Using Agda to Explore Path-Oriented Models of Type Theory

Andrew Pitts

joint work with Ian Orton

UNIVERSITY OF CAMBRIDGE

Computer Laboratory

Outline

- The mathematical problem
  - find new models of Homotopy Type Theory
- Why use an interactive theorem prover for this?
  - and why Agda particularly?

HoTT 101

Martin-Löf Type Theory

is formulated in terms of judgements

"A is a type"	$(A\ \mathtt{type})$
"A and B are equal types"	$(A = B\ \mathtt{type})$
"a is a thing of type A"	$(a : A)$
"a and b are equal things of type A"	$(a = b : A)$

not in first-order logic – just an inductive definition using hypothetical judgements. . .

Martin-Löf Type Theory

is formulated in terms of hypothetical judgements, e.g.

$$x : A, y : B(x) \vdash a(x, y) : C(x, y)$$

$$x : A, y : B(x) \vdash a(x, y) = b(x, y) : C(x, y)$$

Martin-Löf Type Theory

is formulated in terms of hypothetical judgements, e.g.

$$x : A, y : B(x) \vdash a(x,y) : C(x,y)$$

$$x : A, y : B(x) \vdash a(x,y) = b(x,y) : C(x,y)$$

involving dependent types.

Expressive power comes via three higher-order aspects. . .

Higher-order features

function types

$$A \to B$$

$$(A \to B) \to C$$

$$((A \to B) \to C) \to D$$

$$\vdots$$

CS	Logic	Math
✔	✔	✖

Higher-order features

universes – types whose elements are (codes for) types

```
Prop : Set : Type : TYPE ···
```

CS	Logic	Math
✖	✔	✖

Higher-order features

identity types – types of proofs of equality

formation: $x : A, y : A \vdash \mathtt{Id}_A(x, y)\ \mathtt{type}$
introduction: $x : A \vdash \mathtt{refl} : \mathtt{Id}_A(x, x)$
elimination & computation: [details omitted – DIY!]

Higher-order features

(intensional) identity types

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

$$\mathtt{Id}_{\mathtt{Id}_A(x,y)}(p, q)$$

$$\mathtt{Id}_{\mathtt{Id}_{\mathtt{Id}_A(x,y)}(p,q)}(P, Q)$$

$$\mathtt{Id}_{\mathtt{Id}_{\mathtt{Id}_{\mathtt{Id}_A(x,y)}(p,q)}(P,Q)}(\mathcal{P}, \mathcal{Q})$$

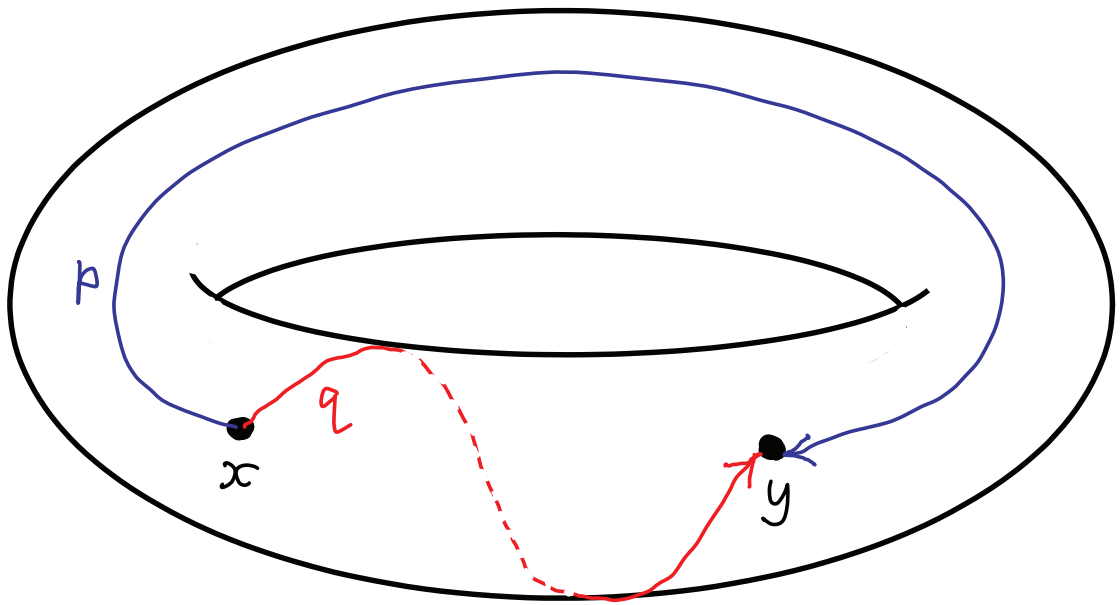
$$\vdots$$

CS	Logic	Math
✖	✖	✔

Equality-as-path

[Awodey-Warren, Voevodsky,. . .]

elements of $\mathtt{Id}_A(x, y)$ are analogous to
paths from point x to point y in a space A



