Formalized Spectral Sequences in Homotopy Type Theory

Floris van Doorn

Carnegie Mellon University

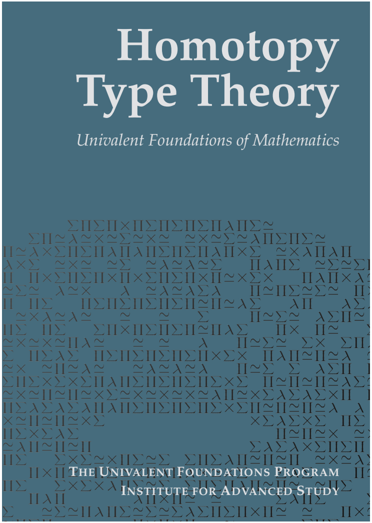
September 14, 2017

Joint work with Jeremy Avigad, Steve Awodey, Ulrik Buchholtz, Egbert Rijke and Mike Shulman.

Introduction

- In the logic of Homotopy Type Theory (HoTT) (today)
- we constructed the cohomological Atiyah-Hirzebruch and Serre spectral sequences
- and formalize it in the proof assistant Lean

Homotopy Type Theory



homotopytypetheory.org/book/

Homotopy Type Theory

Homotopy type theory (HoTT) refers to the homotopical interpretation of Martin Löf type theory by Awodey, Warren and Voevodsky.

In HoTT we can do *Synthetic Homotopy Theory*:

Study types in type theory as spaces in homotopy theory.

Advantages over regular homotopy theory:

- More general
- Constructive
- Feasible to formalize
- Novel ways of reasoning

Martin Löf Type Theory

In Martin Löf type theory

- there are types: $\mathbb{N}$, $\mathbb{Z}$, $\mathbb{R} \times \mathbb{R}$, vector $\mathbb{C}$ 3;
- there are terms: $3 : \mathbb{N}$, $(\sqrt{2}, e) : \mathbb{R} \times \mathbb{R}$, $(i, 0, e^i)$: vector $\mathbb{C}$ 3;
- terms have a unique type. The $7 : \mathbb{N}$ is not the same object as the $7 : \mathbb{R}$;
- there is a notion of computation: $2 + 3$ computes to 5 in $\mathbb{N}$ and $\pi_2(\sqrt{2}, e)$ computes to e in $\mathbb{R}$;
- the congruence closure of computation is denoted by $\equiv$. So $\pi_2(0, 2 + 3) \equiv 5$.

Function Types

If A and B are types, we can form the function types $A \to B$.

The function $x \mapsto f(x)$ is denoted $\lambda x.f\ x$.

Now $(\lambda x.b)a$ (where b can depend on x) computes to $b[a/x]$ (where we substitute a for x in b).

Dependent Function Types

Suppose we have a *type family* $P(x)$, which is a type depending on a term $x : A$ (such as vector, which depends on a type and a natural number).

We can form the *dependent function type* $\Pi(x : A),\ P(x)$ consisting of functions which sends $x : A$ to an element in $P(x)$.

Example $\Pi(n : \mathbb{N}),$ vector $\mathbb{R}\ n$ consists of functions which give a choice of a vector of length n for each natural number $\mathbb{N}$.

If zeroes $:\equiv \lambda n.\, \underbrace{(0, 0, \ldots, 0)}_{n}$, then

$$\text{zeroes} : \Pi(n : \mathbb{N}),\ \text{vector}\ \mathbb{R}\ n$$

and zeroes$(3) \equiv (0, 0, 0)$.

Dependent Pair Types

If A and B are types, we can form the cartesian product $A \times B$.

More generally, if we have a type family $P(x)$ depending on $x : A$ we can form the *dependent pair type* $\Sigma(x : A),\ P(x)$ consisting of pairs (a, b) with $a : A$ and $b : P(a)$.

Example $(4, (1, 2, 3, 4)) : \Sigma(n : \mathbb{N}),$ vector $\mathbb{R}\ n$.

$\Sigma(A : \mathrm{Type}),\ A$ consists of *pointed types*: pairs consisting of a type A and a term $a : A$.

