Transfer of reasoning along equivalence of structures

Benedikt Ahrens

Outline

Indiscernibility of identicals

Indiscernibility of identicals

$$x = y \quad \rightarrow \quad \forall P\,(P(x) \leftrightarrow P(y))$$

- Reasoning **in logic** is invariant under equality
- **In mathematics**, reasoning should be invariant under weaker notion of sameness!

Equivalence principle

Reasoning in mathematics should be **invariant under** the appropriate notion of **sameness**.

Structure Identity Principle

Notion of sameness depends on the objects under consideration:

An equivalence principle for group theorists

$$G \cong H \to \forall \text{ group-theoretic properties } P, (P(G) \longleftrightarrow P(H))$$

An equivalence principle for category theorists

$$A \simeq B \to \forall \text{ category-theoretic properties } P, (P(A) \longleftrightarrow P(B))$$

Structure Identity Principle (SIP)

Isomorphic mathematical structures are structurally identical; i.e. have the same structural properties.

Structure Identity Principle

Notion of sameness depends on the objects under consideration:

An equivalence principle for group theorists

$$G \cong H \to \forall \text{ group-theoretic properties } P, (P(G) \longleftrightarrow P(H))$$

An equivalence principle for category theorists

$$A \simeq B \to \forall \text{ category-theoretic properties } P, (P(A) \longleftrightarrow P(B))$$

Structure Identity Principle (SIP)

Isomorphic mathematical structures are structurally identical; i.e. have the same structural properties.

What are "structural" properties?

Violating the equivalence principle

What is **not** a structural property?

Exercise

Find a statement about categories that is not invariant under the equivalence of categories

Violating the equivalence principle

What is **not** a structural property?

Exercise

Find a statement about categories that is not invariant under the equivalence of categories

A solution

"The category $\mathscr{C}$ has exactly one object."

Violating the equivalence principle

What is **not** a structural property?

Exercise

Find a statement about categories that is not invariant under the equivalence of categories

A solution

"The category $\mathscr{C}$ has exactly one object."

How can we identify "non-structural" statements?

Reasoning modulo equivalence

A mathematician's view

- Understands intuitively if a given statement is invariant under equivalence

Why make it more explicit?

- Might want to transfer constructions, not just proofs.
- For complicated mathematical objects, equivalences are complicated.
- Computer proof assistants require all the details.

A language for invariant properties

Michael Makkai, *Towards a Categorical Foundation of Mathematics*:

> *The basic character of the Principle of Isomorphism is that of a* ***constraint on the language*** *of Abstract Mathematics; a welcome one, since it provides for the separation of sense from nonsense.*

A language for invariant properties

Michael Makkai, *Towards a Categorical Foundation of Mathematics*:

> *The basic character of the Principle of Isomorphism is that of a* ***constraint on the language*** *of Abstract Mathematics; a welcome one, since it provides for the separation of sense from nonsense.*

Makkai's FOLDS (First Order Logic with Dependent Sorts)

A language for categorical structures in which only invariant properties can be expressed

- FOLDS is an interface to some underlying foundation
- Invariance only for properties, not for constructions

Univalent Foundations and the Univalence Principle

Voevodsky's goals

- Univalent Foundations as an "invariant language"
- Invariance not only for statements, but also for constructions: **any construction on objects in UF can be transported along equivalences of objects**

Univalent Foundations and the Univalence Principle

Voevodsky's goals

- Univalent Foundations as an "invariant language"
- Invariance not only for statements, but also for constructions: **any construction on objects in UF can be transported along equivalences of objects**

[. . .] My homotopy lambda calculus is an attempt to create a system which is very good at dealing with equivalences. In particular it is supposed to have the property that given any type expression F(T) depending on a term subexpression t of type T and an equivalence t->t' (a term of the type Eq(T;t,t')) there is a mechanical way to create a new expression F' now depending on t' and an equivalence between F(T) and F'(T') (note that to get F' one can not just substitute t' for t in F – the resulting expression will most likely be syntactically incorrect).