Equality-as-path

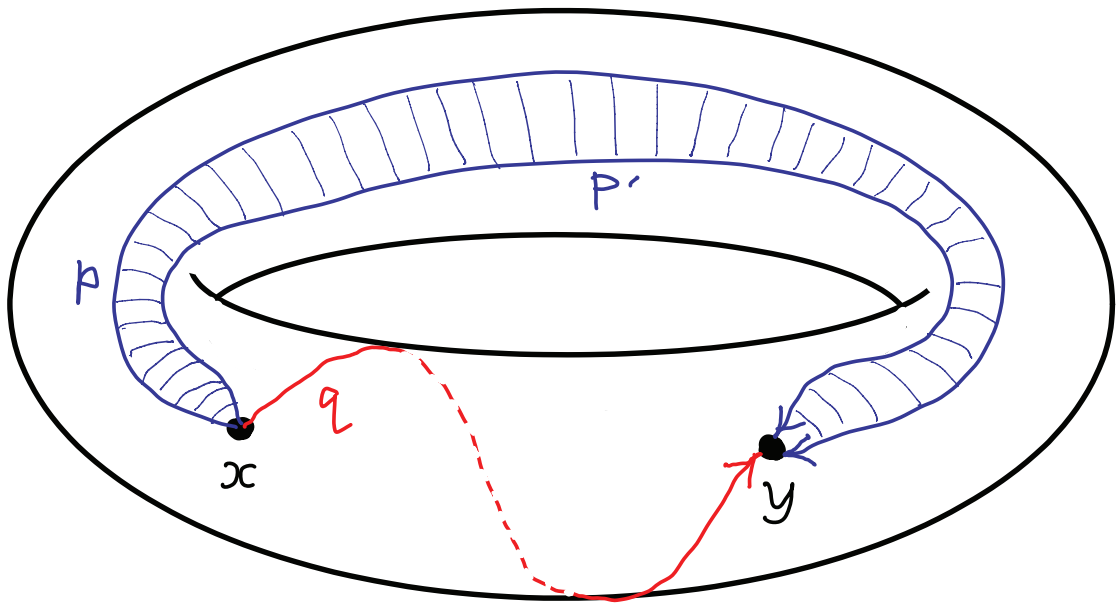
[Awodey-Warren, Voevodsky,. . .]

elements of $\mathtt{Id}_A(x, y)$ are analogous to
paths from point x to point y in a space A

elements of $\mathtt{Id}_{\mathtt{Id}_A(x,y)}(p, q)$ as homotopies between paths p and q,

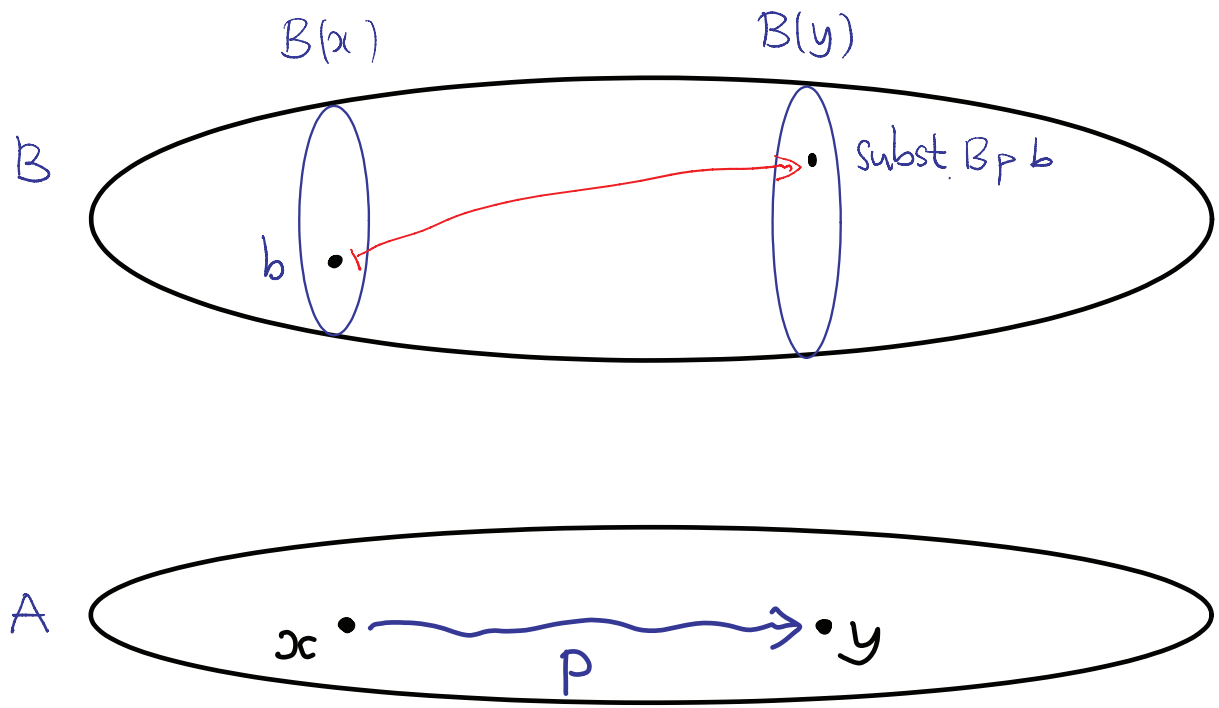
etc.

