p), is $\mathsf{Id}_B(f(x), f(y))$.

In case $(x, x, \mathsf{refl}(x))$, use $\mathsf{refl}(f(x)) : \mathsf{Id}_B(f(x), f(x))$.

$$\mathsf{ap}_{A,B} := \lambda f, x, y, p.\ \mathsf{J}(x, y, p, (x, y, p)\mathsf{Id}_B(f(x), f(y)), (x)\mathsf{refl}(f(x)))$$

□

$$\frac{p : \mathsf{Id}_A(a, b) \quad x, y : A,\ u : \mathsf{Id}(x, y) \vdash E : \mathsf{type} \quad x : A \vdash d : E[x/y, \mathsf{refl}(x)/u]}{\mathsf{J}(a, b, p, (x, y, u)E, (x)d) : E[a/x, b/y, p/u]}$$

Transport along equalities

Proposition

Given types $A, B : \mathsf{U}$*, an equality* $p : \mathsf{Id}_{\mathsf{U}}(A, B)$*, and an element* $a : \mathsf{T}(A)$*, can transport* a *along* p*, to an element* $\mathsf{trans}(p, a) : \mathsf{T}(B)$*.*

$$\mathsf{trans} : \Pi(A, B : \mathsf{U},\ p : \mathsf{Id}_{\mathsf{U}}(A, B))\ \mathsf{T}(A) \to \mathsf{T}(B)$$

Transport along equalities

Proposition

Given types $A, B : \mathsf{U}$, *an equality* $p : \mathsf{Id}_{\mathsf{U}}(A, B)$, *and an element* $a : \mathsf{T}(A)$, *can* *transport* *a along* p, *to an element* $\mathsf{trans}(p, a) : \mathsf{T}(B)$.

$$\mathsf{trans} : \Pi(A, B : \mathsf{U},\ p : \mathsf{Id}_{\mathsf{U}}(A, B))\ \mathsf{T}(A) \to \mathsf{T}(B)$$

Proof.

Id-elimination again. Goal type, for A, B, p, is $\mathsf{T}(A) \to \mathsf{T}(B)$.

In case $(A, A, \mathsf{refl}(A))$, use $\mathsf{id}_{\mathsf{T}(A)} := \lambda x : \mathsf{T}(A).\ x$. □

Transport along equalities

Proposition

Given types $A, B : \mathsf{U}$*, an equality* $p : \mathsf{Id}_{\mathsf{U}}(A, B)$*, and an element* $a : \mathsf{T}(A)$*, can* *transport* a *along* p*, to an element* $\mathsf{trans}(p, a) : \mathsf{T}(B)$*.*

$$\mathsf{trans} : \Pi(A, B : \mathsf{U},\ p : \mathsf{Id}_{\mathsf{U}}(A, B))\ \mathsf{T}(A) \to \mathsf{T}(B)$$

Proof.

Id-elimination again. Goal type, for A, B, p, is $\mathsf{T}(A) \to \mathsf{T}(B)$.

In case $(A, A, \mathsf{refl}(A))$, use $\mathsf{id}_{\mathsf{T}(A)} := \lambda x : \mathsf{T}(A).\ x$. □

Groupoid semantics

- ap: action of functor $f : A \to B$ on morphisms
- trans: morphisms in U are functors.

Identity in Bool

What do identity types look like? Example: Bool.

Proposition

In Bool*,* true *is not equal to* false*.*

$$\mathsf{Id}_{\mathsf{Bool}}(\mathsf{true}, \mathsf{false}) \to \mathsf{0}$$

Identity in Bool

What do identity types look like? Example: Bool.

Proposition

In Bool, true *is not equal to* false.

$$\mathsf{Id}_{\mathsf{Bool}}(\mathsf{true}, \mathsf{false}) \to \mathsf{0}$$

Proof.