Email to Dan Grayson, Sept 2006

Outline

Overview of types in type theory

Type former	Notation	(special case)	canonical term
Inhabitant	$a : A$		
Dependent type	$x : A \vdash B(x)$		
Sigma type	$\sum_{x:A} B(x)$	$A \times B$	(a, b)
Product type	$\prod_{x:A} B(x)$	$A \to B$	$\lambda(x : A).b$
Coproduct type	$A + B$		$\mathsf{inl}(a)$, $\mathsf{inr}(b)$
Identity type	$a = b$		$\mathsf{refl}(a) : a = a$
Universe	$\mathscr{U}$		
Base types	Nat, Bool, 1, 0		

Identity vs equality

Inhabitants of $x = y$ behave like equality in many ways

- $\mathsf{refl}(x) : x = x$
- $\mathsf{sym} : x = y \to y = x$
- $\mathsf{trans} : x = y \times y = z \to x = z$

Transport

$$\mathsf{transport} : x = y \to \prod_{B:A\to\mathcal{U}} \big(B(x) \simeq B(y)\big)$$

Inhabitants of $x = y$ behave **un**like equality

- Can iterate identity type
- Cannot show that any two identities are identical

The important features of univalent foundations

Homotopy levels

- Stratification of types according to "complexity" of their identity types
- Logic: notion of **propositions** given by one layer of this hierarchy

Univalence axiom

Specifies the identity type of a universe:

$$(X = Y) \to (X \simeq Y)$$
$$\mathsf{refl}(X) \mapsto 1_X$$

is an equivalence

Contractible types, propositions and sets

- A is **contractible**

$$\mathsf{isContr}(A) :\equiv \sum_{x:A} \prod_{y:A} y = x$$

- A is a **proposition**

$$\mathsf{isProp}(A) :\equiv \prod_{x,y:A} x = y$$

- A is a **set**

$$\mathsf{isSet}(A) :\equiv \prod_{x,y:A} \mathsf{isProp}(x = y)$$

$$\mathsf{Prop} :\equiv \sum_{X:\mathcal{U}} \mathsf{isProp}(X) \qquad \mathsf{Set} :\equiv \sum_{X:\mathcal{U}} \mathsf{isSet}(X)$$

Contractible types, propositions and sets

- A is **contractible**

$$\mathsf{isContr}(A) \;:\equiv\; \sum_{x:A} \prod_{y:A} y = x$$

- A is a **proposition**

$$\mathsf{isProp}(A) \;:\equiv\; \prod_{x,y:A} \mathsf{isContr}(x = y)$$

- A is a **set**

$$\mathsf{isSet}(A) \;:\equiv\; \prod_{x,y:A} \mathsf{isProp}(x = y)$$

$$\mathsf{Prop} \;:\equiv\; \sum_{X:\mathcal{U}} \mathsf{isProp}(X) \qquad \mathsf{Set} \;:\equiv\; \sum_{X:\mathcal{U}} \mathsf{isSet}(X)$$

Equivalences

Definition

A map $f : A \to B$ is an **equivalence** if it has contractible fibers, i.e.,

$$\mathsf{isEquiv}(f) \quad :\equiv \quad \prod_{b:B} \mathsf{isContr}\left(\sum_{a:A} f(a) = b\right)$$

The type of equivalences:

$$A \simeq B \quad :\equiv \quad \sum_{f:A\to B} \mathsf{isEquiv}(f)$$

Outline

Different notions of equality

Synthetic vs. analytic equalities

In MLTT, we always have a **synthetic** equality type between $a, b : T$

$$a =_T b.$$

Depending on T, we might have a type of **analytic** equalities

$$a \simeq_T b.$$

Univalence Principle for T and $\simeq_T$ says that this map is an equivalence

$$(a =_T b) \to (a \simeq_T b)$$

Different notions of equality

Synthetic vs. analytic equalities

In MLTT, we always have a **synthetic** equality type between $a, b : T$

$$a =_T b.$$

Depending on T, we might have a type of **analytic** equalities

$$a \simeq_T b.$$

Univalence Principle for T and $\simeq_T$ says that this map is an equivalence

$$(a =_T b) \to (a \simeq_T b)$$

Univalence **Axiom**: for $T \equiv \mathcal{U}$ and $(X \simeq_{\mathcal{U}} Y) \equiv$ "equivalences $X \to Y$":

$$(X =_{\mathcal{U}} Y) \to (X \simeq_{\mathcal{U}} Y)$$

is an equivalence.

