Higher inductive types

Michael Shulman

March 13, 2012

Higher inductive types

Idea

- Inductive types are a good way to build sets: we specify the elements of a set by giving constructors.
- To build a space (or ∞-groupoid), we need to specify not only elements, but paths and higher paths.
- Is there an analogous notion of higher inductive type?

```
Inductive circle : Type :=
| base : circle
| loop : (base = base).
```

Can we make sense of this?

The circle

$$\frac{}{S^1 : \mathsf{Type}}$$

$$\frac{}{\mathsf{base} : S^1} \qquad \frac{}{\mathsf{loop} : (\mathsf{base} = \mathsf{base})}$$

$$\frac{C : \mathsf{Type} \qquad b : C \qquad \ell : (b = b)}{\mathsf{match}(b, \ell) : S^1 \to C}$$

$$\frac{\vdots}{\mathsf{match}(b, \ell)(\mathsf{base}) \to_\beta b} \qquad \frac{\vdots}{\mathsf{map}(\mathsf{match}(b, \ell), \mathsf{loop}) \to_\beta \ell}$$

The circle

$$\frac{}{S^1 : \mathsf{Type}}$$

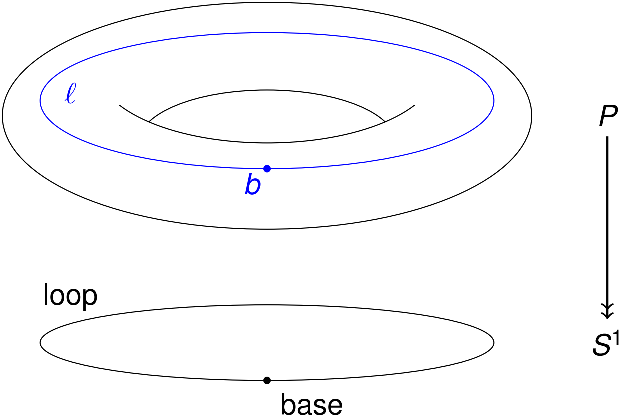
$$\frac{}{\mathsf{base} : S^1} \qquad \frac{}{\mathsf{loop} : (\mathsf{base} = \mathsf{base})}$$

$$\frac{????}{\mathsf{match}(b, \ell) : \prod_{x : S^1} P(x)}$$

Dependent loops

As hypotheses of the dependent eliminator for S^1, we need

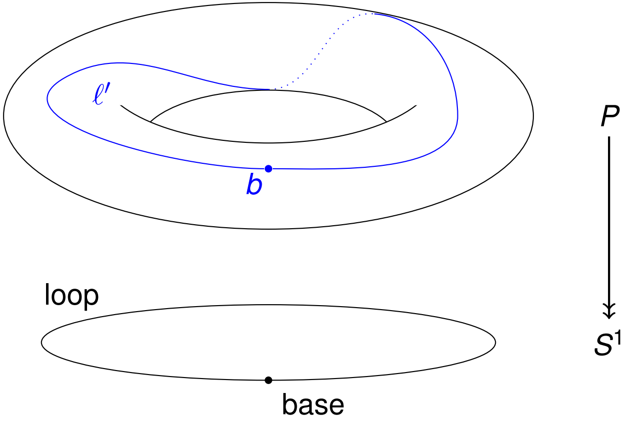
1. A point $b : P(\text{base})$.
2. A path ℓ from b to b lying over "loop".



Dependent loops

As hypotheses of the dependent eliminator for S^1, we need

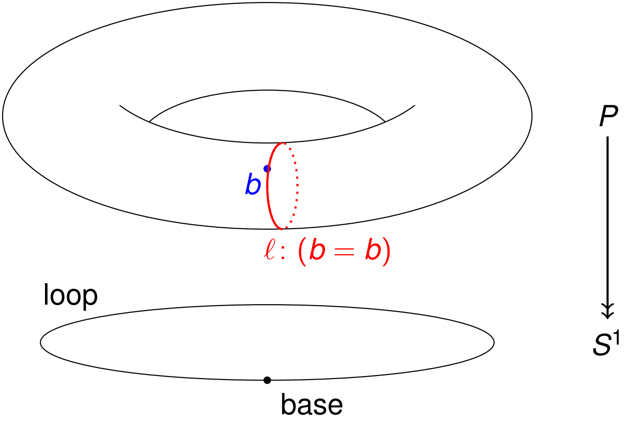
1. A point $b\colon P(\text{base})$.
2. A path ℓ from b to b lying over "loop".