$p \to p'$
$p \not\to q$

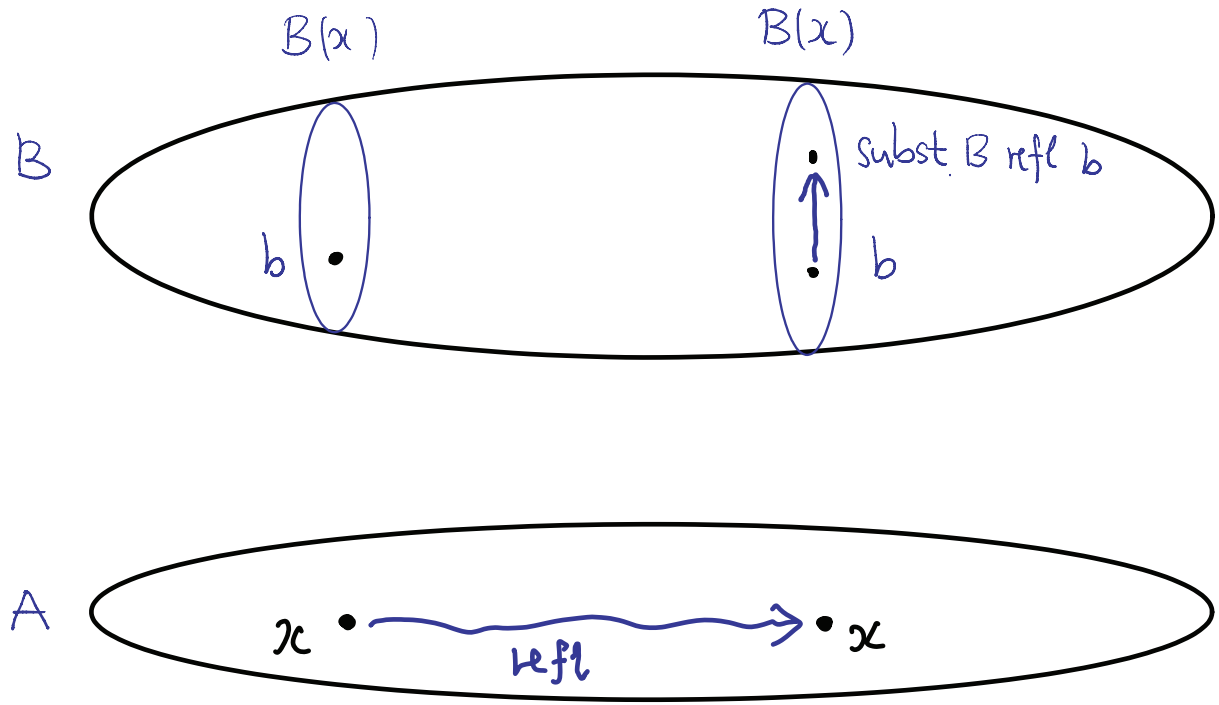


Type Theory	Homotopy Theory
$p : \mathtt{Id}_A(x, y)$	path p from x to y in a space A
$\mathtt{refl} : \mathtt{Id}_A(x, x)$	degenerate path constantly at x

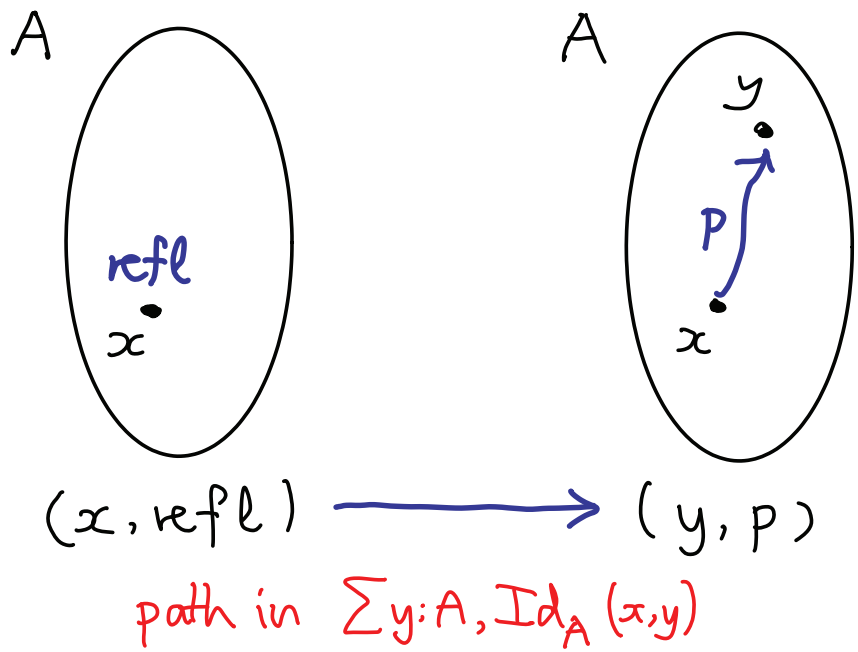
Type Theory	Homotopy Theory
$p : \mathrm{Id}_A(x, y)$	path p from x to y in a space A
$\mathtt{refl} : \mathrm{Id}_A(x, x)$	degenerate path constantly at x
$\mathtt{subst}$	transport along paths in fibred space