Transport along equivalence of types

$$x =_{\mathcal{U}} y \to \prod_{(P:\mathcal{U}\to\mathcal{U})} (P(x) \simeq P(y))$$

$$\text{univalence} : \prod_{(x,y:\mathcal{U})} (x =_{\mathcal{U}} y \xrightarrow{\sim} x \simeq y)$$

$$x \simeq y \to \prod_{(P:\mathcal{U}\to\mathcal{U})} (P(x) \simeq P(y))$$

Transport along isomorphism of groups

$$x =_{\mathsf{Grp}} y \to \prod_{(P:\mathsf{Grp}\to\mathcal{U})} (P(x) \simeq P(y))$$

$$\text{univalence} : \prod_{(x,y:\mathsf{Grp})} (x =_{\mathsf{Grp}} y \rightharpoonup x \cong y)$$

$$x \cong y \to \prod_{(P:\mathsf{Grp}\to\mathcal{U})} (P(x) \simeq P(y))$$

Transport along isomorphism of groups

$$x =_{\mathsf{Grp}} y \to \prod_{(P:\mathsf{Grp}\to\mathcal{U})} (P(x) \simeq P(y))$$

$$\mathsf{univalence} : \prod_{(x,y:\mathsf{Grp})} (x =_{\mathsf{Grp}} y \rightharpoonup x \cong y)$$

$$x \cong y \to \prod_{(P:\mathsf{Grp}\to\mathcal{U})} (P(x) \simeq P(y))$$

Structure Identity Principle

- One can show $(x =_{\mathsf{Grp}} y \rightharpoonup x \cong y)$, using univalence for types
- Works similarly for many other structures **built from (types that are) sets**
- See Coquand & Danielsson and HoTT book (Section 9.9)

Transport along isomorphism of groups

$$x =_{\mathsf{Grp}} y \to \prod_{(P:\mathsf{Grp}\to\mathcal{U})} (P(x) \simeq P(y))$$

$$\mathsf{univalence} : \prod_{(x,y:\mathsf{Grp})} (x =_{\mathsf{Grp}} y \rightharpoonup x \cong y)$$

$$x \cong y \to \prod_{(P:\mathsf{Grp}\to\mathcal{U})} (P(x) \simeq P(y))$$

Structure Identity Principle

- One can show $(x =_{\mathsf{Grp}} y \rightharpoonup x \cong y)$, using univalence for types
- Works similarly for many other structures **built from (types that are) sets**
- See Coquand & Danielsson and HoTT book (Section 9.9)

What about things that form a **higher** category, e.g., categories themselves?

Categories in type theory

A **category** $\mathscr{C}$ is given by

- a type $\mathscr{C}_0 : \mathscr{U}$ of **objects**
- for any $a, b : \mathscr{C}_0$, a set $\mathscr{C}(a,b) : \mathscr{U}$ of **morphisms**
- operations: identity & composition

$$1_a : \mathscr{C}(a,a)$$
$$(\circ)_{a,b,c} : \mathscr{C}(b,c) \to \mathscr{C}(a,b) \to \mathscr{C}(a,c)$$

- axioms: unitality & associativity

$$1 \circ f = f \qquad f \circ 1 = f \qquad (h \circ g) \circ f = h \circ (g \circ f)$$

