Set-level mathematics

Carlo Angiuli
(adapted from slides of Benedikt Ahrens)

School on Univalent Mathematics, Minneapolis, 2024

You are here

Lectures 1–4: basics of Univalent Foundations and Coq.

Lectures 5–7: developing mathematics in UF.

In Lecture 3, Paige introduced UF, hlevels, propositions, and sets.

This lecture is a deeper dive into sets in UF.

Outline

1. Reminder: homotopy levels and sets
2. What types are (or aren't) sets?
3. Algebraic structures
4. Subtypes, relations, set-level quotients

Outline

1. Reminder: homotopy levels and sets
2. What types are (or aren't) sets?
3. Algebraic structures
4. Subtypes, relations, set-level quotients

Homotopy levels

$$\mathsf{isofhlevel} : \mathsf{Nat} \to \mathsf{Type} \to \mathsf{Type}$$

$$\mathsf{isofhlevel}(0)(X) \quad :\equiv \quad \mathsf{isContr}(X)$$

$$\mathsf{isofhlevel}(S(n))(X) \quad :\equiv \quad \prod_{x,y:X} \mathsf{isofhlevel}(n, x = y)$$

X has exactly one element

Homotopy levels

$$\mathsf{isofhlevel} : \mathsf{Nat} \to \mathsf{Type} \to \mathsf{Type}$$

$$\mathsf{isofhlevel}(0)(X) :\equiv \mathsf{isContr}(X)$$

$$\mathsf{isofhlevel}(S(n))(X) :\equiv \prod_{x,y:X} \mathsf{isofhlevel}(n, x = y)$$

X has exactly one element

$$\mathsf{isProp}(X) :\equiv \mathsf{isofhlevel}(1)(X)$$

$$\equiv \prod_{x,y:X} \mathsf{isContr}(x = y)$$

$$\mathsf{Prop} :\equiv \sum_{X:\mathsf{Type}} \mathsf{isProp}(X)$$

types with at most one element (proof)

Definition

$$\mathsf{isSet}(X) :\equiv \mathsf{isofhlevel}(2)(X)$$
$$\equiv \prod_{x,y:X} \mathsf{isProp}(x = y)$$
$$\mathsf{Set} :\equiv \sum_{X:\mathsf{Type}} \mathsf{isSet}(X)$$

at most one path (equality) between any two elements

Sets

Sets are the types which satisfy Uniqueness of Identity Proofs:

- If $p, q : x = y$ then $p = q$.
- In particular, if $p : x = x$ then $p = \mathsf{refl}(x)$.

But working with sets in UF is not the same as working in a type theory with global UIP.

Outline

Decidable equality

Definition

A type X is **decidable** if

$$X + \neg X$$

Definition

A type X has **decidable equality** if all of its path types are decidable, i.e.,

$$\prod_{x,y:A} (x = y) + \neg(x = y)$$

Hedberg's theorem

Theorem (Hedberg)

If a type X has decidable equality, then X is a set.

Exercise

Prove that Bool and Nat have decidable equality.

Closure properties

- $\sum_{x:A} B(x)$ is a set if A and all $B(x)$ are
- $A \times B$ is a set if A and B are
- $\prod_{x:A} B(x)$ is a set if all $B(x)$ are
- $A \to B$ is a set if B is
- A is a set if it is a proposition

Exercise

Do you know

- a type that is a set?
- a type for which you don't know (yet) whether it is a set?
- a type for which you know it is not a set?

Another set

$\sum_{X:\mathsf{Type}} \mathsf{isProp}(X)$

Exercise

Prove that Prop is a set.

The proof relies on the univalence axiom for the universe Type.

(Note that Prop does **not** have decidable equality.)

Non-sets

Is there a type that is not a set?

- In Martin-Löf type theory, some types can *not* be shown to be sets.
- In univalent type theory, some types can be shown to *not* be sets.

Suppose that Type is a univalent universe containing the type Bool.

Exercise

Prove that Type is not a set.

(Which property of Bool does the proof of the above result exploit?)

Exercise

Prove that Set is not a set. (What hlevel does it have, if any?)

Sets and propositions

It is often useful to ensure that types intended to capture "properties" are propositions.

Definition

An **even natural number** is a term of type

$$\mathsf{Even} \quad :\equiv \quad \sum_{n:\mathsf{Nat}} \mathsf{iseven}(n)$$

i.e., a pair of a natural number $n : \mathsf{Nat}$ and a proof p that n is even.

When are two Evens equal? Hopefully, when they are equal as Nats:

$$(n,p) = (n',p') \quad \simeq \quad n = n'$$

Reminder: paths between pairs

Given $B : A \to \mathsf{Type}$ and $a, a' : A$ and $b : B(a)$ and $b' : B(a')$,

$$(a,b) = (a',b') \quad \simeq \quad \sum_{p : a = a'} \mathsf{transport}^B\big(p, b\big) = b'$$

Exercise

If $B(x)$ is a proposition for all $x : A$, then this can be simplified to:

$$(a,b) = (a',b') \quad \simeq \quad a = a'$$

By the above, Evens will have the "right" notion of equality if $\mathsf{iseven}(n)$ is a proposition.

Sets and propositions

One important property of sets is that equations in sets are propositional (hence "properties"). But in general, one must be careful about equational conditions. . .

Example

Given $f : X \to Y$,

$$\mathsf{isInjective}(f) \quad :\equiv \quad \prod_{x,x':X} f(x) = f(x') \to x = x'$$

is **not** a proposition in general, but it is if X is a set.

Exercise

Define $\mathsf{isInjective}(f)$ in a such a way that it is a proposition for X and Y of any level.