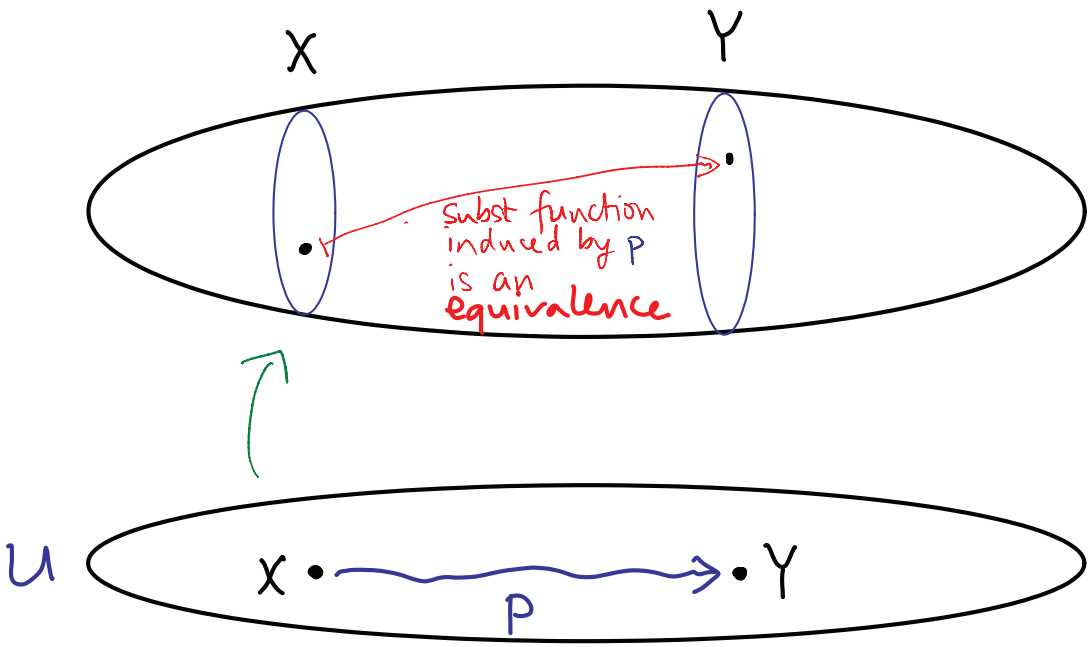
Type Theory	Homotopy Theory
$p : \mathtt{Id}_A(x, y)$	path p from x to y in a space A
$\mathtt{refl} : \mathtt{Id}_A(x, x)$	degenerate path constantly at x
$\mathtt{subst}$	transport along paths in fibred space



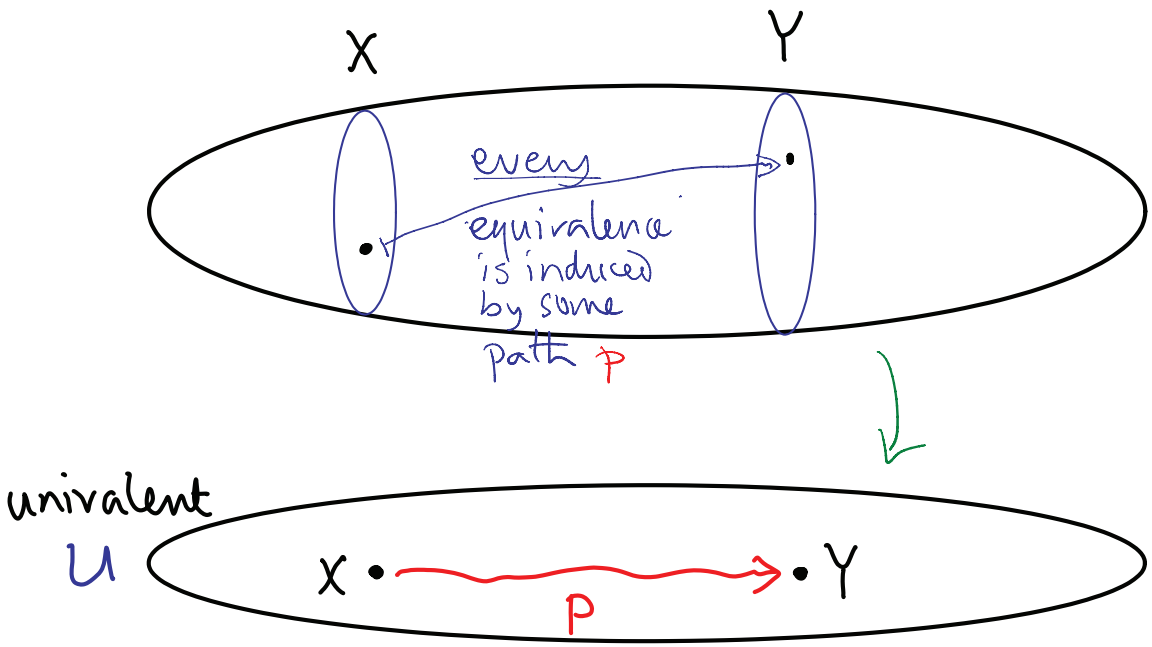
Type Theory	Homotopy Theory
$p : \mathrm{Id}_A(x, y)$	path p from x to y in a space A
$\mathtt{refl} : \mathrm{Id}_A(x, x)$	degenerate path constantly at x
$\mathtt{subst}$	transport along paths in fibred space
$\exists!$	path-contractible spaces [Voevodsky]



Type Theory	Homotopy Theory
$p : \mathtt{Id}_A(x, y)$	path p from x to y in a space A
$\mathtt{refl} : \mathtt{Id}_A(x, x)$	degenerate path constantly at x
$\mathtt{subst}$	transport along paths in fibred space
$\exists!$	path-contractible spaces [Voevodsky]
univalent universes [Voevodsky]	only properties invariant under homotopy equivalence



Type Theory	Homotopy Theory
$p : \mathtt{Id}_A(x, y)$	path p from x to y in a space A
$\mathtt{refl} : \mathtt{Id}_A(x, x)$	degenerate path constantly at x
$\mathtt{subst}$	transport along paths in fibred space
$\exists!$	path-contractible spaces [Voevodsky]
univalent universes [Voevodsky]	only properties invariant under homotopy equivalence



Type Theory	Homotopy Theory
$p : \mathtt{Id}_A(x, y)$	path p from x to y in a space A
$\mathtt{refl} : \mathtt{Id}_A(x, x)$	degenerate path constantly at x
$\mathtt{subst}$	transport along paths in fibred space
$\exists!$	path-contractible spaces [Voevodsky]
univalent universes [Voevodsky]	only properties invariant under homotopy equivalence

Existing models of univalent type theory:

– Kan simplicial sets in classical set theory [Voevodsky]

– uniform-Kan cubical sets in constructive set theory [Coquand, *et al*]

We need more (and simpler) ones!