Dependent loops

As hypotheses of the dependent eliminator for S^1, we need

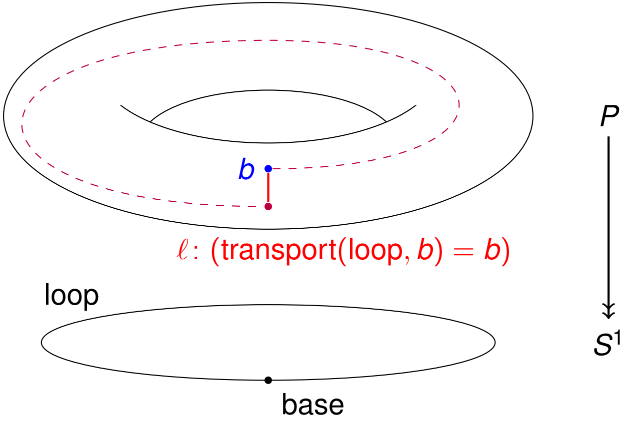
1. A point $b \colon P(\text{base})$.
2. A path ℓ from b to b lying over "loop".



Dependent loops

As hypotheses of the dependent eliminator for S^1, we need

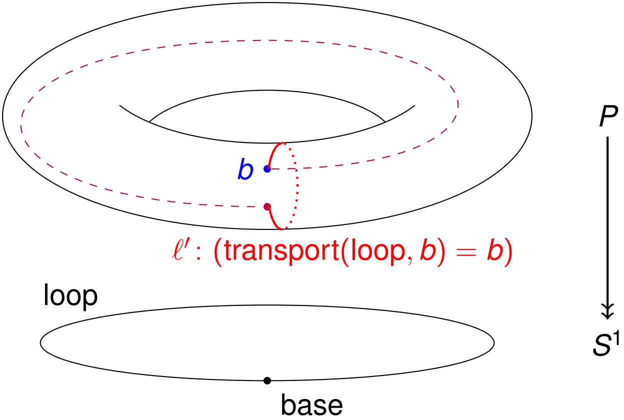
1. A point $b : P(\text{base})$.
2. A path ℓ from b to b lying over "loop".



Dependent loops

As hypotheses of the dependent eliminator for S^1, we need

1. A point $b : P(\text{base})$.
2. A path ℓ from b to b lying over "loop".



The dependent eliminator

$$\frac{x\colon S^1 \vdash P(x)\colon \mathsf{Type} \qquad \vdash b\colon P(\mathsf{base}) \qquad \vdash \ell\colon (\mathsf{transport}(\mathsf{loop}, b) = b)}{\mathsf{match}(b,\ell)\colon \prod_{x\colon S^1} P(x)}$$

$$\frac{\vdots}{\mathsf{match}(b,\ell)(\mathsf{base}) \to_\beta b} \qquad \frac{\vdots}{\mathsf{map}(\mathsf{match}(b,\ell), \mathsf{loop}) \to_\beta \ell}$$

Computation rules

- The computation rules for ordinary inductive types are definitional: they actually compute.
- To obtain rules like that for HITs would require modifying the Coq source code. As "axioms" we can only assert propositional "computation" rules, e.g. that

$$\Big(\text{match}(b, \ell)(\text{base}) = b\Big)$$

  is inhabited.

Computation rules

Even in theory, definitional computation rules for path-constructors like “loop” are a bit questionable.

$$\mathrm{map}(\mathrm{match}(b, \ell), \mathrm{loop}) \to_\beta \ell$$

- The operation “map” has many distinct (but equivalent) definitions. A definitional computation rule would single one out arbitrarily.
    - Gets worse in higher dimensions, where we need many more complicated versions of “map”.
- So far, the only way we have to construct set-theoretic models of HITs produces only propositional computation rules for path-constructors (but definitional rules for point-constructors like “base”).
- We don’t know of any application that requires a definitional computation rule for path-constructors.

