

# Impredicative Encodings in HoTT (or: Toward a Realizability $\infty$ -Topos)

Steve Awodey  
Carnegie Mellon University

Big Proofs  
Issac Newton Institute  
Cambridge, 11 July 2017

# Overview and Acknowledgements

- ▶ I will sketch some **work in progress** on impredicative encodings of inductive types in Homotopy Type Theory.
- ▶ Some parts are **joint work** with others from the CMU HoTT Group, particularly Jonas Frey and Pieter Hofstra, and our excellent PhD students Floris van Doorn, Clive Newstead, Egbert Rijke, and Sam Speight.
- ▶ For the model, we are **building on** recent work of Coquand et al., Gambino & Sattler, Orton & Pitts, and others, as well as some “classical” results.

# Outline

- I. Basic Ideas of Homotopy Type Theory
- II. Impredicative Encodings
- III. A Cubical Realizability Model

# I. Basic HoTT

- ▶ Homotopy Type Theory is based on a recently discovered connection between Logic and Topology.
- ▶ The system of intensional Martin-Löf type theory (constructive foundations) can be interpreted into abstract homotopy theory (mathematics of space).
- ▶ This permits computerized proof systems based on MLTT to be used to formalize higher mathematical reasoning.
- ▶ It also suggests new logical principles, such as the univalence axiom and higher inductive types.
- ▶ Higher inductive types are used to add some basic spaces like the spheres  $S^n$  and constructions like quotient types  $X/\sim$ .
- ▶ However, simply adding these new principles as axioms lacks a computational justification.

# Dependent Type Theory (Howard, Martin-Löf, Tait, ...)

Dependent type theory consists of:

- ▶ **Types:**  $X, Y, \dots, A \times B, A \rightarrow B, \dots$
- ▶ **Terms:**  $x : A, b : B, \langle a, b \rangle, \lambda x. b(x), \dots$
- ▶ **Dependent Types:**  $x : A \vdash B(x)$ 
  - ▶  $\sum_{x:A} B(x)$
  - ▶  $\prod_{x:A} B(x)$
- ▶ **Equations**  $s = t : A$

Formal calculus of typed terms and equations, presented as a deductive system by rules of inference.

Intended as a foundation for constructive mathematics, but now also widely used in programming languages and computerized proof assistants.

# Propositions as Types (Curry, Howard, Scott, ...)

The system has a dual interpretation:

- ▶ once as **mathematical** objects: types are “sets” and their terms are “elements”, which are being constructed,
- ▶ once as **logical** objects: types are “propositions” and their terms are “proofs”, which are being derived.

This is known as the **Curry-Howard correspondence**:

0	1	$A + B$	$A \times B$	$A \rightarrow B$	$\sum_{x:A} B(x)$	$\prod_{x:A} B(x)$
$\perp$	$\top$	$A \vee B$	$A \wedge B$	$A \Rightarrow B$	$\exists_{x:A} B(x)$	$\forall_{x:A} B(x)$

Gives the system a **constructive character**.

## Identity types (Martin-Löf, Lawvere)

It's natural to add a primitive **identity type** between terms of the same type,  $x, y : A$ :

$$\text{Id}_A(x, y)$$

**Logically** this is the proposition “ $x = y$ ”.

0	1	$A + B$	$A \times B$	$A \rightarrow B$	$\sum_{x:A} B(x)$	$\prod_{x:A} B(x)$	$\text{Id}_A(x, y)$
$\perp$	$\top$	$A \vee B$	$A \wedge B$	$A \Rightarrow B$	$\exists_{x:A} B(x)$	$\forall_{x:A} B(x)$	$x = y$

Terms that are “identified” may remain distinct syntactically. This “intensionality” gives the system good computational properties.

But what is  $\text{Id}_A(x, y)$  **mathematically**? What are the **terms**  $p : \text{Id}_A(x, y)$  of these new types? Can **they** differ?