Category Theory

Use categorical algebra to organize what is needed for a model.

- ✖ categories of (tame) topological spaces

Category Theory

Use categorical algebra to organize what is needed for a model.

- ✖ categories of (tame) topological spaces
- ✔ Grothendieck's toposes = categories of set-valued sheaves on generalised notion of space

Category Theory

Use categorical logic to organize what is needed for a model.

- ✖ categories of (tame) topological spaces
- ✔ Grothendieck's toposes = categories of set-valued sheaves on generalised notion of space
- ? elementary toposes = models of intuitionistic HOL

Can use the language of (intuitionistic, extensional) higher-order logic / type theory to explore relevant constructions in a topos

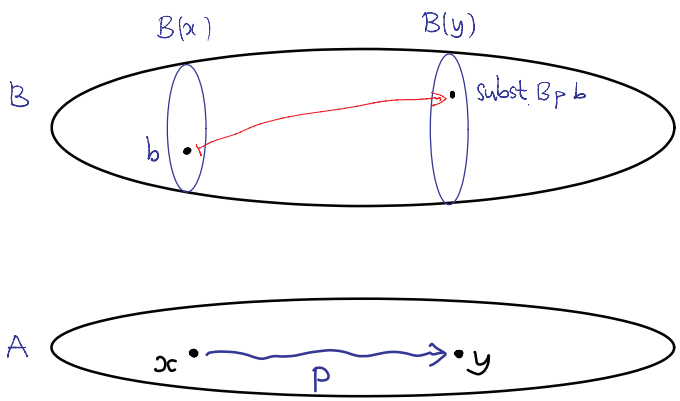
– makes things much easier to understand (for non arrow-heads).

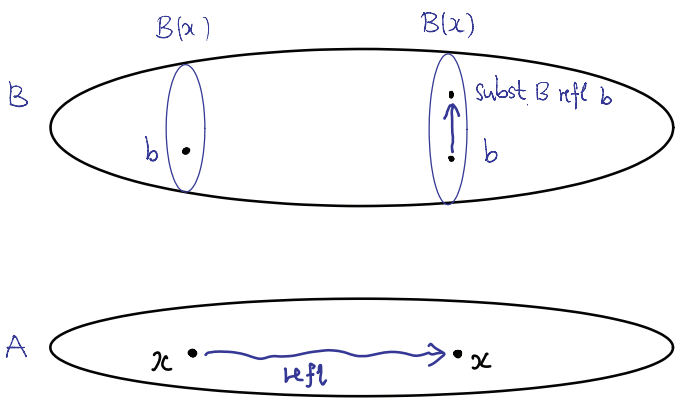
An experiment

Work in the internal language of a topos equipped with an interval $\mathtt{I}$ (with end-points $\mathtt{0}, \mathtt{1} : \mathtt{I}$) and consider associated path types:

formation: $x, y : A \vdash \mathtt{Path}_A(x, y)\ \mathtt{type}$
introduction: $f : \mathtt{I} \to A \vdash \mathtt{path}(f) : \mathtt{Path}_A(f\,\mathtt{0}, f\,\mathtt{1})$

$+$ simplest possible `subst` structure, "Dold fibrations"





An experiment

Work in the internal language of a topos equipped with an interval $\mathtt{I}$ (with end-points $\mathtt{0}, \mathtt{1} : \mathtt{I}$) and consider associated path types:

formation: $x, y : A \vdash \mathtt{Path}_A(x, y)\ \mathtt{type}$
introduction: $f : \mathtt{I} \to A \vdash \mathtt{path}(f) : \mathtt{Path}_A(f\,\mathtt{0}, f\,\mathtt{1})$

$+$ simplest possible `subst` structure, "Dold fibrations"

What properties of $\mathtt{I}$ allow one to get a model of MLTT $+$ univalence this way? (with identity types given by `Path`)

We have been using Agda to help us explore this question.

Agda

http://wiki.portal.chalmers.se/agda

At its core:

inductive definitions of indexed families of data types involving dependently typed functions.

User declares what are the types of data constructors and then Agda assists the user to define well-typed functions on the data using dependently-typed patterns [Coquand], by gradually replacing meta-variables ("holes").

Functional programming = theorem proving:
propositions are (some) types, proofs are elements of types.

Agda

http://wiki.portal.chalmers.se/agda

At its core:

inductive definitions of indexed families of data types involving dependently typed functions.

User declares what are the types of data constructors and then Agda assists the user to define well-typed functions on the data using dependently-typed patterns [Coquand], by gradually replacing meta-variables ("holes").

Functional programming = theorem proving:
propositions are (some) types, proofs are elements of types.

Simple access to these is the deal clincher for me (background: semantics of programming languages).

Agda

- ✔ Just one language (very coherent, with a beautiful, flexible Haskell-style concrete syntax)
- ✔ No separate tactic language (other than elisp!)