Categories in type theory

A **category** $\mathscr{C}$ is given by

- a type $\mathscr{C}_0 : \mathscr{U}$ of **objects**
- for any $a, b : \mathscr{C}_0$, a set $\mathscr{C}(a,b) : \mathscr{U}$ of **morphisms**
- operations: identity & composition

$$1_a : \mathscr{C}(a,a)$$
$$(\circ)_{a,b,c} : \mathscr{C}(b,c) \to \mathscr{C}(a,b) \to \mathscr{C}(a,c)$$

- axioms: unitality & associativity

$$1 \circ f = f \qquad f \circ 1 = f \qquad (h \circ g) \circ f = h \circ (g \circ f)$$

A **univalent category** is a category $\mathscr{C}$ such that

$$(a = b) \to (a \cong b)$$

is an equivalence for all $a, b : \mathscr{C}_0$.

Local univalence implies global univalence

Theorem (A., Kapulkin, Shulman)

For categories A and B, let $A \simeq B$ denote the type of equivalences from A to B. If A and B are ***univalent****, we have*

$$(A =_{\mathsf{uCat}} B) \simeq (A \simeq B).$$

Transport along equivalence of **univalent** categories

$$x =_{\mathsf{uCat}} y \to \prod_{(P:\mathsf{uCat}\to\mathscr{U})} (P(x) \simeq P(y))$$

$$\mathsf{univalence} : \prod_{(x,y:\mathsf{uCat})} (x =_{\mathsf{uCat}} y \xrightarrow{\sim} x \simeq y)$$

$$x \simeq y \to \prod_{(P:\mathsf{uCat}\to\mathscr{U})} (P(x) \simeq P(y))$$

Transport along equivalence of **univalent** categories

$$x =_{\mathsf{uCat}} y \to \prod_{(P:\mathsf{uCat}\to\mathscr{U})} (P(x) \simeq P(y))$$

$$\mathsf{univalence} : \prod_{(x,y:\mathsf{uCat})} (x =_{\mathsf{uCat}} y \xrightarrow{\sim} x \simeq y)$$

$$x \simeq y \to \prod_{(P:\mathsf{uCat}\to\mathscr{U})} (P(x) \simeq P(y))$$

Univalence Principle for categories

- Holds only for categories that satisfy themselves a univalence condition: **local univalence implies global univalence**
- Univalent categories are the right notion of categories in Univalent Foundations

Univalence Principle (joint work with North, Shulman, Tsementzis

1. Define **signature**, **axiom**, and **theory** for mathematical structures, including higher-categorical ones
2. Given a theory $\mathscr{T} = (\mathscr{L}, T)$, define
   - $\mathscr{T}$-models
   - Univalence of $\mathscr{T}$-models
   - Equivalence between $\mathscr{T}$-models
3. Prove a univalence result for univalent $\mathscr{T}$-models:

$$\mathsf{univalence} : \prod_{(x,y:\mathsf{uMod}_{\mathscr{T}})} ((x =_{\mathsf{uMod}_{\mathscr{T}}} y) \rightharpoonup (x \simeq_{\mathsf{uMod}_{\mathscr{T}}} y))$$

Other examples

- First-order logic, e.g., groups, rings
- Higher-order logic, e.g., topological spaces, suplattices
- Categories
- Dagger categories
- (Ana)functors
- Profunctors
- Displayed categories / Fibrations
- Bicategories
- Double categories
- ...

Univalence Principle

Technical challenge

Define a notion of "isomorphism" (called **indiscernibility**) that

1. works for any signature/theory
2. specializes to categorical isomorphism for the theory of categories

In the rest of the talk

1. Show what the local univalence condition means for 1-categorical structures
2. Our example: monoids

Outline

Monoids in type theory

In type theory, a monoid is a tuple $(M, \mu, e, \alpha, \lambda, \rho)$ where

1. $M : \mathbf{Set}$
2. $\mu : M \times M \to M$
3. $e : M$
4. $\alpha : \Pi_{(a,b,c:M)} \mu(\mu(a,b),c) = \mu(a,\mu(b,c))$
5. $\lambda : \Pi_{(a:M)} \mu(e,a) = a$
6. $\rho : \Pi_{(a:M)} \mu(a,e) = a$

Monoids in type theory

In type theory, a monoid is a tuple $(M, \mu, e, \alpha, \lambda, \rho)$ where

1. $M : \mathbf{Set}$
2. $\mu : M \times M \to M$
3. $e : M$
4. $\alpha : \Pi_{(a,b,c:M)} \mu(\mu(a,b),c) = \mu(a,\mu(b,c))$
5. $\lambda : \Pi_{(a:M)} \mu(e,a) = a$
6. $\rho : \Pi_{(a:M)} \mu(a,e) = a$

Why $M : \mathbf{Set}$?