## The homotopy interpretation (Awodey-Warren)

Suppose we have terms of ascending identity types:

$$a, b : A$$

$$p, q : \text{Id}_A(a, b)$$

$$\alpha, \beta : \text{Id}_{\text{Id}_A(a,b)}(p, q)$$

$$\dots : \text{Id}_{\text{Id}_{\text{Id}}\dots}(\dots)$$

Consider the following interpretation:

Types	$\rightsquigarrow$	Spaces
Terms	$\rightsquigarrow$	Maps
$a : A$	$\rightsquigarrow$	Points $a : 1 \rightarrow A$
$p : \text{Id}_A(a, b)$	$\rightsquigarrow$	Paths $p : a \Rightarrow b$
$\alpha : \text{Id}_{\text{Id}_A(a,b)}(p, q)$	$\rightsquigarrow$	Homotopies $\alpha : p \Rrightarrow q$
$\vdots$		



# The homotopy interpretation (Awodey-Warren)

This takes the familiar **topological interpretation** of the *simply-typed*  $\lambda$ -calculus:

types  $\rightsquigarrow$  spaces

terms  $\rightsquigarrow$  continuous functions

and extends it via the **basic idea**:

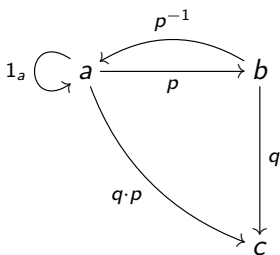
$p : \text{Id}_X(a, b) \rightsquigarrow p$  is a path from point  $a$  to point  $b$  in  $X$

This then **forces**:

- ▶ dependent types to be fibrations,
- ▶ Id-types to be path spaces,
- ▶ homotopic maps to be identical.

# The fundamental groupoid of a type (Hofmann-Streicher)

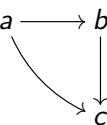
In topology, the points and paths in any space bear the structure of a **groupoid**: a category in which every arrow has an inverse.



In the same way the **terms**  $a, b, c : X$  and **identity terms**  $p : \text{Id}_X(a, b)$  and  $q : \text{Id}_X(b, c)$  of any type  $X$  also form a groupoid.

# The fundamental groupoid of a type (Hofmann-Streicher)

The provable laws of identity provide the **groupoid operations**:

$r : \text{Id}(a, a)$	reflexivity	$a \longrightarrow a$
$s : \text{Id}(a, b) \rightarrow \text{Id}(b, a)$	symmetry	$a \overset{\curvearrowright}{\longleftarrow} b$
$t : \text{Id}(a, b) \times \text{Id}(b, c) \rightarrow \text{Id}(a, c)$	transitivity	$a \longrightarrow b$ 

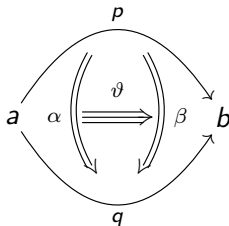
But as in topology, the **groupoid equations**:

$p \cdot (q \cdot r) = (p \cdot q) \cdot r$	associativity
$p^{-1} \cdot p = 1 = p \cdot p^{-1}$	inverse
$1 \cdot p = p = p \cdot 1$	unit

do not hold **strictly**, but only “**up to homotopy**”, i.e. up to higher Id-terms.

## The fundamental $\infty$ -groupoid of a type (Lumsdaine, Garner-van den Berg)

Thus each type bears the structure of an  $\infty$ -**groupoid**, with terms, identities between terms, identities between identities, ...



Such structures occur elsewhere in Mathematics, e.g. in Grothendieck's **homotopy hypothesis**.

## Homotopy Levels (Voevodsky)

The universe of types is naturally **stratified** by the level at which the fundamental  $\infty$ -groupoid becomes trivial (if it ever does).

$A$  is **contractible**:  $\sum_{x:A} \prod_{y:X} \text{Id}_A(x, y)$ ,  $A$  is essentially a point.

$A$  is a **proposition**:  $\prod_{x,y:A} \text{Contr}(\text{Id}_A(x, y))$ , identity is contractible.

$A$  is a **set**:  $\prod_{x,y:A} \text{Prop}(\text{Id}_A(x, y))$ , identity is a proposition.

$A$  is a **1-type**:  $\prod_{x,y:A} \text{Set}(\text{Id}_A(x, y))$ , identity is a set.

$A$  is an  $(n+1)$ -**type**:  $\prod_{x,y:A} n\text{Type}(\text{Id}_A(x, y))$ , identity is an  $n$ -type.

This revises **Propositions-as-Types**: higher types are **structures**, rather than mere **propositions**.

## II. Impredicative Encodings

In impredicative type theories such as **Girard's System F** one can form new types by quantifying  $\prod_X$  over all types  $X$ . This can be used to “encode” some of the other type-forming operations.

