Computer-Checked Proofs in the Logic of Homotopy Theory

Dan Licata
Institute for Advanced Study

Homotopy theory

A branch of topology,
the study of spaces and continuous deformations

[image from wikipedia]

Homotopy

Deformation of one path into another

a

p

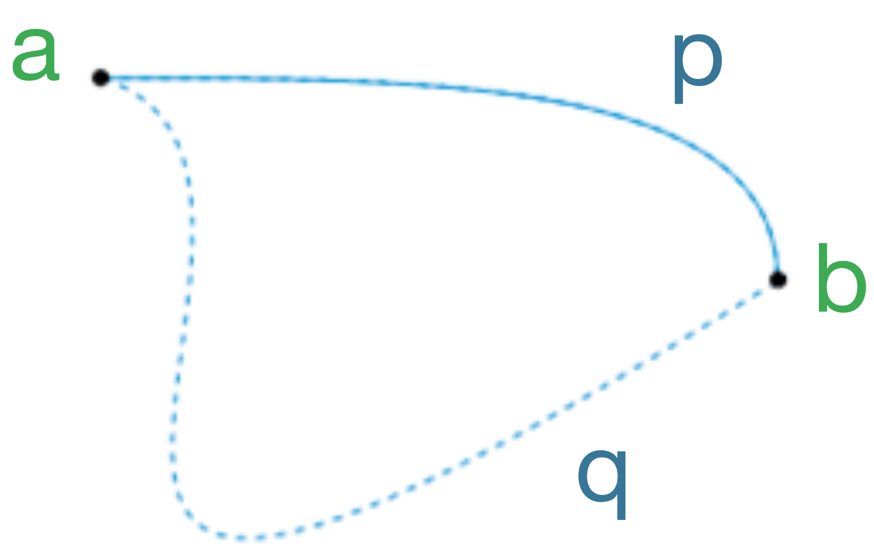
b

q

[image from wikipedia]

Homotopy

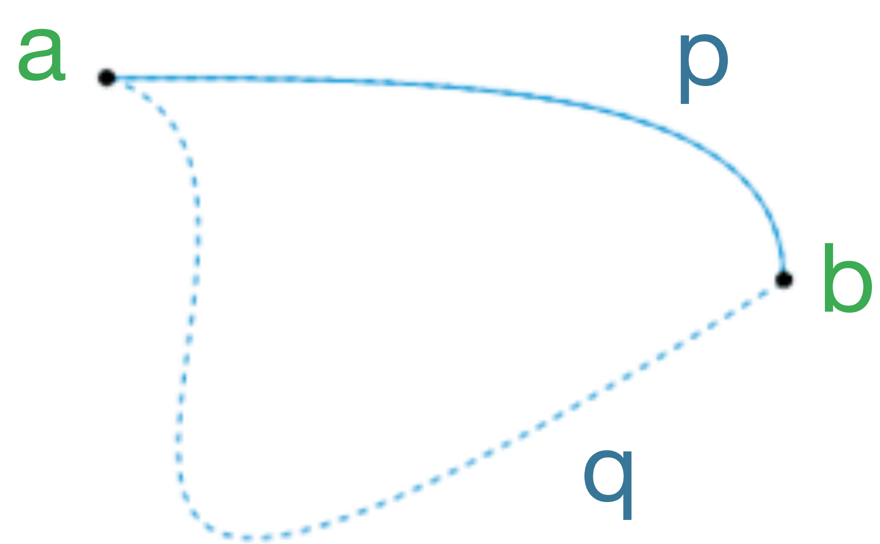
Deformation of one path into another



[image from wikipedia]