The Interval

```
Inductive interval : Type :=
| zero : interval
| one : interval
| segment : (zero = one).
```

- Unsurprisingly, this type is provably contractible.
- But surprisingly, it is not useless. If it has definitional computation for its point-constructors zero and one, then it implies function extensionality.

The 2-sphere

```
Inductive sphere : Type :=
| base2 : sphere
| loop2 : (refl base2 = refl base2).
```

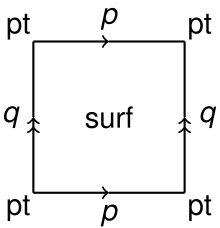
OR:

```
Inductive sphere : Type :=
| northpole : sphere
| southpole : sphere
| greenwich : (northpole = southpole)
| dateline : (northpole = southpole)
| east : (greenwich = dateline)
| west : (greenwich = dateline).
```

etc…

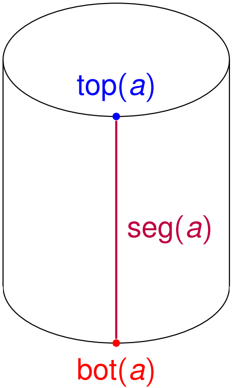
The torus

```
Inductive torus : Type :=
| pt : torus
| p : (pt = pt)
| q : (pt = pt)
| surf : (p @ q == q @ p).
```



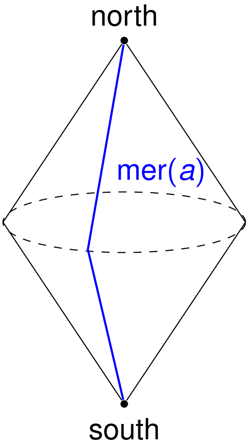
Cylinders

```
Inductive cyl (A:Type) : Type :=
| top : A -> cyl A
| bot : A -> cyl A
| seg : forall (a:A), (top a = bot a).
```



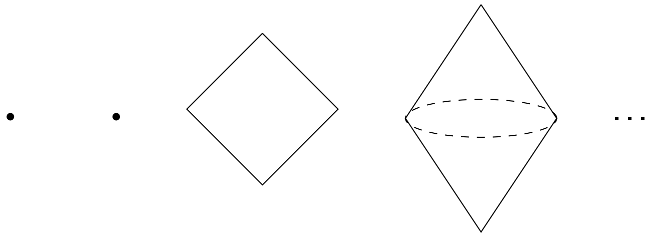
Suspension

```
Inductive susp (A:Type) : Type :=
| north : susp A
| south : susp A
| mer : A -> (north = south).
```



Higher spheres

```
Fixpoint sphere (n:nat) : Type :=
  match n with
    | 0    => unit + unit
    | S n' => susp (sphere n')
  end.
```



Nontriviality

Theorem
The type S^1 is contractible $\iff$ all types are h-sets.

Proof.
$\Leftarrow$: If S^1 is an h-set, then $\text{loop} = \text{refl}(\text{base})$.

$\Rightarrow$: Any path $p\colon (x = x)$ is the image of "loop" in S^1 under a map $S^1 \to X$, so if $\text{loop} = \text{refl}(\text{base})$ then $p = \text{refl}(x)$. □

HITs by themselves don't guarantee the homotopy theory is nontrivial. We need something else, like univalence.

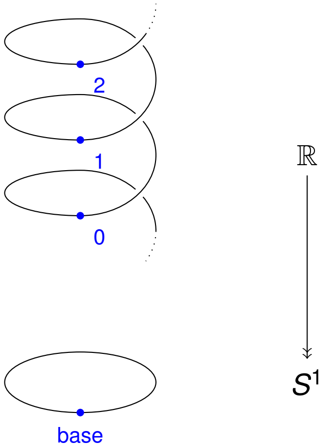
$\pi_1(S^1) \cong \mathbb{Z}$, classically

$$\pi_1(S^1) \cong \mathbb{Z}$$

How do we prove this classically?