For example, the encoding of  $\mathbb{N}$  in System F is

$$\mathbb{N} = \prod_X (X \rightarrow X) \rightarrow (X \rightarrow X).$$

Many other inductive types can be encoded in a similar way.

## What good is impredicativity?

- ▶ Impredicativity allows us to **construct** (rather than postulate) many inductive types within a simpler system with good computational behavior. This provides a **justification** for the rules of inductive types, a **computational reduction** of the associated terms, and even a proof of formal **consistency**.
- ▶ Impredicative encodings of inductive types were used in the original **Calculus of Constructions** of Coquand and Huet, and are still present in the **Coq proof assistant**.
- ▶ Impredicative encoding of **higher** inductive types in HoTT could potentially provide the same benefits.
- ▶ A **drawback** of the encodings of inductive types in System F and CoC is that they do not yield the usual elimination rules.
- ▶ In HoTT we can **sharpen the encodings** and construct even higher inductive types that do satisfy the usual rules.

## Impredicative encoding in HoTT

For impredicative encodings in HoTT, we use the  $\prod$ -operation over a universe  $\mathbb{U}$  of (small) types that is “impredicative” in the sense that it satisfies the following rule:

$$\frac{A \text{ Type} \quad x : A \vdash B(x) : \mathbb{U}}{\prod_{x:A} B(x) : \mathbb{U}}$$

This is to be compared with the usual rule, which has the form:

$$\frac{A : \mathbb{U} \quad x : A \vdash B(x) : \mathbb{U}}{\prod_{x:A} B(x) : \mathbb{U}}$$

Thus  $\mathbb{U}$  is assumed to be closed under “large” products, in addition to the usual “small” type formers  $\sum$  and  $\text{Id}$ .



## Impredicative encoding of $A + B$

Consider the System F encoding of the sum  $A + B$  of any two types  $A$  and  $B$ ,

$$A + B = \prod_X (A \rightarrow X) \rightarrow ((B \rightarrow X) \rightarrow X).$$

The join of two **propositions**  $A$  and  $B$  does indeed satisfy

$$A \vee B = \prod_{X:\text{Prop}} (A \rightarrow X) \rightarrow ((B \rightarrow X) \rightarrow X),$$

where  $\text{Prop} = \sum_{X:\mathbb{U}} \text{Prop}(X)$ .

NB:

$$\prod_{X:\sum_{X:\mathbb{U}} \text{Prop}(X)} (\dots) \simeq \prod_{X:\mathbb{U}} \text{Prop}(X) \rightarrow (\dots)$$

## Impredicative encoding of $A + B$

But if  $A$  and  $B$  are **sets**, the type:

$$A + B \stackrel{?}{=} \prod_{X:\text{Set}} (A \rightarrow X) \rightarrow ((B \rightarrow X) \rightarrow X),$$

where  $\text{Set} = \sum_{X:\mathbb{U}} \text{Set}(X)$ , has only a **weak elimination property**. It fails the so-called  $\eta$ -rule that makes the recursor unique.

This means we do not get the usual **dependent elimination rule**, or **induction principle**, for this type. (In HoTT, dependent elimination is equivalent to simple elimination +  $\eta$  by a result of Awodey-Gambino-Sojakova 2016.)

## Impredicative encoding of $A$

We can **sharpen up** the encoding using Id-types as follows. Let  $A$  be a set. Then there is an embedding-retraction pair:

$$\begin{array}{ccc} A & \xrightarrow{e} & \prod_{X:\text{Set}} (A \rightarrow X) \rightarrow X \\ & \searrow \scriptstyle = & \downarrow \scriptstyle r \\ & & A \end{array}$$

A term  $\alpha : \prod_{X:\text{Set}} (A \rightarrow X) \rightarrow X$  is a **family of maps**,

$$\alpha_X : X^A \longrightarrow X, \quad X : \text{Set}.$$

We can cut this type down to the image of  $e$  by requiring that these maps be **natural in  $X$** .