Monoids in type theory

In type theory, a monoid is a tuple $(M, \mu, e, \alpha, \lambda, \rho)$ where

1. $M : \mathbf{Set}$
2. $\mu : M \times M \to M$
3. $e : M$
4. $\alpha : \Pi_{(a,b,c:M)} \mu(\mu(a,b),c) = \mu(a,\mu(b,c))$
5. $\lambda : \Pi_{(a:M)} \mu(e,a) = a$
6. $\rho : \Pi_{(a:M)} \mu(a,e) = a$

Why $M : \mathbf{Set}$?

Abstractly, a monoid is a (dependent) pair $(data, proof)$ where

- $data$ is 1.–3.
- $proof$ is 4.–6.

The type of monoids

- We want two monoids $(data, proof)$ and $(data', proof')$ to be the same if $data$ is the same as $data'$.
- This is guaranteed when the types of $proof$ and $proof'$ are **propositions**.
- This in turn is guaranteed when M is a **set**.

Summarily:

$$\mathsf{Monoid} :\equiv \sum_{(M:\mathsf{Set})} \sum_{(\mu,e):\mathsf{MonoidStr}(M)} \mathsf{MonoidAxioms}(M,(\mu,e))$$

Can show

$$\mathsf{isProp}(\mathsf{MonoidAxioms}(M,(\mu,e)))$$

Monoid isomorphisms

Given $\mathbf{M} \equiv (M, \mu, e, \alpha, \lambda, \rho)$ and $\mathbf{M}' \equiv (M', \mu', e', \alpha', \lambda', \rho')$, a **monoid isomorphism** is a bijection $f : M \cong M'$ preserving μ and e.

Monoid isomorphisms

Given $\mathbf{M} \equiv (M, \mu, e, \alpha, \lambda, \rho)$ and $\mathbf{M}' \equiv (M', \mu', e', \alpha', \lambda', \rho')$, a **monoid isomorphism** is a bijection $f : M \cong M'$ preserving μ and e.

$$
\begin{aligned}
\mathbf{M} = \mathbf{M}' &\simeq (M, \mu, e) = (M', \mu', e') \\
&\simeq \sum_{p : M = M'} (\mathsf{transport}^{Y \mapsto (Y \times Y \to Y)}(p, \mu) = \mu') \\
&\qquad \times (\mathsf{transport}^{Y \mapsto Y}(p, e) = e') \\
&\simeq \sum_{f : M \cong M'} \left(f \circ \mu \circ (f^{-1} \times f^{-1}) = \mu'\right) \\
&\qquad \times (f \circ e = e') \\
&\simeq \mathbf{M} \cong \mathbf{M}'
\end{aligned}
$$

Transport along monoid isomorphism

We now have two ingredients:

1. $$\mathsf{transport}_{\mathbf{M},\mathbf{M}'} : (\mathbf{M} = \mathbf{M}') \to \prod_{B:\mathsf{Monoid}\to\mathcal{U}} \left(B(\mathbf{M}) \simeq B(\mathbf{M}')\right)$$

2. $$(\mathbf{M} = \mathbf{M}') \quad \simeq \quad (\mathbf{M} \cong \mathbf{M}')$$

Composing these, we get

$$\mathsf{transport}_{\mathbf{M},\mathbf{M}'} : (\mathbf{M} \cong \mathbf{M}') \to \prod_{B:\mathsf{Monoid}\to\mathcal{U}} \left(B(\mathbf{M}) \simeq B(\mathbf{M}')\right)$$