Agda

- ✔ Just one language (very coherent, with a beautiful, flexible Haskell-style concrete syntax)
- ✔ No separate tactic language (other than elisp!)
- ✖ No tactics (yet)

Agda

- ✔ Just one language (very coherent, with a beautiful, flexible Haskell-style concrete syntax)
- ✔ No separate tactic language (other than elisp!)
- ✖ No tactics (yet)
- ✔ Test-bed for new ideas.
- ✖ Test-bed for new ideas.

Agda

- ✔ Just one language (very coherent, with a beautiful, flexible Haskell-style concrete syntax)
- ✔ No separate tactic language (other than elisp!)
- ✖ No tactics (yet)
- ✔ Test-bed for new ideas.
- ✖ Test-bed for new ideas.
- ? Can it cope with big proofs?

Agda

- ✔ Just one language (very coherent, with a beautiful, flexible Haskell-style concrete syntax)
- ✔ No separate tactic language (other than elisp!)
- ✖ No tactics (yet)
- ✔ Test-bed for new ideas.
- ✖ Test-bed for new ideas.
- ? Can it cope with big proofs?
- ✔ Access to unsafe features – great for experimentation rather than trusted verification.

An experiment

Work in the internal language of a topos

To simulate enough of this, we use unsafe features of Agda

- `postulate`s & `TrustMe`
- local use of `--type-in-type`
- `REWRITE` & `POLARITY` pragmas

to add to MLTT

either impredicative universe of h-propositions [Escardo]

or Hofmann's predicative quotient types

Adding quotient types to Agda

```
postulate
  _/_ : (A : Set)(R : A → A → Set) → Set  -- formation

{-# POLARITY _/_ ++ * #-}                 -- make _/R strictly +ve

module _ (A : Set)(R : A → A → Set) where
  postulate
    [_] : A → A / R                       -- introduction

    by : ∀{x y} → R x y → [ x ] ≡ [ y ]   -- introduction equality

    qelim :                               -- elimination
      (B : A / R → Set)
      (f : (x : A) → B [ x ])
      (e : (x y : A)(r : R x y) → subst B (by r) (f x) ≡ f y)
      → ----------------------------------------------------
      (y : A / R) → B y

    qcomp :                               -- computation
      (B : A / R → Set)
      (f : (x : A) → B [ x ])
      (e : (x y : A)(r : R x y) → subst B (by r) (f x) ≡ f y)
      → ----------------------------------------------------
      (x : A) → qelim B f e [ x ] ≡ f x

  {-# REWRITE qcomp #-}                   -- make qcomp definitional
```

Adding quotient types to Agda

```
postulate
  _/_ : (A : Set)(R : A → A → Set) → Set   -- formation
```

No need to assume R is an equivalence relation.

We work modulo axiom K (uniqueness of identity proofs), so do not need to postulate more than qcomp.

As well as quotients with the expected universal property, this gives:

- propositional truncation, mere existence and disjunction
- function extensionality

BUT programming with qelim is a pain – there has to be a better way.

```
  qcomp :                                          -- computation
    (B : A / R → Set)
    (f : (x : A) → B [ x ])
    (e : (x y : A)(r : R x y) → subst B (by r) (f x) ≡ f y)
    → ------------------------------------------------
    (x : A) → qelim B f e [ x ] ≡ f x

{-# REWRITE qcomp #-}                              -- make qcomp definitional
```

An experiment

Work in the internal language of a topos equipped with an interval $\mathtt{I}$ (with end-points $\mathtt{0}, \mathtt{1} : \mathtt{I}$) and consider associated path types:

formation: $x, y : A \vdash \mathtt{Path}_A(x, y)\ \mathtt{type}$
introduction: $f : \mathtt{I} \to A \vdash \mathtt{path}(f) : \mathtt{Path}_A(f\,\mathtt{0}, f\,\mathtt{1})$

$+$ simplest possible `subst` structure, "Dold fibrations"

What properties of $\mathtt{I}$ allow one to get a model of MLTT $+$ univalence this way? (with identity types given by `Path`)

An interval theory

non-trivial $\neg(\mathtt{0} = \mathtt{1})$

connected $(\forall i : \mathtt{I},\ P(i) \vee \neg P(i)) \to \forall i, j : \mathtt{I},\ P(i) \to P(j)$

total order $\forall i, j : \mathtt{I},\ i \leq j \ \vee \ j \leq i$

with monus

$\forall i, j : \mathtt{I},\ (j \leq i \to i \dot{-} j \leq i) \wedge (i \dot{-} i = \mathtt{0}) \wedge (i \dot{-} (i \dot{-} j) = j)$

Theorem (in Agda) For any such $\mathtt{I}$ in a topos, Dold fibrations give a model of MLTT with $\mathbf{\Sigma}$, $\mathbf{\Pi}$, $\mathtt{Id}$, $\mathbf{0}$, $\mathbf{1}$ $+$, $\boldsymbol{W}$ types, ...