## Impredicative encoding of $A$

**Naturality** means that for all sets  $Y$  and all maps  $f : X \rightarrow Y$ , the following commutes.

$$\begin{array}{ccc} X^A & \xrightarrow{\alpha_X} & X \\ f^A \downarrow & & \downarrow f \\ Y^A & \xrightarrow{\alpha_Y} & Y \end{array}$$

The **sharper encoding** we seek is therefore:

$$A \simeq \sum_{\alpha:A^*} \prod_{X,Y:\text{Set}} \prod_{f:X \rightarrow Y} \text{Id}(\alpha_Y \circ f^A, f \circ \alpha_X),$$

where

$$A^* = \prod_{X:\text{Set}} (A \rightarrow X) \rightarrow X.$$

# Impredicative encoding of $A$

## Theorem (Main Lemma)

*For any set  $A$  in HoTT with an impredicative universe, there is a natural equivalence,*

$$A \simeq \sum_{\alpha:A^*} \prod_{X,Y:\text{Set}} \prod_{f:X \rightarrow Y} \text{Id}(\alpha_Y \circ f^A, f \circ \alpha_X),$$

where

$$A^* = \prod_{X:\text{Set}} (A \rightarrow X) \rightarrow X.$$

## Impredicative encoding of $A + B$

Returning to  $A + B$ , by the main lemma we have the following comparison with the System F encoding:

$$\begin{aligned} A + B \subseteq (A + B)^* &= \prod_{X:\text{Set}} ((A + B) \rightarrow X) \rightarrow X \\ &\simeq \prod_{X:\text{Set}} ((A \rightarrow X) \times (B \rightarrow X)) \rightarrow X \\ &\simeq \prod_{X:\text{Set}} (A \rightarrow X) \rightarrow ((B \rightarrow X) \rightarrow X). \end{aligned}$$

We can therefore **sharpen up** the encoding by naturality just as before, since  $(A \rightarrow X) \times (B \rightarrow X)$  is functorial in  $X$ .

## Impredicative encoding of $\mathbb{N}$

The encoding of  $\mathbb{N}$  in System F was

$$\mathbb{N} = \prod_X (X \rightarrow X) \rightarrow (X \rightarrow X).$$

Again, we can sharpen this encoding using Id-types as follows.

### Theorem

*For any functor  $T : \text{Set} \rightarrow \text{Set}$ , the category of  $T$ -algebras has an initial object,*

$$i : T(I) \rightarrow I,$$

*where  $I$  is the limit of the forgetful functor  $U : \text{TAlg} \rightarrow \text{Set}$ ,*

$$I = \varprojlim_{A:\text{TAlg}} UA \rightarrow \prod_{A:\text{TAlg}} UA \rightrightarrows \prod_{\substack{A,B:\text{TAlg} \\ h:A \rightarrow B}} UB.$$

## Impredicative encoding of $\mathbb{N}$

The type  $\text{TAlg}$  occurring in the index is the type of  $T$ -algebras,

$$\text{TAlg} = \sum_{X:\text{Set}} TX \rightarrow X.$$

So for the **initial algebra**  $i : TI \rightarrow I$  we have,

$$\begin{aligned} I &= \underset{A:\text{TAlg}}{\text{lim}} UA \subseteq \prod_{A:\text{TAlg}} UA \\ &\simeq \prod_{A:\sum_{X:\text{Set}} TX \rightarrow X} UA \\ &\simeq \prod_{(X,t):\sum_{X:\text{Set}} TX \rightarrow X} X \\ &\simeq \prod_{X:\text{Set}} \prod_{t:TX \rightarrow X} X \\ &\simeq \prod_{X:\text{Set}} (TX \rightarrow X) \rightarrow X. \end{aligned}$$

The equalizer  $I$  is then **definable** using a suitable Id-type.



## Impredicative encoding of $\mathbb{N}$

Now apply the foregoing to get  $\mathbb{N}$  as the **initial algebra** of the endofunctor  $TX = X + 1$ ,

$$\begin{aligned}\mathbb{N} &= \varprojlim_{A:\mathbf{TAlg}} UA \subseteq \prod_{X:\mathbf{Set}} ((X + 1) \rightarrow X) \rightarrow X \\ &\simeq \prod_{X:\mathbf{Set}} (X \rightarrow X) \rightarrow (X \rightarrow X).\end{aligned}$$

Again our sharper encoding is a **definable subtype** of the System F encoding. As before, the **induction principle** follows from recursion together with the uniqueness of the recursor.

## Impredicative encoding of inductive types

Many other Set-level encodings can be done in this way: quotients, propositional and set truncations, coproducts of families, etc.