Define $\mathsf{isTrue} : \mathsf{Bool} \to \mathsf{U}$, by $\mathsf{isTrue} := \lambda b.\ \mathsf{if}\ b\ \mathsf{then}\ \mathsf{1}^{\mathsf{U}}\ \mathsf{else}\ \mathsf{0}^{\mathsf{U}}$.

If $p : \mathsf{Id}_{\mathsf{Bool}}(\mathsf{true}, \mathsf{false})$, then $\mathsf{ap}(\mathsf{isTrue}, p) : \mathsf{Id}_{\mathsf{U}}(\mathsf{1}^{\mathsf{U}}, \mathsf{0}^{\mathsf{U}})$; so $\mathsf{trans}(\mathsf{ap}(\mathsf{isTrue}, p), \mathsf{tt}) : \mathsf{0}$. □

Identity in Bool

What do identity types look like? Example: Bool.

Proposition

In Bool, true *is not equal to* false.

$$\mathsf{Id}_{\mathsf{Bool}}(\mathsf{true}, \mathsf{false}) \to 0$$

Proof.

Define isTrue : Bool → U, by isTrue := $\lambda b.$ if b then 1^{U} else 0^{U}.

If p : $\mathsf{Id}_{\mathsf{Bool}}(\mathsf{true}, \mathsf{false})$, then $\mathsf{ap}(\mathsf{isTrue}, p) : \mathsf{Id}_{\mathsf{U}}(1^{\mathsf{U}}, 0^{\mathsf{U}})$; so $\mathsf{trans}(\mathsf{ap}(\mathsf{isTrue}, p), \mathsf{tt}) : 0$. □

Fact

$\mathsf{Id}_{\mathsf{Bool}}(\mathsf{true}, \mathsf{true})$ *has a unique element*

Identity in inductive types

Fact

For $b : \mathsf{Bool}$*, have an equivalence:*

$$\mathsf{Id}_{\mathsf{Bool}}(\mathsf{true}, b) \simeq \mathsf{T}(\mathsf{if}\ b\ \mathsf{then}\ 1^{\mathsf{U}}\ \mathsf{else}\ 0^{\mathsf{U}})$$

This characterises identity types of Bool (up to equivalence).

Similarly, can characterise Id-types of Σ, 1, 0, Nat. . .

Generally: of all inductive types.

Identity in inductive types

Fact

For $b : \mathsf{Bool}$, have an equivalence:

$$\mathsf{Id}_{\mathsf{Bool}}(\mathsf{true}, b) \simeq \mathsf{T}(\mathsf{if}\ b\ \mathsf{then}\ 1^{\mathsf{U}}\ \mathsf{else}\ 0^{\mathsf{U}})$$

This characterises identity types of Bool (up to equivalence).

Similarly, can characterise Id-types of Σ, 1, 0, Nat...

Generally: of all inductive types.

Remaining type: function types, and U. For these: cannot characterise Id-types just from $\mathbf{ML_1}$. Need extra axioms!

Identity in function types

Extensionality: objects should be equal if their uses are the same.

Functions are used through their values, so functions should be equal if all their values are.

Axiom (Functional extensionality)

Given functions $f, g : \Pi(x : A)B$, if the values $f(x)$ and $g(x)$ are equal for every $x : A$, then f and g are equal as functions.

$\mathsf{funext} : \Pi(f, g : \Pi(x : A)B)\ (\Pi(x : A)\mathsf{Id}_B(f(x), g(x))) \to \mathsf{Id}_{\Pi(x:A)B}(f, g)$

Fact

The resulting map

$$\mathsf{funext}_{f,g} : (\Pi(x : A)\mathsf{Id}_B(f(x), g(x))) \to \mathsf{Id}_{\Pi(x:A)B}(f, g)$$

is always an equivalence.

Functional extensionality characterises equality in function types.

Identity in the unvierse?

Question

What should equality in U be?

Elements of U are used via their types of elements.

Identity in the unvierse?

Question

What should equality in U be?

Elements of U are used via their types of elements.

In ZF(C) and similar, sets are equal if they have the same (absolute, global) elements.