Homotopy

Deformation of one path into another

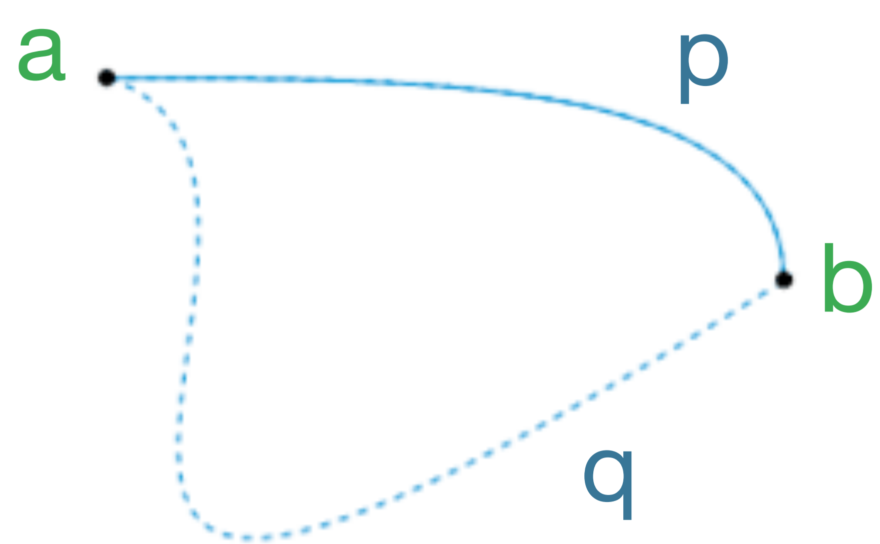


= 2-dimensional *path between paths*

[image from wikipedia]

Homotopy

Deformation of one path into another



= 2-dimensional *path between paths*

Homotopy theory is the study of spaces by way of their paths, homotopies, homotopies between homotopies, ….

[image from wikipedia]

Synthetic vs Analytic

Synthetic geometry (Euclid)

POSTULATES.

I.

LET it be granted that a straight line may be drawn from any one point to any other point.

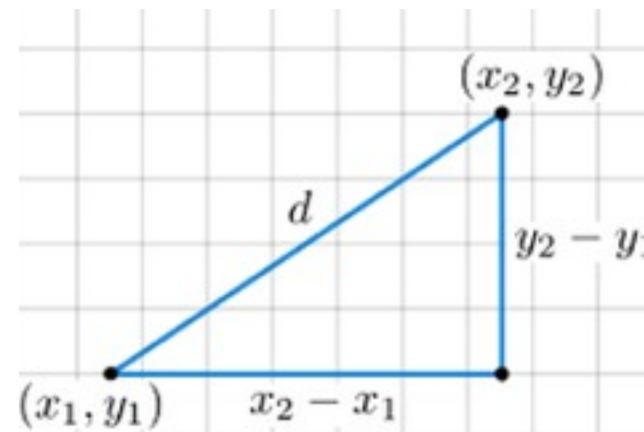
II.

That a terminated straight line may be produced to any length in a straight line.

III.

And that a circle may be described from any centre, at any distance from that centre.

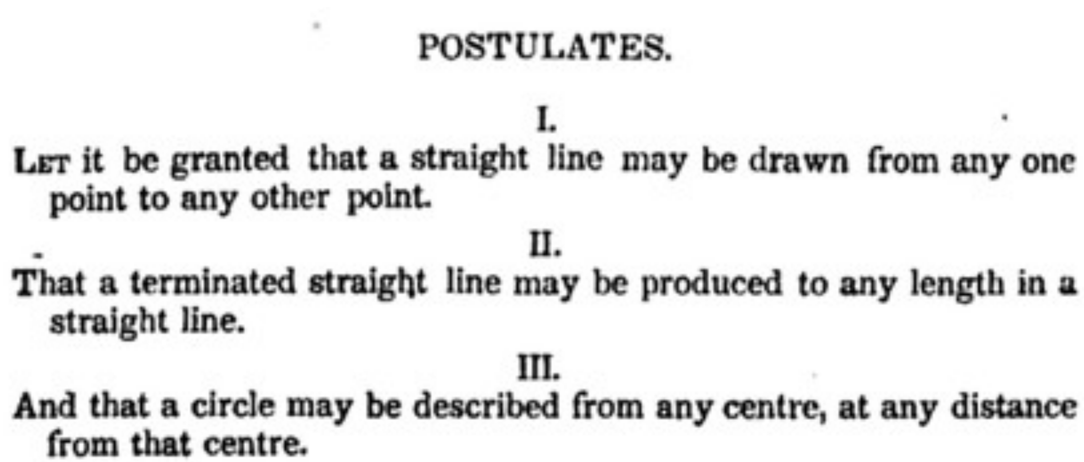
Analytic geometry (Descartes)



[image from wikipedia]

Synthetic vs Analytic

Synthetic geometry (Euclid)


POSTULATES.

I.

Let it be granted that a straight line may be drawn from any one point to any other point.