Curry-Howard Isomorphism

Propositions can be interpreted as types

	type theory	**logic**
A	type	proposition
$a : A$	term/element	proof
$A + B$	sum type	disjunction $\vee$
$A \times B$	product type	conjunction $\wedge$
$A \to B$	function type	implication $\to$
$P : A \to \mathrm{Type}$	type family	predicate
$\Sigma(x : A),\ P(x)$	sigma type	existential quantifier $\exists$
$\Pi(x : A),\ P(x)$	dependent function type	universal quantifier $\forall$
$a =_A b$	identity type	equality

Curry-Howard Isomorphism

Examples addition commutes: $\Pi(n\ m : \mathbb{N}),\ n + m = m + n$

property of inequality:

$\Pi(n\ m : \mathbb{N}),\ n \leq m \to \Sigma(k : \mathbb{N}),\ n + k = m$

$\Sigma(n : \mathbb{Z}),\ n < 0$ is the type of negative integers (or the proposition that there exists a negative integer).

Homotopy Type Theory

Homotopy Type Theory offers a different interpretation of types.

	type theory	**logic**	**homotopy theory**
A	type	formula	space*
$a : A$	term/element	proof	point
$A \times B$	product type	conjunction	binary product sp.
$A \to B$	function type	implication	mapping space
$P : A \to \text{Type}$	dependent type	predicate	fibration
$\Sigma(x : A),\ P(x)$	sigma type	ex. quantifier	total space
$\Pi(x : A),\ P(x)$	dep. fn. type	un. quantifier	product space
$a =_A b$	identity type	equality	path space

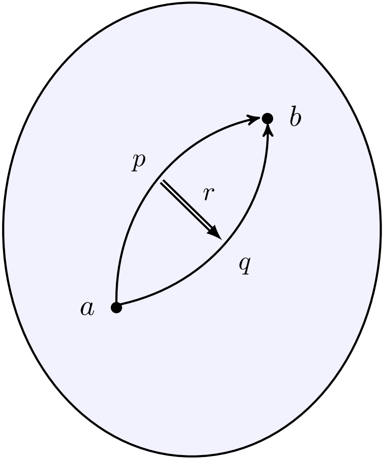
I will use these notions interchangably.

Types as Spaces

A type A can have

- points $a, b : A$
- paths $p, q : a = b$
- paths between paths $r : p = q$

$$\vdots$$



Identity Type

The identity type is central in homotopy type theory.

It corresponds to *equality* in logic and to the *path space* in homotopy theory.

Different ways to think about the identity type:

- **Type theory**: The identity type $a =_A (-)$ is generated by reflexivity: $\mathsf{refl}_a : a =_A a$.
- **Logic**: Equality is the least (free) reflexive relation.
- **Homotopy theory**: The path space with one point fixed is contractible.

(This does not mean every proof of equality is reflexivity)

Path Induction

This is made precise by *path induction*:

- If $C : \Pi(x : A),\ a = x \to \mathrm{Type}$,
- to prove/construct an element of $\Pi(x : A).\Pi(p : a = x),\ C(x, p)$
- it is sufficient to prove/construct an element of $C(a, \mathsf{refl}_a)$

What this means is that if we have a path $p : a = x$ where the right endpoint is a variable, we may assume that p is reflexivity and that x is a.

Path Induction

Example If A is a type with points x, y and z. If $p : x = y$ and $q : y = z$, we have a *concatenation* $p \cdot q : x = z$.

Path Induction

Example If A is a type with points x, y and z. If $p : x = y$ and $q : y = z$, we have a *concatenation* $p \cdot q : x = z$.

Proof Since the right endpoint of q is a variable, we may assume q is reflexivity and that y is z. Then we need to construct $p \cdot \mathsf{refl}_y : x = y$, which we define as $p \cdot \mathsf{refl}_y :\equiv p$.

Path Induction

Example If A is a type with points x, y and z. If $p : x = y$ and $q : y = z$, we have a *concatenation* $p \cdot q : x = z$.

Proof Since the right endpoint of q is a variable, we may assume q is reflexivity and that y is z. Then we need to construct $p \cdot \mathsf{refl}_y : x = y$, which we define as $p \cdot \mathsf{refl}_y :\equiv p$.

```
variables {A : Type} {w x y z : A}
definition concat (p : x = y) (q : y = z) : x = z :=
by induction q; exact p
```

Path Induction

Example If A is a type with points x, y and z. If $p : x = y$ and $q : y = z$, we have a *concatenation* $p \cdot q : x = z$.