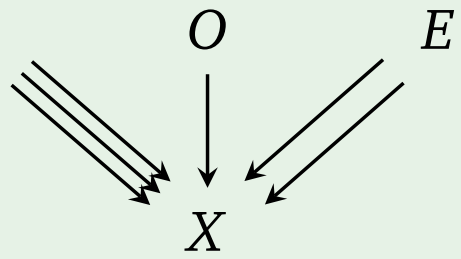
Outline

The signature of monoids

Example

Signature $\mathscr{L}_{\text{mon}}$ for a monoid:

$$\mathscr{L}_{\text{mon}} = \quad \mu \quad O \quad E \quad X$$



A **structure** M for this signature consists of

1. a type MX
2. a family of types $M\mu(x,y,z)$ for $x,y,z : MX$
3. a family of types $MO(x)$ for $x : MX$
4. a family of types $ME(x,y)$ for $x,y : MX$

The theory of monoids

Not all structures represent monoids. **Axioms** specify those structures that are a monoid:

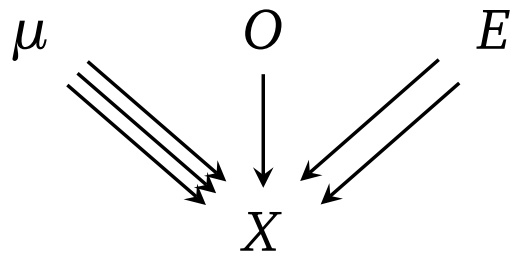
Axioms of a monoid

1. Monoid axioms:

$$\forall(x,y,z,z' : X).\mu(x,y,z) \to \mu(x,y,z') \to E(z,z')$$
$$\forall(x,y : X).\exists(z : X).\mu(x,y,z)$$
$$\forall(x,x',y,z : X).E(x,x') \to \mu(x,y,z) \to \mu(x',y,z)$$
$$\forall(x,y : X).E(x,y) \to E(y,x)$$
$$\dots$$

2. "Homotopical axioms":
   - 2.1 MX is a set
   - 2.2 $M\mu(x,y,z)$, $MO(x)$, $ME(x,y)$ are pointwise propositions

Indiscernibility for elements of a monoid



Given $a, b : MX$, an **indiscernibility** $a \asymp b$ consists of "equivalences of types of everything above a and b"

$$M\mu(a,y,z) \simeq M\mu(b,y,z)$$
$$M\mu(x,a,z) \simeq M\mu(x,b,z)$$
$$M\mu(x,y,a) \simeq M\mu(x,y,b)$$
$$M\mu(a,a,z) \simeq M\mu(b,b,z)$$
$$\dots$$
$$MO(a) \simeq MO(b)$$
$$ME(a,y) \simeq ME(b,y)$$
$$\dots$$

Indiscernibility

1. $a \asymp b$ means that a and b behave in the same way within the structure.
2. In a model M of the theory of monoids, $a \asymp b$ reduces to $ME(a, b)$.
3. Definition of indiscernibility carries over to any $\mathscr{L}$, and any sort in $\mathscr{L}$.

Indiscernibility

1. $a \asymp b$ means that a and b behave in the same way within the structure.
2. In a model M of the theory of monoids, $a \asymp b$ reduces to $ME(a,b)$.
3. Definition of indiscernibility carries over to any $\mathscr{L}$, and any sort in $\mathscr{L}$.

Definition

1. Given $v, w : M\mu(a,b,c)$, an indiscernibility $v \asymp w$ is given by an equivalence

$$\mathbf{1} \simeq \mathbf{1}$$

   (since there is nothing above μ in $\mathscr{L}_{\text{mon}}$). Hence $(v \asymp w) = \mathbf{1}$.
2. Similar for $v, w : MO(a)$, and $v, w : ME(a,b)$.