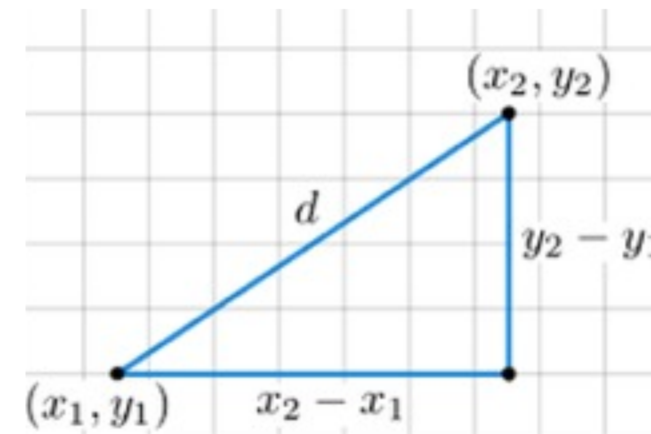
II.

That a terminated straight line may be produced to any length in a straight line.

III.

And that a circle may be described from any centre, at any distance from that centre.

Analytic geometry (Descartes)



Classical homotopy theory is analytic:

- a space is a set of points equipped with a topology
- a path is a continuous map [0,1] → X

[image from wikipedia]

Synthetic homotopy theory

homotopy theory	**type theory**
space	<type>
points	<element> : <type>
paths	<proof> : <elem$_1$> = <elem$_2$>
homotopies	<2-proof> : <proof$_1$> = <proof$_2$>
⋮	⋮

Synthetic homotopy theory

homotopy theory	**type theory**
space	<type>
points	<element> : <type>
paths	<proof> : <elem$_1$> = <elem$_2$>
homotopies	<2-proof> : <proof$_1$> = <proof$_2$>
⋮	⋮

Id(<elem1>,<elem2>)