An interval theory

non-trivial $\neg(0 = 1)$

connected $(\forall i : \mathtt{I},\ P(i) \vee \neg P(i)) \to \forall i, j : \mathtt{I},\ P(i) \to P(j)$

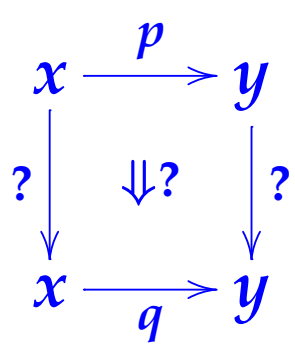
total order $\forall i, j : \mathtt{I},\ i \leq j \ \vee \ j \leq i$

with monus

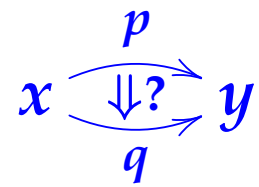
$\forall i, j : \mathtt{I},\ (j \leq i \to i \dot{-} j \leq i) \wedge (i \dot{-} i = 0) \wedge (i \dot{-} (i \dot{-} j) = j)$

Theorem (in Agda) For any such $\mathtt{I}$ in a topos, Dold fibrations give a model of MLTT with $\boldsymbol{\Sigma}$, $\boldsymbol{\Pi}$, $\mathtt{Id}$, $\mathbf{0}$, $\mathbf{1}$ $+$, $\boldsymbol{W}$ types, . . .

Among other things, Agda helped us to calculate correctly with (higher) paths. E.g. stops one confusing



with



```
IdActId :
    ∀{ℓ ℓ' Γ} →
    (A : Γ → Set ℓ')
    {{_ : isFib A}}
    (a b : (x : Γ) → A x)
    (x : Γ)
    (q : Id A a b x)
    → ----------------------------
    q ~ IdAct A a b x x (~refl x) q
IdActId A {{α}} a b x q =
  let f = q at_ in
  path:
    q
  ~[ •refl q ]
    q • ~refl (a x)
  ~[ ~cong (q •_) (~symm (~invl (~substrefl α (a x)))) ]
    q • ~symm (~substrefl α (a x)) • ~substrefl α (a x)
  ~[ ~cong (λ p → q • ~symm (~substrefl α (a x)) • p) (refl• (~substrefl α (a x))) ]
    q • ~symm (~substrefl α (a x)) • ~refl (~subst α (~refl x) (a x)) • ~substrefl α (a x)
  ~[ ⟨ j ⟩((q from j) • ~symm (~substrefl α (f j)) • ⟨ i ⟩(~subst α (~refl x) (f (cvx 0 i j))) • ~substrefl α (a x)) ]
    ~refl (b x) • ~symm (~substrefl α (b x)) • ⟨ i ⟩(~subst α (~refl x) (f i)) • ~substrefl α (a x)
  ~[ ~symm (refl• (~symm (~substrefl α (b x)) • ⟨ i ⟩(~subst α (~refl x) (f i)) • ~substrefl α (a x))) ]
    ~symm (~substrefl α (b x)) • ⟨ i ⟩(~subst α (~refl x) (f i)) • ~substrefl α (a x)
  ~[ ~cong (λ p → ~symm p • ⟨ i ⟩(~subst α (~refl x) (f i)) • ~substrefl α (a x)) (I~*0id A {{α}} b x) ]
    ~symm (I~*0 A b (~refl x)) • ⟨ i ⟩(~subst α (~refl x) (f i)) • ~substrefl α (a x)
  ~[ ~cong (λ p → ~symm (I~*0 A b (~refl x)) • ⟨ i ⟩(~subst α (~refl x) (f i)) • p) (I~*0id A {{α}} a x) ]
    ~symm (I~*0 A b (~refl x)) • ⟨ i ⟩(~subst α (~refl x) (f i)) • (I~*0 A a (~refl x))
  endp
```

An interval theory

non-trivial $\neg(\mathtt{0} = \mathtt{1})$

connected $(\forall i : \mathtt{I},\ P(i) \vee \neg P(i)) \to \forall i, j : \mathtt{I},\ P(i) \to P(j)$

total order $\forall i, j : \mathtt{I},\ i \leq j \ \vee \ j \leq i$

with monus

$\forall i, j : \mathtt{I},\ (j \leq i \to i \dot{-} j \leq i) \wedge (i \dot{-} i = \mathtt{0}) \wedge (i \dot{-} (i \dot{-} j) = j)$

Theorem (in Agda) For any such $\mathtt{I}$ in a topos, Dold fibrations give a model of MLTT with $\mathbf{\Sigma}$, $\mathbf{\Pi}$, $\mathtt{Id}$, $\mathbf{0}$, $\mathbf{1}$ $+$, $\boldsymbol{W}$ types, ...

Theorem (pencil-and-paper) There are examples of such toposes

An interval theory

non-trivial $\neg(\mathtt{0} = \mathtt{1})$

connected $(\forall i : \mathtt{I},\ P(i) \vee \neg P(i)) \to \forall i, j : \mathtt{I},\ P(i) \to P(j)$

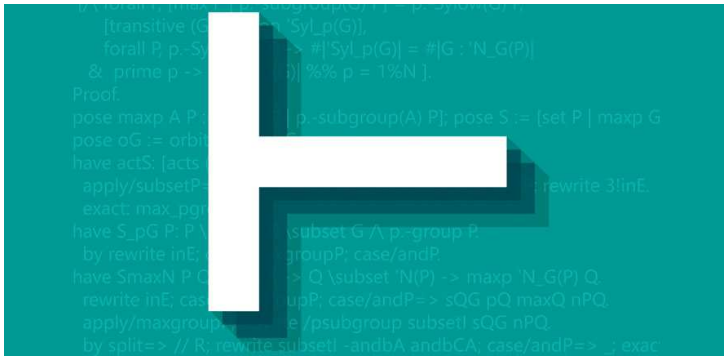
total order $\forall i, j : \mathtt{I},\ i \leq j \ \vee \ j \leq i$

with monus

$\forall i, j : \mathtt{I},\ (j \leq i \to i \dot{-} j \leq i) \wedge (i \dot{-} i = \mathtt{0}) \wedge (i \dot{-} (i \dot{-} j) = j)$