Isomorphism vs. equivalence

Example

Given $f : X \to Y$,

$$\mathsf{isiso}(f) \quad :\equiv \quad \sum_{g:Y\to X} (g \circ f = 1_X) \times (f \circ g = 1_Y)$$

is **not** a proposition in general, but it is if X and Y are sets.

Warning

Stating the univalence axiom with isomorphisms instead of equivalences yields an inconsistency.

But, when X and Y are sets, then $\mathsf{isiso}(f) \simeq \mathsf{isequiv}(f)$.

Outline

Monoids

In set theory, a **monoid** is a triple (M, μ, e) of

- a set M
- a multiplication $\mu : M \times M \to M$
- a unit $e \in M$

satisfying the axioms of associativity, left neutrality, and right neutrality.

(Many examples, such as the natural numbers or integers with $\mu =$ addition, $e = 0$.)

Monoids in type theory

In type theory, a **monoid** is a 6-tuple $(M, \mu, e, \alpha, \lambda, \rho)$ of

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

Monoids in type theory

In type theory, a **monoid** is a 6-tuple $(M, \mu, e, \alpha, \lambda, \rho)$ of

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

Why $M : \mathsf{Set}$ instead of $M : \mathsf{Type}$?

The type of monoids

When are two monoids equal? We want equality of monoids to depend only on M and the data (μ, e), not on the proofs (α, λ, ρ) that the data satisfy the monoid axioms.

Reparenthesizing as $(M, (\mu, e), (\alpha, \lambda, \rho))$,

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

we see:

- This is ensured if the type $\mathsf{MonoidAxioms}(M, (\mu, e))$ is a **proposition**.
- This is in turn guaranteed as long as M is a **set**, hence the restriction.

Monoid isomorphisms

Definition

A **monoid isomorphism** between $\mathbf{M} \equiv (M, \mu, e, \alpha, \lambda, \rho)$ and $\mathbf{M}' \equiv (M', \mu', e', \alpha', \lambda', \rho')$ is an isomorphism of sets $f : M \cong M'$ which preserves the multiplication/unit, i.e.,

$$\prod_{a,b:M} f(\mu(a,b)) = \mu'(f(a), f(b))$$

$$f(e) = e'$$

Exercise

The type of equalities $\mathbf{M} = \mathbf{M}'$ of monoids is equivalent to the type of monoid isomorphisms $\mathbf{M} \cong \mathbf{M}'$.

Monoid isomorphisms

Proof sketch:

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

Transport along monoid isomorphisms

We now have two ingredients:

1. $(\mathbf{M} = \mathbf{M}') \quad \simeq \quad (\mathbf{M} \cong \mathbf{M}')$
2. $\mathsf{transport}^T : (\mathbf{M} = \mathbf{M}') \to T(\mathbf{M}) \to T(\mathbf{M}')$ for any $T : \mathsf{Monoid} \to \mathsf{Type}$

Combining these, we get

$$(\mathbf{M} \cong \mathbf{M}') \to T(\mathbf{M}) \to T(\mathbf{M}')$$

In other words, any property or structure on monoids expressible in UF can be transported along isomorphism of monoids.

Example

If $\mathbf{M}$ is commutative and $\mathbf{M} \cong \mathbf{M}'$, then $\mathbf{M}'$ is commutative.
(Regardless of what commutative means!)

Structure Identity Principle

This is known as the *Structure Identity Principle* (Coquand, Aczel):

> *Isomorphic mathematical structures are equal as structured types,*
> *and hence have the same structural properties.*

The Structure Identity Principle holds in Univalent Foundations for many algebraic structures; isomorphic such structures have **all** the same (definable) properties.

Outline

Predicates on types

A **subtype** A on a type X is a (prop-valued) predicate, i.e., a map

$$A : X \to \mathsf{Prop}$$

Exercise

Prove that the type of subtypes of X is a set.

The **carrier** of a subtype A is the type of elements of X satisfying A:

$$\mathsf{carrier}(A) :\equiv \sum_{x:X} A(x)$$

Relations on a type

A **binary relation** R on a type X is a map

$$R : X \to X \to \mathsf{Prop}$$

Exercise

Prove that the type of binary relations on X is a set.

Properties of such relations are defined as usual, e.g.,

$$\mathsf{reflexive}(R) \quad :\equiv \quad \prod_{x:X} R(x)(x)$$

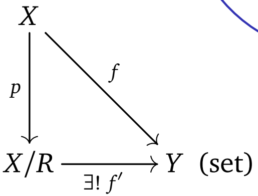
Exercise

Formulate the properties of being symmetric, transitive, and an equivalence relation.

Set-level quotient

The **quotient** of a type X by an equivalence relation R on X is a pair $(X/R, p)$ of a type X/R and a map $p : X \to X/R$ such that any R-compatible map f **into a set** Y factors uniquely via p:

$$\prod_{x,y:A} R\,x\,y \to f(x) = f(y)$$



The quotient $(X/R, p)$ is unique if it exists. MLTT does not have quotients in general, but in UF we can define them as the set of equivalence classes of R, as in set theory.

The quotient set

Definition

A subtype $A : X \to \mathsf{Prop}$ is an **equivalence class of** R if

$$\mathsf{iseqclass}(A,R) \quad :\equiv \quad ||\mathsf{carrier}(A)|| \times \left(\prod_{x,y:X} Rxy \to Ax \to Ay\right) \times \left(\prod_{x,y:X} Ax \to Ay \to Rxy\right)$$

Then we may define:

$$X/R \quad :\equiv \sum_{A:X\to\mathsf{Prop}} \mathsf{iseqclass}(A,R)$$

Exercise

Define $p : X \to X/R$, prove that X/R is a set, and prove that $(X/R, p)$ has the universal property of a quotient.