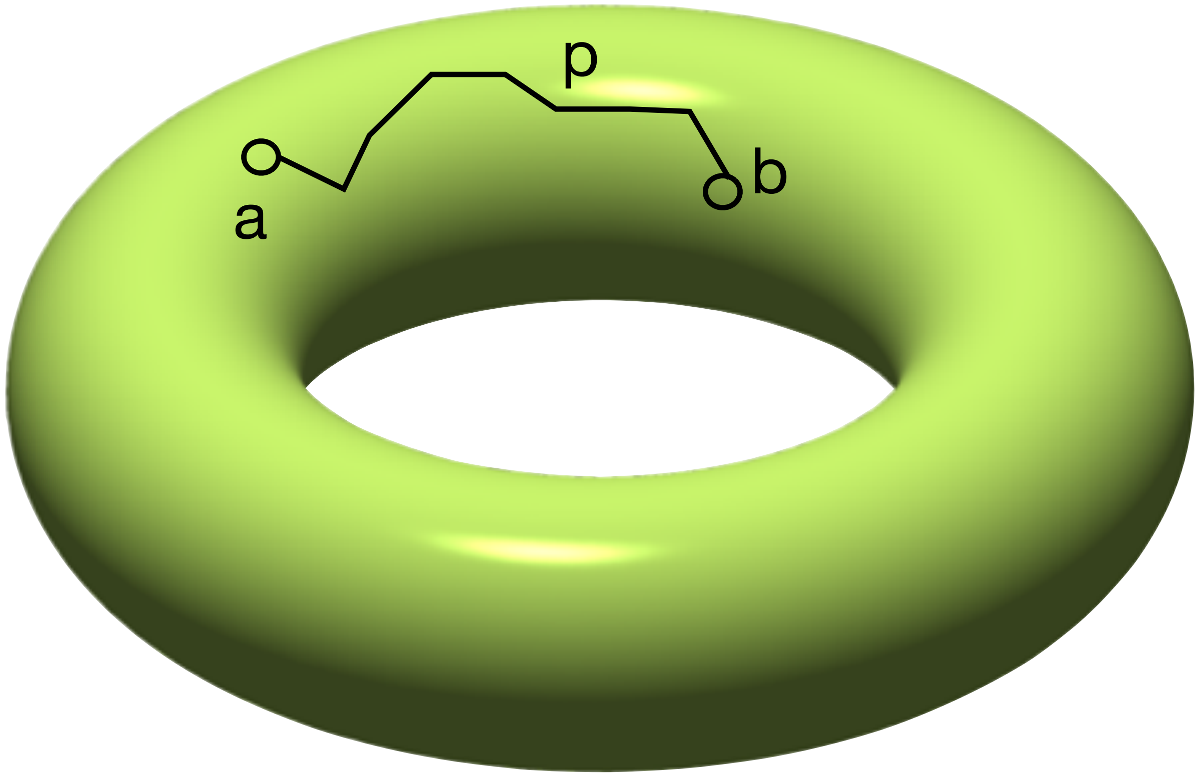
Synthetic homotopy theory

homotopy theory	**type theory**
space	<type>
points	<element> : <type>
paths	<proof> : <elem$_1$> = <elem$_2$>
homotopies	<2-proof> : <proof$_1$> = <proof$_2$>
⋮	⋮

*A path is **not** a map [0,1]→X; it is a basic notion*

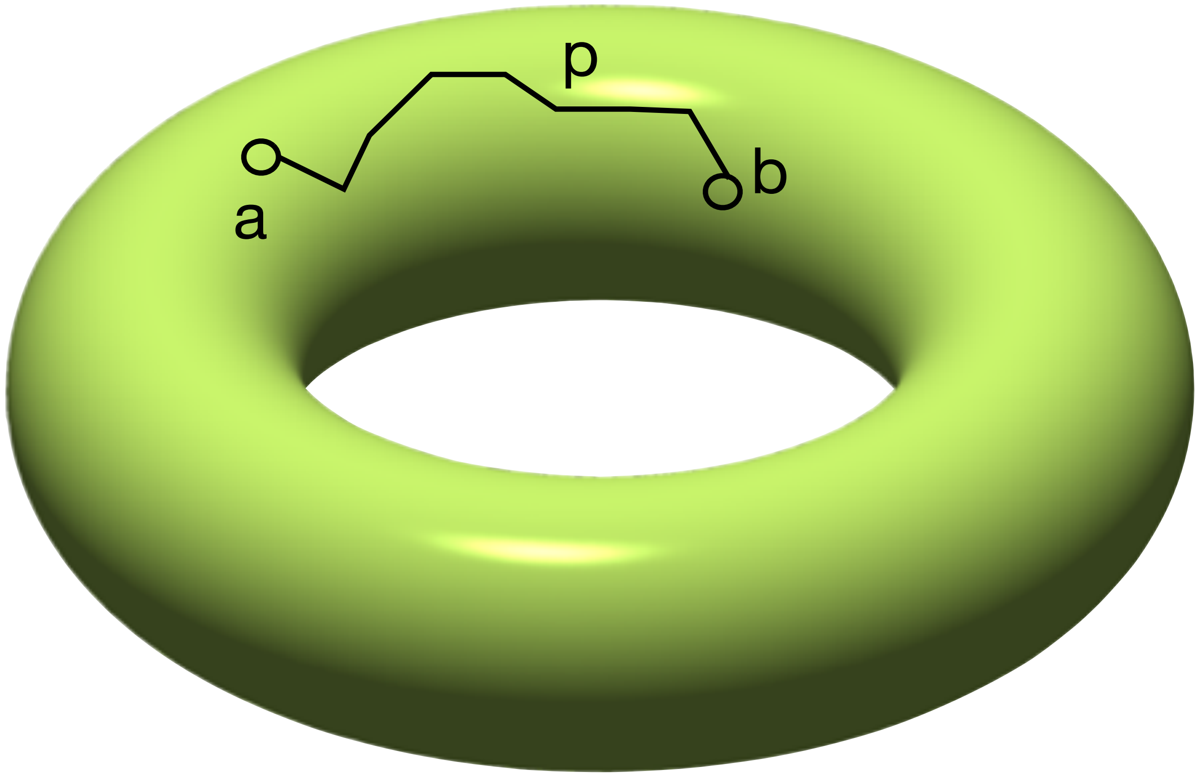
Id(<elem1>,<elem2>)