Proof Since the right endpoint of q is a variable, we may assume q is reflexivity and that y is z. Then we need to construct $p \cdot \mathsf{refl}_y : x = y$, which we define as $p \cdot \mathsf{refl}_y :\equiv p$.

```
variables {A : Type} {w x y z : A}
definition concat (p : x = y) (q : y = z) : x = z :=
by induction q; exact p

definition con.assoc (p : w = x) (q : x = y) (r : y = z) :
  (p · q) · r = p · (q · r) :=
by induction r; reflexivity
```

Path Induction

Example If A is a type with points x, y and z. If $p : x = y$ and $q : y = z$, we have a *concatenation* $p \cdot q : x = z$.

Proof Since the right endpoint of q is a variable, we may assume q is reflexivity and that y is z. Then we need to construct $p \cdot \mathsf{refl}_y : x = y$, which we define as $p \cdot \mathsf{refl}_y :\equiv p$.

```
variables {A : Type} {w x y z : A}
definition concat (p : x = y) (q : y = z) : x = z :=
by induction q; exact p

definition con.assoc (p : w = x) (q : x = y) (r : y = z) :
  (p · q) · r = p · (q · r) :=
by induction r; reflexivity
```

After induction on `r`, the goal is `(p · q) · refl y = p · (q · refl y)`

Identity Type of Type Constructors

We can characterize the equality in various types.

$$
\begin{array}{rcll}
(a,b) =_{A\times B} (a',b') & \text{is} & (a =_A a') \times (b =_B b') & \\
f = g & \text{is} & \Pi x,\ f(x) = g(x) & \text{(function extensionality)} \\
A =_{\mathrm{Type}} B & \text{is} & A \simeq B & \text{(univalence, Voevodsky)}
\end{array}
$$

This is done in *cubical type theory*.

Truncated Types

Some types are *truncated*, which means all higher paths are trivial in theese types.

A type A is *contractible* ((-2)-type) if it has exactly one element, if

$$\Sigma(x : A),\ \Pi(y : A),\ x = y.$$

A type A is a *proposition* ((-1)-type) if it has at most one element, if

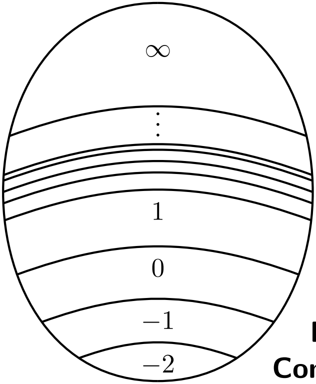
$$\Pi(x\ y : A),\ x = y.$$

In either of the above cases, all (higher) paths in A are trivial.

A is a *set* (0-type) if for all $x\ y : A$ the type $x = y$ is a proposition.

A is an $(n + 1)$-type if for all $x\ y : A$ the type $x = y$ is an n-type.

Truncated Types (Voevodsky)



$(n+1)$-**Type**: all paths n-types

1-Type: all paths are sets

Set: satisfies UIP / axiom K

Proposition: as at most one point

Contractible: has exactly one point

Truncation

Given A, we can form the n-*truncation* $\|A\|_n$.

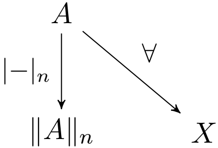
$\|A\|_n$ is the "best approximation" of A which is n-truncated.

$$\begin{array}{c} A \\ {\scriptstyle |-|_n}\big\downarrow \\ \|A\|_n \end{array}$$

Truncation

Given A, we can form the n-*truncation* $\|A\|_n$.

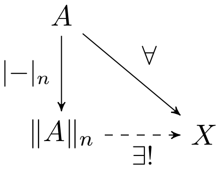
$\|A\|_n$ is the "best approximation" of A which is n-truncated.



Truncation

Given A, we can form the n-*truncation* $\|A\|_n$.

$\|A\|_n$ is the "best approximation" of A which is n-truncated.



Fundamental group

Given a pointed type (A, a_0). The type $\Omega A :\equiv (a_0 = a_0, \mathsf{refl}_{a_0})$ has

- multiplication given by concatenation
- inverses given by path inverses
- identity given by reflexivity

These operations satisfy the group laws.

It is not quite a group since it has higher structure. For example, there might be multiple proofs that $(p \cdot q) \cdot r = p \cdot (q \cdot r)$.