In $\mathbf{ML_1}$, no absolute elements. Cannot compare elements of different types.

Identity in the unvierse?

Question

What should equality in U be?

Elements of U are used via their types of elements.

In ZF(C) and similar, sets are equal if they have the same (absolute, global) elements.

In $\mathbf{ML_1}$, no absolute elements. Cannot compare elements of different types.

To compare types: have to look at maps between them.

Idea

Elements of U should be equal if their types of elements are…

…isomorphic? bijectable? equivalent?

Equivalence

Definition

A left inverse for $f : A \to B$ is a function $g : B \to A$, together with a family of paths $\alpha : \Pi(x : A)\, \mathsf{Id}_A(g(f(x)), x)$.

A right inverse for $f : A \to B$ is a function $g : B \to A$, together with a family of paths $\beta : \Pi(x : A)\, \mathsf{Id}_B(f(g(x)), x)$.

An equivalence from A to B is a function $f : A \to B$, together with both a left inverse (g_l, α) and a right inverse (g_r, β).

$$\mathsf{Equiv}(A, B) := \Sigma(f : A \to B)\ (\Sigma(g_l : B \to A)\, \Pi(x : A)\, \mathsf{Id}_A(g_l(f(x)), x)) \\ \times (\Sigma(g_r : B \to A)\ \Pi(x : A)\, \mathsf{Id}_B(f(g_r(x)), x))$$

Equivalences: properties

Exercise

- For any type A, id_A is an equivalence.
- The composite of equivalences is an equivalence.
- A left or right inverse of an equivalence is an equivalence.

Equivalences: properties

Exercise

- For any type A, id_A is an equivalence.
- The composite of equivalences is an equivalence.
- A left or right inverse of an equivalence is an equivalence.

Factcercise

- If f is an equivalence, then its left inverse is unique. (That is. any two left inverses (g_l, α) for f are equal.)
- Similarly, the right inverse of an equivalence is unique.
- Hence, the full "is an equivalence" data for f is unique.

Equivalences: properties

Exercise

- For any type A, id_A is an equivalence.
- The composite of equivalences is an equivalence.
- A left or right inverse of an equivalence is an equivalence.

Factcercise

- If f is an equivalence, then its left inverse is unique. (That is. any two left inverses (g_l, α) for f are equal.)
- Similarly, the right inverse of an equivalence is unique.
- Hence, the full "is an equivalence" data for f is unique.

So: being an equivalence is merely a property.

(This is the reason for using two one-sided inverses, rather a single two-sided inverse $(f, (g, \alpha, \beta))$.)

Univalence

Proposition

For any $A, B : \mathsf{U}$, $p : \mathsf{Id}_{\mathsf{U}}(A, B)$, *the map* $\mathsf{trans}(p) : \mathsf{T}(A) \to \mathsf{T}(B)$ *is an equivalence.*

Proof.

In case $A, A, \mathsf{refl}(A)$, by definition $\mathsf{trans}(p)$ is $\mathsf{id}_A : \mathsf{T}(A) \to \mathsf{T}(A)$. □

This gives, for $A, B : \mathsf{U}$, canonical map
$\mathsf{trans}_{A,B} : \mathsf{Id}_{\mathsf{U}}(A, B) \to \mathsf{Equiv}(\mathsf{T}(A), \mathsf{T}(B))$.

Axiom (Univalence)

For each $A, B : \mathsf{U}$, $\mathsf{trans}_{A,B}$ is an equivalence:

$$\mathsf{trans}_{A,B} : \mathsf{Id}_{\mathsf{U}}(A, B) \simeq \mathsf{Equiv}(\mathsf{T}(A), \mathsf{T}(B)).$$

Identity in U is equivalence of types.

Write $\mathsf{uval}_{A,B}$ for the inverse map $\mathsf{Equiv}(\mathsf{T}(A), \mathsf{T}(B)) \to \mathsf{Id}_{\mathsf{U}}(A, B)$.