For example, the **propositional truncation** of any type  $A$  is simply

$$\|A\| = \prod_{X:\text{Prop}} (A \rightarrow X) \rightarrow X.$$

The **set truncation** starts with

$$\|A\|_0 \subseteq \prod_{X:\text{Set}} (A \rightarrow X) \rightarrow X,$$

and then adds a naturality condition to sharpen it up, as before.

## Impredicative encoding of higher inductive types: $S^1$

Finally, one can do something similar for some other **higher inductive types**. For example, Shulman proposed the “System F-style” encoding,

$$S^1 = \prod_X \prod_{x:X} (x = x) \rightarrow X.$$

This has the same problem as the System F encoding of  $\mathbb{N}$ : no uniqueness for the eliminator, and so no induction principle.

But we can remedy this in the same way as before, by restricting the  $\prod_X$  to 1-types, and then adding higher coherence conditions, reflecting the fact that  $S^1$  is a 1-type rather than a set.

## Impredicative encoding of higher inductive types: $S^1$

Indeed, by the **universal property of the circle** we have

$$(S^1 \rightarrow X) \simeq \sum_{x:X} (x = x).$$

By the main lemma, we therefore get

$$\begin{aligned} S^1 &\subseteq \prod_{X:\text{Type}_1} (S^1 \rightarrow X) \rightarrow X \\ &\simeq \prod_{X:\text{Type}_1} \left( \sum_{x:X} (x = x) \right) \rightarrow X \\ &\simeq \prod_{X:\text{Type}_1} \prod_{x:X} (x = x) \rightarrow X. \end{aligned}$$

We then sharpen up the encoding as before, but now adding **higher coherence** conditions expressed using higher Id-types.

The same method encodes some other n-types, like groupoid quotients and n-truncations  $\|X\|_n$ .

That was fun ...

## That was fun ... but is it safe?

- ▶ Is it consistent to have a universe  $\mathbb{U}$  that is both **impredicative** and **univalent** ?
- ▶ Yes! In a topos, the **subobject classifier**  $\Omega$  is a universe of propositions that is both impredicative and univalent.
- ▶ What about a **proof-relevant** universe  $\mathbb{U}$ , i.e. not a poset?
- ▶ Models of System F and CoC can be made using **realizability**: the category  $\mathcal{A}sm$  of assemblies has an internal category  $\mathcal{M}$  of modest sets that is **complete** and is not a partial order. (Hyland)
- ▶ A proof-relevant, impredicative universe that is also **univalent** therefore lives inside the **groupoid model** of type theory built inside of  $\mathcal{A}sm$ , with the groupoid  $\mathbb{M}$  of modest sets as a universe (Hofmann-Streicher 1995 + Awodey-Bauer 2013).
- ▶ But the universe  $\mathbb{M}$  in  $\mathbf{Gpd}(\mathcal{A}sm)$  consists only of **sets**, just as the  $\Omega$  in a topos consists only of **propositions**.

### III. A Cubical Realizability Model

We can now extend the pattern in the foregoing.

- ▶ In order to get a model with arbitrary  $n$ -types, we generalize from groupoids to  $\infty$ -**groupoids** inside a realizability model  $\mathcal{A}sm$  of impredicativity.
- ▶ These are  $\mathcal{A}sm$ -valued **Kan complexes**, i.e. certain presheaves with values in  $\mathcal{A}sm$ .
- ▶ Our presheaves are **cubical** rather than simplicial, since this makes for better Kan complexes in the constructive setting (Coquand).
- ▶ The **realizability  $\infty$ -topos**  $RT_\infty$  is a QMC based on cubical presheaves in  $\mathcal{A}sm$ .
- ▶ Our **present goal** is to show that the complete internal subcategory  $\mathbb{M}$  consisting of the **modest Kan complexes** provides a univalent universe.

## Some details: Graphs, Setoids, Groupoids, etc.

- ▶ The “realizability setoid model”  $RT_0$  consists of certain graphs in the category of assemblies  $\mathcal{A}sm$ ,

$$RT_0 \hookrightarrow \mathcal{A}sm(\cdot \stackrel{\Leftarrow}{\Leftarrow} \cdot)$$

namely, those that are “setoids”, i.e. reflexive, symmetric, and transitive. This is essentially the realizability topos  $RT$ .