Taking the set-truncation "kills off" this higher structure. Therefore, $\pi_1(A) :\equiv \|\Omega A\|_0$ forms a group, the *fundamental group*.

The higher homotopy groups are $\pi_n(A) :\equiv \|\Omega^n A\|_0$.

Fundamental group

Traditionally, the definition of the fundamental group of (A, a_0) consists of continuous functions $f : [0,1] \to A$ such that $f(0) = f(1) = a_0$ modulo homotopy.

In HoTT, we don't need to define real numbers, the interval, continuous functions or homotopy of paths to define the fundamental group.

Traditionally, if you want to define an operation on the fundamental group, you have to prove that the operation does the same on homotopic paths.

In HoTT all functions automatically respect homotopic paths, since they are equal.

Higher Inductive Types

In Type Theory there are *inductive types*, in which you specify its points.

Examples. $\mathbb{N}$ is generated by 0 and succ
$A + B$ is generated by either $a : A$ or $b : B$
$a =_A (-)$ is generated by $\mathsf{refl}_a : a =_A a$

In homotopy theory we can build cell complexes inductively.

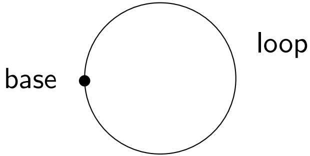
In HoTT we can combine these into *higher inductive types* [Shulman, Lumsdaine, 2012].

The circle

Example. The circle $\mathbb{S}^1$

HIT $\mathbb{S}^1 :\equiv$

- base : $\mathbb{S}^1$
- loop : base $=$ base



Recursion Principle. To define $f : \mathbb{S}^1 \to A$ we need to define $f(\text{base}) : A$ and a path $f(\text{base}) = f(\text{base})$ which is the path showing that f respects loop.

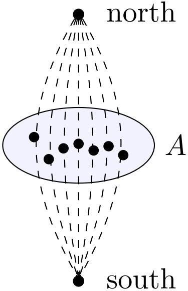
Using univalence, we can prove $\pi_1(\mathbb{S}^1) = \mathbb{Z}$.

The suspension

Example. The suspension ΣA

HIT $\Sigma A :\equiv$

- $\mathrm{north}, \mathrm{south} : \Sigma A$
- $\mathrm{merid} : A \to (\mathrm{north} = \mathrm{south})$



Remark. $\mathbb{S}^1 \simeq \Sigma\mathbf{2}$

Definition. The n-sphere is defined by $\mathbb{S}^{n+1} :\equiv \Sigma\mathbb{S}^n$ and $\mathbb{S}^0 :\equiv \mathbf{2}$

Homotopy groups of spheres

	$\mathbb{S}^0$	$\mathbb{S}^1$	$\mathbb{S}^2$	$\mathbb{S}^3$	$\mathbb{S}^4$	$\mathbb{S}^5$	$\mathbb{S}^6$	$\mathbb{S}^7$	$\mathbb{S}^8$
π_1	0	$\mathbb{Z}$	0	0	0	0	0	0	0
π_2	0	0	$\mathbb{Z}$	0	0	0	0	0	0
π_3	0	0	$\mathbb{Z}$	$\mathbb{Z}$	0	0	0	0	0
π_4	0	0	$\mathbb{Z}_2$	$\mathbb{Z}_2$	$\mathbb{Z}$	0	0	0	0
π_5	0	0	$\mathbb{Z}_2$	$\mathbb{Z}_2$	$\mathbb{Z}_2$	$\mathbb{Z}$	0	0	0
π_6	0	0	$\mathbb{Z}_{12}$	$\mathbb{Z}_{12}$	$\mathbb{Z}_2$	$\mathbb{Z}_2$	$\mathbb{Z}$	0	0
π_7	0	0	$\mathbb{Z}_2$	$\mathbb{Z}_2$	$\mathbb{Z}\times\mathbb{Z}_{12}$	$\mathbb{Z}_2$	$\mathbb{Z}_2$	$\mathbb{Z}$	0
π_8	0	0	$\mathbb{Z}_2$	$\mathbb{Z}_2$	$\mathbb{Z}_2^2$	$\mathbb{Z}_{24}$	$\mathbb{Z}_2$	$\mathbb{Z}_2$	$\mathbb{Z}$
π_9	0	0	$\mathbb{Z}_3$	$\mathbb{Z}_3$	$\mathbb{Z}_2^2$	$\mathbb{Z}_2$	$\mathbb{Z}_{24}$	$\mathbb{Z}_2$	$\mathbb{Z}_2$
π_{10}	0	0	$\mathbb{Z}_{15}$	$\mathbb{Z}_{15}$	$\mathbb{Z}_{24}\times\mathbb{Z}_3$	$\mathbb{Z}_2$	0	$\mathbb{Z}_{24}$	$\mathbb{Z}_2$
π_{11}	0	0	$\mathbb{Z}_2$	$\mathbb{Z}_2$	$\mathbb{Z}_{15}$	$\mathbb{Z}_2$	$\mathbb{Z}$	0	$\mathbb{Z}_{24}$
π_{12}	0	0	$\mathbb{Z}_2^2$	$\mathbb{Z}_2^2$	$\mathbb{Z}_2$	$\mathbb{Z}_{30}$	$\mathbb{Z}_2$	0	0
π_{13}	0	0	$\mathbb{Z}_{12}\times\mathbb{Z}_2$	$\mathbb{Z}_{12}\times\mathbb{Z}_2$	$\mathbb{Z}_2^3$	$\mathbb{Z}_2$	$\mathbb{Z}_{60}$	$\mathbb{Z}_2$	0