Spaces as types



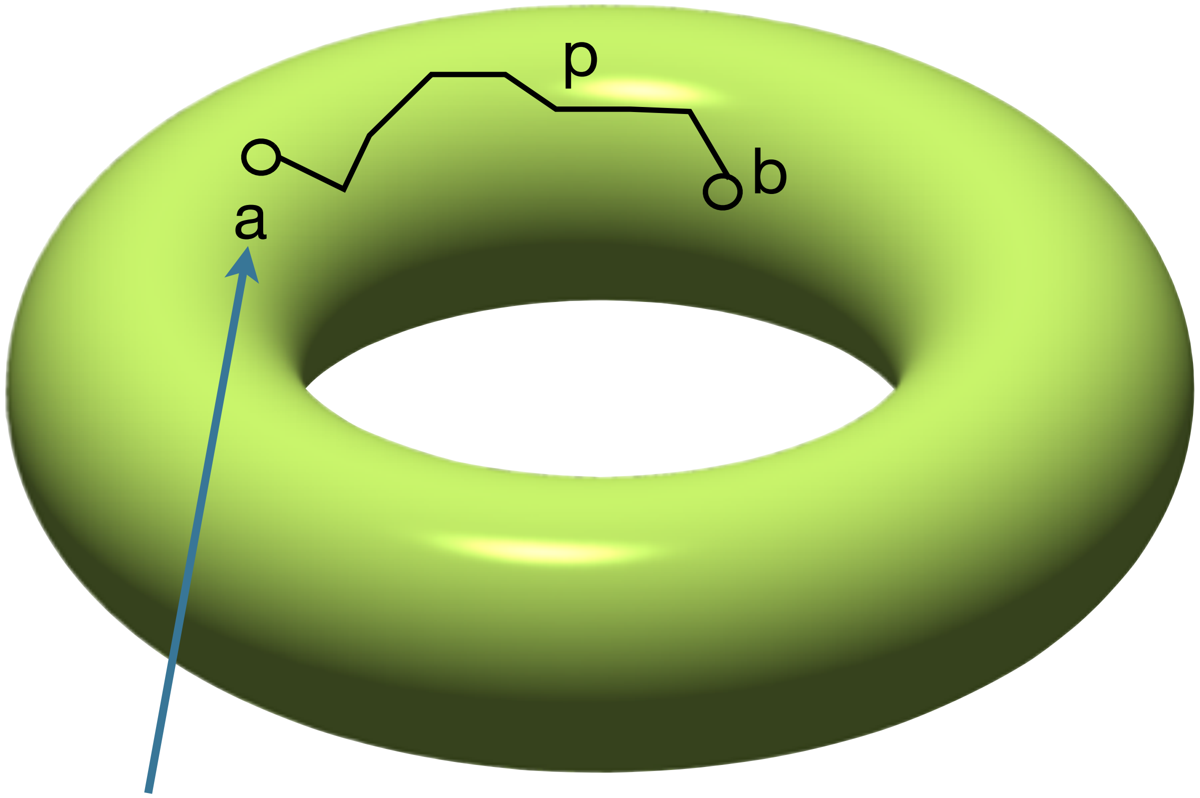
Spaces as types

a space is a type A



Spaces as types

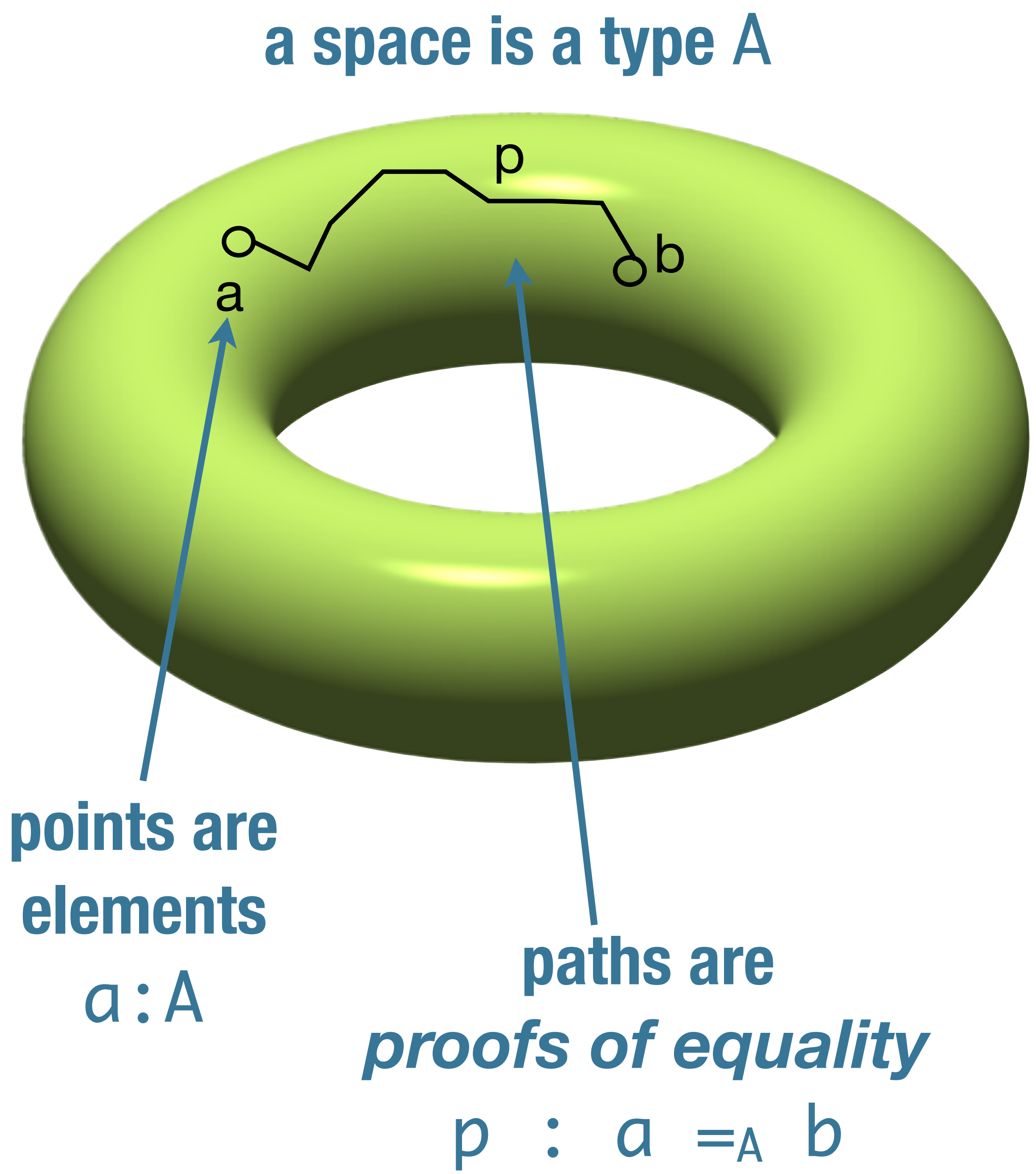