Univalence of models

Definition

A monoid M is **univalent** if the maps

$$(a = b) \to (a \asymp b) \tag{1}$$

$$(v = w) \to (v \asymp w) \tag{2}$$

are equivalences for $a, b : MX$ and $v, w : M\mu, MO, ME$.

Univalence of models

Definition

A monoid M is **univalent** if the maps

$$(a = b) \to (a \asymp b) \tag{1}$$

$$(v = w) \to (v \asymp w) \tag{2}$$

are equivalences for $a, b : MX$ and $v, w : M\mu, MO, ME$.

Observations

1. Since $(w \asymp w') = 1$, condition (2) is equivalent to $M\mu$, MO, and ME being pointwise **propositions**.
2. Since $(a \asymp b) = ME(a, b)$, condition (1) is equivalent to M being a **set with identity** $a = b$ **given by** $E(a, b)$.

Univalence of models

Definition

A monoid M is **univalent** if the maps

$$(a = b) \to (a \asymp b) \tag{1}$$

$$(v = w) \to (v \asymp w) \tag{2}$$

are equivalences for $a, b : MX$ and $v, w : M\mu, MO, ME$.

Observations

1. Since $(w \asymp w') = 1$, condition (2) is equivalent to $M\mu$, MO, and ME being pointwise **propositions**.
2. Since $(a \asymp b) = ME(a, b)$, condition (1) is equivalent to M being a **set with identity** $a = b$ **given by** $E(a, b)$.

Summary: a univalent monoid in this sense is exactly the same a monoid as previously defined.

Equivalence of models

Given monoids M, N, an **equivalence** is

$$e_X : MX \to NX$$

$$e_\mu : \prod_{x,y,z:MX} M\mu(x,y,z) \to N\mu(ex,ey,ez)$$

$$e_O : \prod_{x:MX} MO(x) \to NO(ex)$$

$$e_E : \prod_{x,y:MX} : ME(x,y) \to NE(ex,ey)$$

such that e_X, $(e_\mu)_{x,y,z}$, $(e_O)_x$, and $(e_E)_{x,y}$ are (split-)surjective.

Observations

1. For univalent monoids, condition of e_E being split-surjective entails that e_X is injective.
2. An equivalence of univalent monoids is an isomorphism of sets that preserves multiplication and unit.

Univalence for univalent monoids

Theorem (Univalence Principle)

For univalent M and N,

$$(M = N) \simeq (M \simeq N)$$

Outline

Where does our work take place?

Working in **two-level type theory** of Annenkov, Capriotti, Kraus, Sattler.

- Univalent Foundations, embedded in an extensional type theory
- Universes $\mathscr{U} \hookrightarrow \mathscr{U}^s$
- $\mathscr{U}$ implements univalent type theory.
- Every type $T : \mathscr{U}^s$ is equipped with a strict equality type $a \equiv_T b$ with the usual rules for the identity type, but which also satisfies UIP.
- Signatures live in $\mathscr{U}^s$ (are meta-mathematical), but models and their morphisms live in $\mathscr{U}$ (are mathematical)

Transporting in practice

Limitations of our work

- Not formalized in a computer proof assistant (WIP by Elif Uskuplu)
- No functions, only relations

Ingredients for transport in a computer proof assistant

1. Mathematical foundation satisfying the univalence axiom (by proof rather than by axiom)
2. Implementation of these foundations as a computer proof assistant
3. Univalence principle for mathematical structures

Transporting in practice

Limitations of our work

- Not formalized in a computer proof assistant (WIP by Elif Uskuplu)
- No functions, only relations

Ingredients for transport in a computer proof assistant

1. Mathematical foundation satisfying the univalence axiom (by proof rather than by axiom)
2. Implementation of these foundations as a computer proof assistant
3. Univalence principle for mathematical structures

Thanks for your attention!

References

- Coquand, Danielsson, “Isomorphism is equality”
- The HoTT book (Section 9.9 for Structure Identity Principle)
- Ahrens, Kapulkin, Shulman, “Univalent categories and the Rezk completion”
- Ahrens, North, Shulman, Tsementzis, “The Univalence Principle”