Homotopy groups of spheres

	$\mathbb{S}^0$	$\mathbb{S}^1$	$\mathbb{S}^2$	$\mathbb{S}^3$	$\mathbb{S}^4$	$\mathbb{S}^5$	$\mathbb{S}^6$	$\mathbb{S}^7$	$\mathbb{S}^8$
π_1	0	$\mathbb{Z}$	0	0	0	0	0	0	0
π_2	0	0	$\mathbb{Z}$	0	0	0	0	0	0
π_3	0	0	$\mathbb{Z}$	$\mathbb{Z}$	0	0	0	0	0
π_4	0	0	$\mathbb{Z}_2$	$\mathbb{Z}_2$	$\mathbb{Z}$	0	0	0	0
π_5	0	0	$\mathbb{Z}_2$	$\mathbb{Z}_2$	$\mathbb{Z}_2$	$\mathbb{Z}$	0	0	0
π_6	0	0	$\mathbb{Z}_{12}$	$\mathbb{Z}_{12}$	$\mathbb{Z}_2$	$\mathbb{Z}_2$	$\mathbb{Z}$	0	0
π_7	0	0	$\mathbb{Z}_2$	$\mathbb{Z}_2$	$\mathbb{Z}\times\mathbb{Z}_{12}$	$\mathbb{Z}_2$	$\mathbb{Z}_2$	$\mathbb{Z}$	0
π_8	0	0	$\mathbb{Z}_2$	$\mathbb{Z}_2$	$\mathbb{Z}_2^2$	$\mathbb{Z}_{24}$	$\mathbb{Z}_2$	$\mathbb{Z}_2$	$\mathbb{Z}$
π_9	0	0	$\mathbb{Z}_3$	$\mathbb{Z}_3$	$\mathbb{Z}_2^2$	$\mathbb{Z}_2$	$\mathbb{Z}_{24}$	$\mathbb{Z}_2$	$\mathbb{Z}_2$
π_{10}	0	0	$\mathbb{Z}_{15}$	$\mathbb{Z}_{15}$	$\mathbb{Z}_{24}\times\mathbb{Z}_3$	$\mathbb{Z}_2$	0	$\mathbb{Z}_{24}$	$\mathbb{Z}_2$
π_{11}	0	0	$\mathbb{Z}_2$	$\mathbb{Z}_2$	$\mathbb{Z}_{15}$	$\mathbb{Z}_2$	$\mathbb{Z}$	0	$\mathbb{Z}_{24}$
π_{12}	0	0	$\mathbb{Z}_2^2$	$\mathbb{Z}_2^2$	$\mathbb{Z}_2$	$\mathbb{Z}_{30}$	$\mathbb{Z}_2$	0	0
π_{13}	0	0	$\mathbb{Z}_{12}\times\mathbb{Z}_2$	$\mathbb{Z}_{12}\times\mathbb{Z}_2$	$\mathbb{Z}_2^3$	$\mathbb{Z}_2$	$\mathbb{Z}_{60}$	$\mathbb{Z}_2$	0