1. Consider the winding map $\mathbb{R} \to S^1$.
2. This is the universal cover of S^1.
3. Thus, its fiber over a point, namely $\mathbb{Z}$, is $\pi_1(S^1)$.

The universal cover of S^1



$\pi_1(S^1) \cong \mathbb{Z}$, homotopically

$\pi_1(S^1) \cong \mathbb{Z}$

A more homotopy-theoretic way to phrase the classical proof:

1. We have a fibration $p\colon \mathbb{R} \to S^1$ with fiber $\mathbb{Z}$.
2. $\mathbb{R}$ is contractible, so the fiber of p is equivalent to the homotopy fiber of $* \to S^1$.
3. For any X, the homotopy fiber of $* \to X$ is the loop space

$$\Omega X = \{\gamma\colon x_0 \rightsquigarrow x_0\}$$

4. Thus $\Omega S^1 \cong \mathbb{Z}$, and in particular $\pi_1(S^1) \cong \mathbb{Z}$.

$\pi_1(S^1) \cong \mathbb{Z}$, type-theoretically

How can we build the fibration $\mathbb{R} \twoheadrightarrow S^1$ in type theory?

- A fibration over S^1 is a dependent type $R \colon S^1 \to \text{Type}$.
- By the eliminator for S^1, a function $R \colon S^1 \to \text{Type}$ is determined by
  - A point $B \colon \text{Type}$ and
  - A path $\ell \colon (B = B)$.
- By univalence, ℓ is a homotopy equivalence $B \simeq B$.

Thus we can take $B = \mathbb{Z}$ and ℓ to be "$+1$".

The rest of the proof is just the same!

Remarks

- That isn't to say it's easy to formalize it all in type theory!
- What we get is $\Omega S^1 \cong \mathbb{Z}$, which is classically stronger than $\pi_1(S^1) \cong \mathbb{Z}$. Here, we don't yet have a definition of π_1.

Work in progress

- Extend this to classifying spaces of other discrete groups.
- Prove the Freudenthal suspension theorem and conclude that $\pi_n(S^n) \cong \mathbb{Z}$.
- Construct the Hopf fibration $S^3 \twoheadrightarrow S^2$
- Construct the long exact sequence of a fibration and conclude that $\pi_3(S^2) \cong \mathbb{Z}$.
- Construct the Serre spectral sequence.
- …

Supports

Recall: A is (-1)-truncated, or an h-prop, if

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

The support of A, denoted $\mathrm{supp}(A)$, is supposed to be:

- an h-prop that contains a point precisely when A does.
- a reflection of A into h-props.

Support as an HIT

Peter Lumsdaine realized we could define:

```
Inductive supp (A:Type) : Type :=
| inhab : A -> supp A
| inhabpath : forall (x y:supp A), (x = y).
```

$$\frac{A\colon \mathrm{Type}}{\mathrm{supp}(A)\colon \mathrm{Type}}$$

$$\frac{x\colon A}{\mathrm{inhab}(x)\colon \mathrm{supp}(A)} \qquad \frac{x\colon A \qquad y\colon A}{\mathrm{inhabpath}(x,y)\colon (x=y)}$$

$$\frac{C\colon \mathrm{Type} \qquad x\colon A \vdash i(x)\colon C \qquad x\colon C, y\colon C \vdash p\colon (x=y)}{\mathrm{match}(i,p)\colon \mathrm{supp}(A) \to C}$$

The rest of logic

$$
\begin{array}{rcl}
P \text{ and } Q & \longleftrightarrow & P \times Q \\
P \text{ implies } Q & \longleftrightarrow & P \to Q \\
\top \text{ (true)} & \longleftrightarrow & \text{unit} \\
\bot \text{ (false)} & \longleftrightarrow & \emptyset \\
(\forall x : A) P(x) & \longleftrightarrow & \prod_{x:A} B(x) \\
 & & \\
P \text{ or } Q & \longleftrightarrow & \text{supp}(P + Q) \\
(\exists x : A) P(x) & \longleftrightarrow & \text{supp}(\sum_{x:A} B(x))
\end{array}
$$