a space is a type A



points are elements

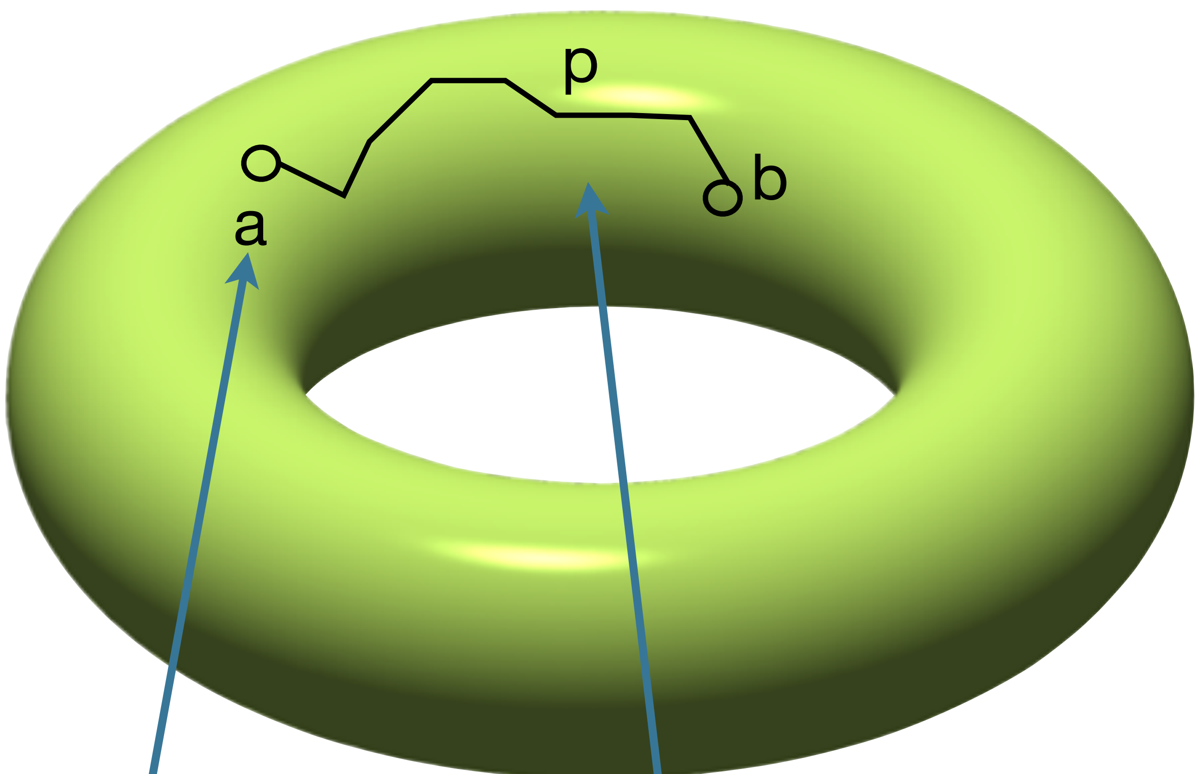
$a : A$

Spaces as types



Spaces as types

a space is a type A



points are elements