- ▶ The “realizability groupoid model”  $RT_1$  consists of certain 2-graphs,

$$RT_1 \hookrightarrow \mathcal{A}sm(\cdot \stackrel{\Leftarrow}{\Leftarrow} \cdot \stackrel{\Leftarrow}{\Leftarrow} \cdot)$$

namely, those that are “groupoids”, i.e. associative and unital.



## Some details: Graphs, Setoids, Groupoids, etc.

- ▶ The “realizability  $\infty$ -groupoid model” consists of certain cubical objects,

$$RT_{\infty} \hookrightarrow \mathcal{A}sm \left( \cdot \begin{array}{c} \leftarrow \\ \leftarrow \\ \leftarrow \end{array} \cdot \begin{array}{c} \leftarrow \\ \leftarrow \\ \leftarrow \\ \leftarrow \\ \leftarrow \end{array} \cdot \begin{array}{c} \leftarrow \\ \leftarrow \\ \leftarrow \\ \leftarrow \\ \leftarrow \\ \leftarrow \\ \leftarrow \end{array} \cdot \dots \right)$$

namely those that are normal, uniform, cubical, Kan complexes.

Here:

- ▶ *cubical* means a presheaf on the cube category,
- ▶ *Kan* means fillers for all open boxes,
- ▶ *uniform* means the fillers are given as structure,
- ▶ *normal* means degenerate fillers for degenerate boxes.

## Some more details

We use the *cartesian* cube category  $\mathbb{C}$ , the dual  $\mathbb{C} = \mathbb{B}^{\text{op}}$  of the category  $\mathbb{B}$  of finite, strictly bipointed sets.

The cubical presheaves valued in  $\mathcal{A}sm$  are thus discrete fibrations on the internal category  $\mathbb{B}$ , forming the LCCC  $\mathcal{A}sm^{\mathbb{B}}$ .

We seek to avoid the general small object argument because  $\mathcal{A}sm$  lacks infinite colimits. So the methods of Quillen, Garner, Gambino-Sattler do not directly apply.

But if we restrict to the subcategory of Kan objects, we can use the *pathspace factorization* in order to factor all maps “algebraically”:

$$\begin{array}{ccc} A & \xrightarrow{f} & B \\ & \searrow L(f) & \nearrow R(f) \\ & P(f) & \end{array}$$

## A theorem (Awodey-Frey-Hofstra)

Let  $\mathcal{E}$  be a LCCC with a natural numbers object.

Form the cubical presheaves  $\mathcal{E}^{\mathbb{B}}$  and the subcategory  $\mathcal{K} \hookrightarrow \mathcal{E}^{\mathbb{B}}$  of normal, uniform Kan objects. Then:

- ▶  $\mathcal{K}$  has a (cloven) WFS in which the left maps are the *strong deformation retracts* and the right maps are those with the *homotopy lifting property*.
- ▶ for any object  $X$  in  $\mathcal{K}$  and any R-map  $A \longrightarrow X$ , the canonical factorization  $A \rightarrow A^{\mathbb{I}} \rightarrow A \times_X A$  of the diagonal is a stable L-R factorization.
- ▶ the Frobenius condition holds for pullbacks of L-maps along R-maps.

Thus we have a “realizability model of HoTT” in the Kan objects of  $\mathcal{Asm}^{\mathbb{B}}$ , in which the identity types are the path spaces,

$$\text{Id}_A = A^{\mathbb{I}}.$$

# The impredicative univalent universe (WIP!)

The **modest Kan objects** in  $\mathcal{A}sm^{\mathbb{B}}$  form a universe

$$p : \tilde{M} \rightarrow M$$

such that:

- ▶  $p : \tilde{M} \rightarrow M$  is closed under  $1, \Sigma, \Pi$ , i.e. the associated **polynomial endofunctor**  $P(X) = \sum_{A:M} X^A$  is a **monad** and an **algebra**.
- ▶  $p : \tilde{M} \rightarrow M$  is closed under  $\text{Id}_A = A^{\mathbb{I}}$ , since this is a “shift”. Moreover, this **pathobject** functor has a **right** adjoint.
- ▶  $p : \tilde{M} \rightarrow M$  is **internally complete**, and so admits impredicative encodings of (higher) inductive types.
- ▶ To do: We know that  $p : \tilde{M} \rightarrow M$  is a **fibration**, but we need to show that  $M$  is **fibrant**! Sattler’s equivalence extension theorem implies both this and **univalence**, but it uses connections; we want to do it without them!