Homotopy groups of spheres

	$\mathbb{S}^0$	$\mathbb{S}^1$	$\mathbb{S}^2$	$\mathbb{S}^3$	$\mathbb{S}^4$	$\mathbb{S}^5$	$\mathbb{S}^6$	$\mathbb{S}^7$	$\mathbb{S}^8$
π_1	0	$\mathbb{Z}$	0	0	0	0	0	0	0
π_2	0	0	$\mathbb{Z}$	0	0	0	0	0	0
π_3	0	0	$\mathbb{Z}$	$\mathbb{Z}$	0	0	0	0	0
π_4	0	0	$\mathbb{Z}_2$	$\mathbb{Z}_2$	$\mathbb{Z}$	0	0	0	0
π_5	0	0	$\mathbb{Z}_2$	$\mathbb{Z}_2$	$\mathbb{Z}_2$	$\mathbb{Z}$	0	0	0
π_6	0	0	$\mathbb{Z}_{12}$	$\mathbb{Z}_{12}$	$\mathbb{Z}_2$	$\mathbb{Z}_2$	$\mathbb{Z}$	0	0
π_7	0	0	$\mathbb{Z}_2$	$\mathbb{Z}_2$	$\mathbb{Z}\times\mathbb{Z}_{12}$	$\mathbb{Z}_2$	$\mathbb{Z}_2$	$\mathbb{Z}$	0
π_8	0	0	$\mathbb{Z}_2$	$\mathbb{Z}_2$	$\mathbb{Z}_2^2$	$\mathbb{Z}_{24}$	$\mathbb{Z}_2$	$\mathbb{Z}_2$	$\mathbb{Z}$
π_9	0	0	$\mathbb{Z}_3$	$\mathbb{Z}_3$	$\mathbb{Z}_2^2$	$\mathbb{Z}_2$	$\mathbb{Z}_{24}$	$\mathbb{Z}_2$	$\mathbb{Z}_2$
π_{10}	0	0	$\mathbb{Z}_{15}$	$\mathbb{Z}_{15}$	$\mathbb{Z}_{24}\times\mathbb{Z}_3$	$\mathbb{Z}_2$	0	$\mathbb{Z}_{24}$	$\mathbb{Z}_2$
π_{11}	0	0	$\mathbb{Z}_2$	$\mathbb{Z}_2$	$\mathbb{Z}_{15}$	$\mathbb{Z}_2$	$\mathbb{Z}$	0	$\mathbb{Z}_{24}$
π_{12}	0	0	$\mathbb{Z}_2^2$	$\mathbb{Z}_2^2$	$\mathbb{Z}_2$	$\mathbb{Z}_{30}$	$\mathbb{Z}_2$	0	0
π_{13}	0	0	$\mathbb{Z}_{12}\times\mathbb{Z}_2$	$\mathbb{Z}_{12}\times\mathbb{Z}_2$	$\mathbb{Z}_2^3$	$\mathbb{Z}_2$	$\mathbb{Z}_{60}$	$\mathbb{Z}_2$	0

Pointed Maps

Definition If (X, x_0) and (Y, y_0) are pointed types, a the type of *pointed maps* $(X, x_0) \to^* (Y, y_0)$ is defined as $\Sigma(f : X \to Y),\ f(x_0) = y_0$.

Definition If $f : X \to^* Y$ is a pointed map, its *(homotopy) fiber* fib_f is defined as $\Sigma(x : X),\ f(x) = y_0$

Long Exact Sequence of Homotopy Groups

Given a pointed map $f : X \to^* Y$ with fiber F.
Then we have the following long exact sequence.

$$
\begin{array}{ccccc}
 & & \vdots & & \\
\pi_2(F) & \xrightarrow[\pi_2(p_1)]{} & \pi_2(X) & \xrightarrow{\pi_2(f)} & \pi_2(Y) \\
 & \swarrow^{\pi_1(\delta)} & & & \\
\pi_1(F) & \xrightarrow[\pi_1(p_1)]{} & \pi_1(X) & \xrightarrow{\pi_1(f)} & \pi_1(Y) \\
 & \swarrow^{\pi_0(\delta)} & & & \\
\pi_0(F) & \xrightarrow[\pi_0(p_1)]{} & \pi_0(X) & \xrightarrow{\pi_0(f)} & \pi_0(Y)
\end{array}
$$

We do *not* need to assume that f is a “fibration”.

Thank you