0-truncation

Recall: A is an h-set if

$$\prod_{\substack{x,y\,:\,A \\ p,q\,:\,(x=y)}} (p = q)$$

```
Inductive pi0 (A:Type) : Type :=
| component : A -> pi0 A
| pi0path : forall (x y:pi0 A) (p q:x=y), (p=q).
```

- Can now define $\pi_1(A) :\equiv \pi_0(\Omega A)$
- And so on . . .

Truncation: summary

- Because we were (magically) able to characterize h-props, h-sets, etc. only using equality types, we can use path-constructors in HITs to “universally force” a type to be an h-prop, an h-set, etc.
- Because “being an h-prop” is itself an h-prop, the path-constructors that force an HIT to be an h-prop have no other effect (give no extra data).
- We can do the same thing with equivalences!

Localization

Given $f\colon A \to B$.

Definition

- Z is *f-local* if $\mathrm{Map}(B, Z) \xrightarrow{f^*} \mathrm{Map}(A, Z)$ is an equivalence.
- An *f-localization* of X is a reflection of X into f-local spaces.

Examples

- If f is $S^n \to D^{n+1}$, then f-local means $(n-1)$-truncated.
- Using $S^1 \xrightarrow{\times p} S^1$ for a prime p, we can build "localization of spaces at p".

Adjoint equivalences

Recall: $f\colon A \to B$ is an adjoint equivalence if we have

- A map $g\colon B \to A$
- A homotopy $r\colon \prod_{a\colon A}(g(f(a)) = a)$
- A homotopy $s\colon \prod_{b\colon B}(f(g(b)) = b)$
- A 2-homotopy $\prod_{a\colon A}(s(f(a)) = \mathrm{map}(f, r(a)))$

The type isAdjointEquiv(f) of such data is always an h-prop.

Localization as a HIT

```
Inductive localize {A B:Type} {f:A->B}
    (X:Type) : Type :=
| to_local : X -> localize X
| linv : (A -> localize X) -> B -> localize X
| lsec : forall (g:A -> localize X) (a:A),
    (linv g (f a) = g a)
| lret : forall (h:B -> localize X) (b:B),
    (linv (h o f) b = h b)
| ltri : forall (h:B -> localize X) (a:A),
    (lsec (h o f) a = lret h (f a)).
```

linv gives $\mathrm{Map}(A, Z) \to \mathrm{Map}(B, Z)$, and the other constructors make it an adjoint inverse to f^*.

The other factorization

Recall:

- A model category has two weak factorization systems:

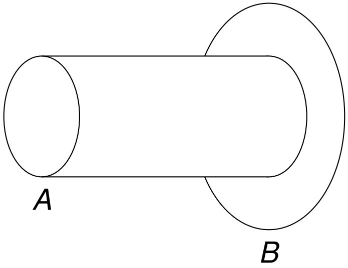
  (acyclic cofibrations, fibrations)
  (cofibrations, acyclic fibrations)

- With identity types, we have the first WFS using the mapping path space:

$$A \to [\![y\colon B,\ x\colon A,\ p\colon (g(x) = y)]\!] \to B$$

- In topology, the second WFS uses the mapping cylinder.

Mapping cylinders




```
Inductive mcyl {A B:Type} (f:A->B) : Type :=
| inl : A -> mcyl f
| inr : B -> mcyl f
| glue : forall (a:A), (inl a = inr (f a)).
```

Does this give us the other WFS in type theory?

Acyclic fibrations

What is an acyclic fibration in type theory?

1. A fibration that is also an equivalence.
2. A fibration $p\colon B \twoheadrightarrow A$ which admits a section $s\colon A \to B$ (hence $ps = 1_A$) such that $sp \sim 1_B$.
3. A dependent type $B\colon A \to \mathrm{Type}$ such that each $B(a)$ is contractible.