Univalence: consequences

Factcercise

For $w : \mathsf{Equiv}(\mathsf{T}(A), \mathsf{T}(B))$, the transport map $\mathsf{trans}(\mathsf{uval}(w)) : \mathsf{T}(A) \to \mathsf{T}(B)$ is just (the underlying function of) w itself.

$$\mathsf{Id}_{\mathsf{T}(A)\to\mathsf{T}(B)}(\mathsf{trans}(\mathsf{uval}(w)), w)$$

Univalence: consequences

Factcercise

For $w : \mathsf{Equiv}(\mathsf{T}(A), \mathsf{T}(B))$, the transport map $\mathsf{trans}(\mathsf{uval}(w)) : \mathsf{T}(A) \to \mathsf{T}(B)$ is just (the underlying function of) w itself.

$$\mathsf{Id}_{\mathsf{T}(A)\to\mathsf{T}(B)}(\mathsf{trans}(\mathsf{uval}(w)), w)$$

Corollary

Define $\mathsf{swap} : \mathsf{Bool} \to \mathsf{Bool}$ *by* $\mathsf{swap}\, b = \mathsf{if}\ b\ \mathsf{then\ false\ else\ true}$*; this is an equivalence.*

Then $\mathsf{uval}(\mathsf{swap})$, $\mathsf{uval}(\mathsf{id}_{\mathsf{Bool}})$ *are distinct in* $\mathsf{Id}_{\mathsf{U}}(\mathsf{Bool}^{\mathsf{U}}, \mathsf{Bool}^{\mathsf{U}})$.

Univalence: consequences

Factcercise

For $w : \mathsf{Equiv}(\mathsf{T}(A), \mathsf{T}(B))$, the transport map $\mathsf{trans}(\mathsf{uval}(w)) : \mathsf{T}(A) \to \mathsf{T}(B)$ is just (the underlying function of) w itself.

$$\mathsf{Id}_{\mathsf{T}(A)\to\mathsf{T}(B)}(\mathsf{trans}(\mathsf{uval}(w)), w)$$

Corollary

Define $\mathsf{swap} : \mathsf{Bool} \to \mathsf{Bool}$ *by* $\mathsf{swap}\, b = \mathsf{if}\ b\ \mathsf{then\ false\ else\ true}$*; this is an equivalence.*

Then $\mathsf{uval}(\mathsf{swap})$*,* $\mathsf{uval}(\mathsf{id}_{\mathsf{Bool}})$ *are distinct in* $\mathsf{Id}_{\mathsf{U}}(\mathsf{Bool}^{\mathsf{U}}, \mathsf{Bool}^{\mathsf{U}})$*.*

Corollary

In the set model, the universe of all small sets is not univalent.

Proof.

Each identity type in the set model has at most one element. □

Univalence in groupoids

Univalence forces a universe to look like a (higher) groupoid of all (small) types, not just a set of types.

Theorem

In the groupoid model, the groupoid of all small sets (discrete groupoids) is a univalent universe.

Proof.

$$[\![\mathsf{Id}_{\mathsf{U}}(A, B)]\!] = \mathrm{Hom}_{\mathsf{U}}(A, B) = \mathrm{Bij}(A, B)$$

$$[\![\mathsf{Equiv}(\mathsf{T}(A), \mathsf{T}(B))]\!] \simeq \mathrm{Equiv}(DA, DB) \simeq \mathrm{Bij}(A, B)$$

□

Fact

A (suitably closed) univalent universe of n-dimensional types is $(n+1)$-dimensional.

So: the groupoid of all small groupoids is not univalent.

References

The Univalent Foundations Program, *Homotopy type theory: Univalent foundations of mathematics*, Tech. report, Institute for Advanced Study, 2013.

Chris Kapulkin, Peter LeF. Lumsdaine, and Vladimir Voevodsky, *The simplicial model of univalent foundations*, preprint, 2012, `arXiv:1211.2851`.