$a : A$

paths are

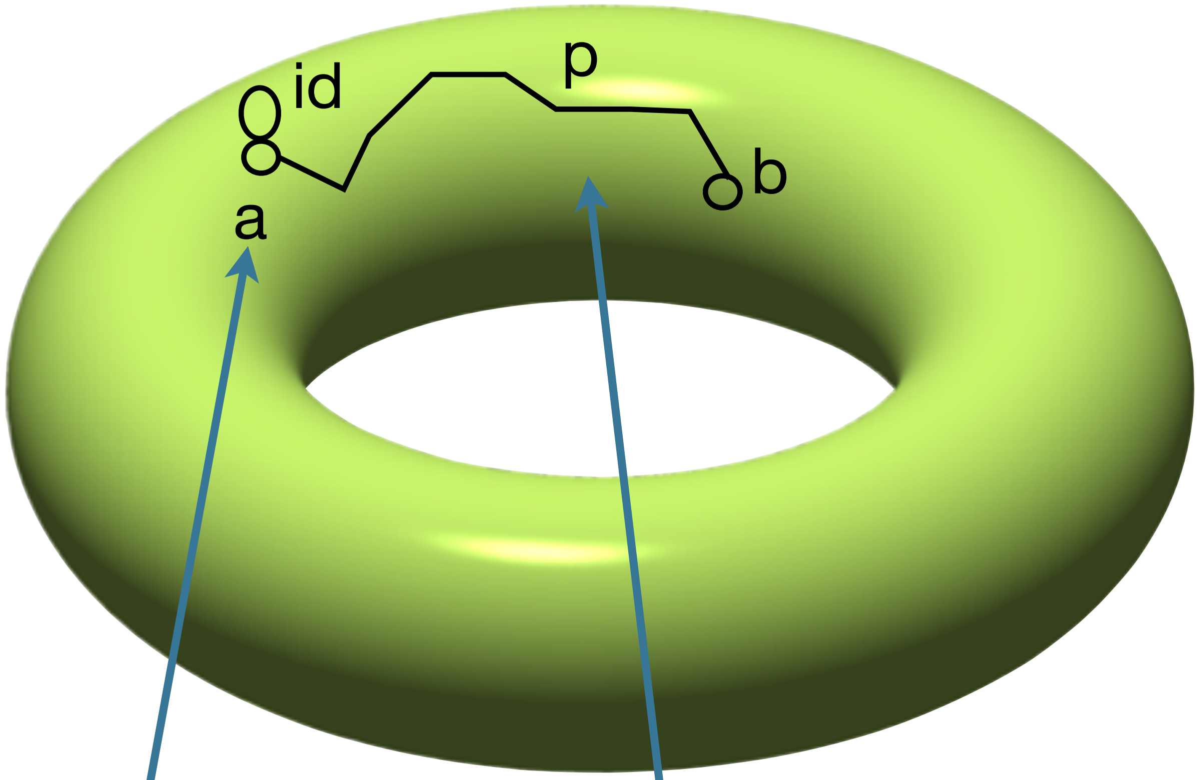
proofs of equality

$p \ : \ a =_A \ b$

path operations

Spaces as types

a space is a type A