Cofibrations

What is a cofibration in type theory?

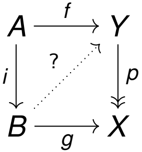
Actually, what is an acyclic cofibration in type theory?

When does $i\colon A \to B$ satisfy $i \boxslash p$ for any fibration p?

Acyclic cofibrations

Theorem (Gambino-Garner)

If B is an inductive type and i is its only constructor, then $i \boxslash p$ for any fibration p.



Proof.

- p is a dependent type $Y : X \to \text{Type}$; we want to define

$$h : \prod_{b : B} Y(g(b))$$

- By induction, it suffices to specify $h(b)$ when $b = i(a)$.
- But then we can take $h(i(a)) :\equiv f(a)$.

□

Path object factorizations

Example

refl: $A \to \mathrm{Id}_A$ is the only constructor of the identity type. Thus,

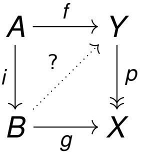
$$A \xrightarrow{\text{refl}} \mathrm{Id}_A \twoheadrightarrow A \times A$$

is an (acyclic cofibration, fibration) factorization.

Some cofibrations

Theorem

If B is an inductive type and $i\colon A \to B$ is one of its constructors, then $i \boxslash p$ for any acyclic fibration p.



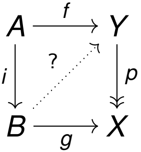
Proof.

- Now we have a section $s\colon \prod_{x\colon X} Y(x)$.
- We define $h\colon \prod_{b\colon B} Y(g(b))$ by induction on B:
  - If $b = i(a)$, take $h(b) :\equiv f(a)$.
  - If b is some other constructor, take $h(b) :\equiv s(g(b))$.

$\square$

Theorem

If B is a higher inductive type and $i\colon A \to B$ *is one of its point-constructors, then* $i \boxslash p$ *for any acyclic fibration p.*



Proof.

- Now we have a section $s\colon \prod_{x\colon X} Y(x)$.
- We define $h\colon \prod_{b\colon B} Y(g(b))$ by induction on B:
  - If $b = i(a)$, take $h(b) :\equiv f(a)$.
  - If b is some other point-constructor, take $h(b) :\equiv s(g(b))$.
  - In the case of path-constructors, use the contractibility of the fibers of p.

□

Cell complexes

In homotopy theory, $i\colon A \to B$ is a relative cell complex if B is obtained from A by successively "gluing on disks along their boundaries".

$$\begin{array}{ccccccc}
\coprod S^{n-1} & \longrightarrow & \coprod D^n & & & & \\
\downarrow & & \ulcorner\downarrow & & & & \\
A & \longrightarrow & B_1 & \longrightarrow & \cdots & \longrightarrow & B
\end{array}$$

These are cofibrations, for the same reason as in type theory.

fibrations	$\longleftrightarrow$	dependent types
cofibrations	$\longleftrightarrow$	inductive constructors

The other factorization

We need a mapping cylinder for $f\colon A \to B$ that is dependent over B.

```
Inductive mcyl {A B:Type} (f:A->B) : B -> Type :=
| inr : forall (b:B), mcyl f b
| inl : forall (a:A), mcyl f (f a)
| glue : forall (a:A), (inl a = inr (f a)).
```

Theorem (Lumsdaine)

1. $\mathrm{inl}\colon A \to [\![\mathrm{mcyl(f)}]\!]$ *is a cofibration.*
2. $[\![\mathrm{mcyl(f)}]\!] \twoheadrightarrow B$ *is an acyclic fibration.*
3. *With HITs, the syntactic category is a model category (except for limits and colimits), in which all objects are fibrant and cofibrant.*

Categorical models

Recall that a category which

1. admits finite limits and colimits
2. has a well-behaved WFS,
3. is compatibly locally cartesian closed, and
4. has a univalent universe

models homotopy type theory.

Theorem (Lumsdaine–Shulman)

If such a category is "locally presentable" and its WFS underlies a nice model structure, then it models all higher inductive types.

(In particular, classical homotopy theory via simplicial sets.)