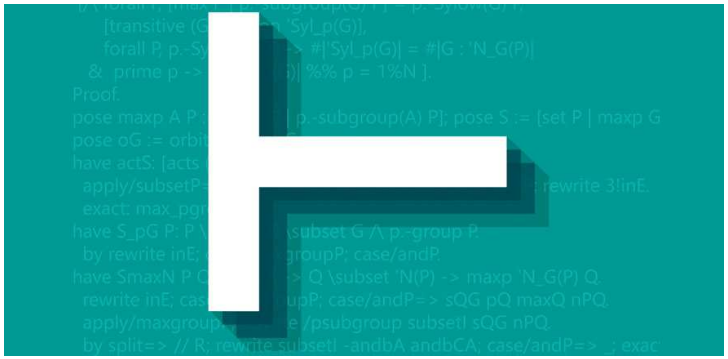
Theorem (in Agda) For any such $\mathtt{I}$ in a topos, Dold fibrations give a model of MLTT with $\mathbf{\Sigma}$, $\mathbf{\Pi}$, $\mathtt{Id}$, $\mathbf{0}$, $\mathbf{1}$ $+$, $\boldsymbol{W}$ types, . . . (univalence?)

Theorem (pencil-and-paper) There are examples of such toposes (but not presheaf toposes )



4. The social exploration and curation of formalised mathematical and scientific knowledge.

Q How do I find out about the (possibly obscure) bits of homotopy theory that I might need to build path-based models of type theory?



4. The social exploration and curation of formalised mathematical and scientific knowledge.

Q How do I find out about the (possibly obscure) bits of homotopy theory that I might need to build path-based models of type theory?

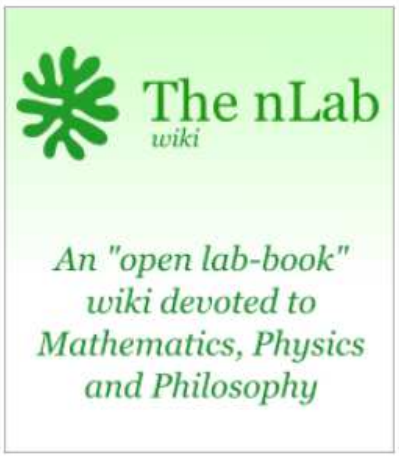
A Search nLab [Urs Schreiber].

ncatlab.org

Schreiber

What is... the nLab

Home Page | All Pages | Latest Revisions | Authors | Feeds | Export | Search



What is... the nLab?

Some thoughts. (See also: Wikipedia on the nLab)

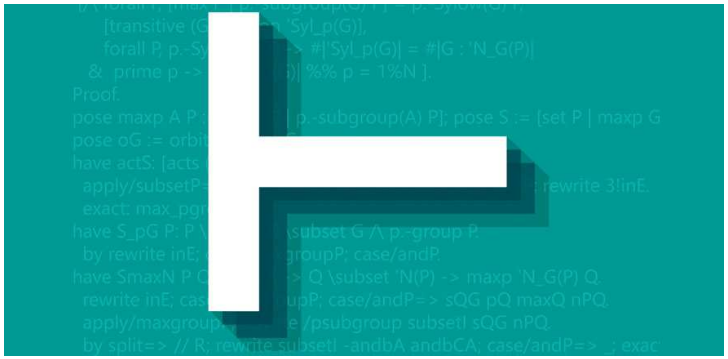
Contents

1. Connect the relevant information.

We all waste too much time with searching for mathematical information that is already out there. As a student, before the dawn of the internet, I wasted days in the library, on chasing references to the secrets of the universe. Now the internet exists, but we still waste time searching randomly. Things have not been connected. The *n*Lab means to connect the dots. The idea is that you stop searching randomly and just follow the links. *Hypertext*. That was the original vision of the web. We need more research-level maths hypertext.

2. Show the big picture.

We are in an age where in theoretical physics we are supposed to work on quantum gravity and unification, needing the very latest of the developments in mathematics. At the same time we still bring up students with old textbooks. This way even the best of them at the end of their study can only grasp a tiny fraction of the big picture, because the knowledge is so scattered in tiny sub-expert communities. This is insane and unnecessary. The *n*Lab means to connect the dots and show the big picture. That's why it's *organized* by higher category theory. This is the structure that helps organize things and bring them together conceptually. (While of course many specific entries need not be category theoretic at all).



4. The social exploration and curation of formalised mathematical and scientific knowledge.

Q How do I find out about the (possibly obscure) bits of homotopy theory that I might need to build path-based models of type theory?

A Search nLab [Urs Schreiber].

For things like nLab

– is more formalisation desirable?

– is more automation (ML) possible?

Wanted

- Mathematics: the world's simplest model of univalent type theory.
- ITP: better support for quotient types (integrated with inductive types + pattern matching).
- "Social exploration": ???

Some details

Ian Orton & AMP, *Axioms for Modelling Cubical Type Theory in a Topos*. In Proc. CSL 2016, LIPIcs 62, pp. 24:1-24:19, 2016.
www.cl.cam.ac.uk/~rio22/agda/cubical-topos/root.html

Ian Orton & AMP, *Models of Type Theory Based on Moore Paths*. In Proc. FSCD 2017, LIPIcs, *to appear*, 2017.

Model of MLTT from an ordered-interval-with-monus:
www.cl.cam.ac.uk/~amp12/agda/interval-theory/Main.html