points are elements

$a : A$

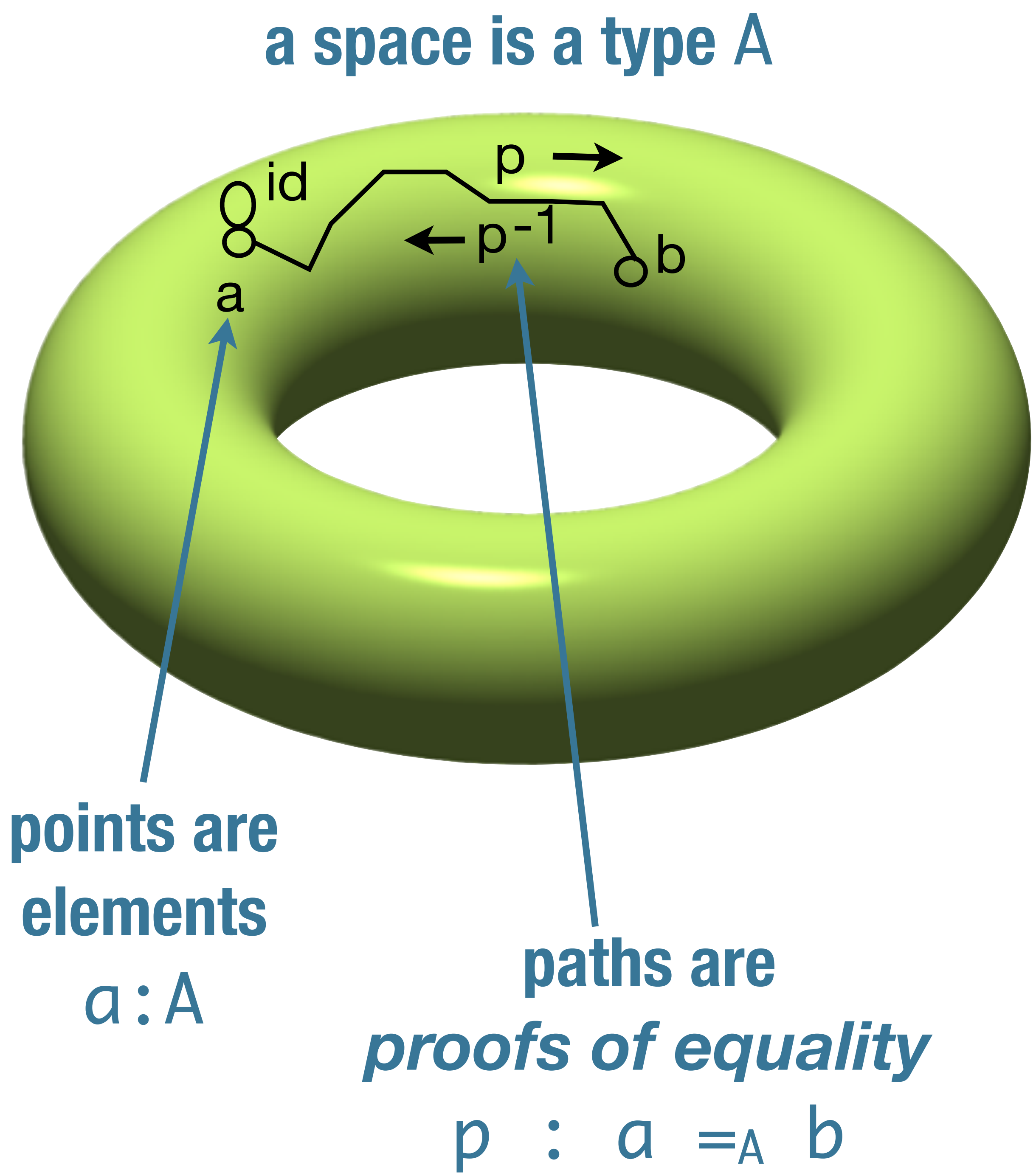
paths are *proofs of equality*

$p \ : \ a =_A b$

path operations

`id     : a = a (refl)`

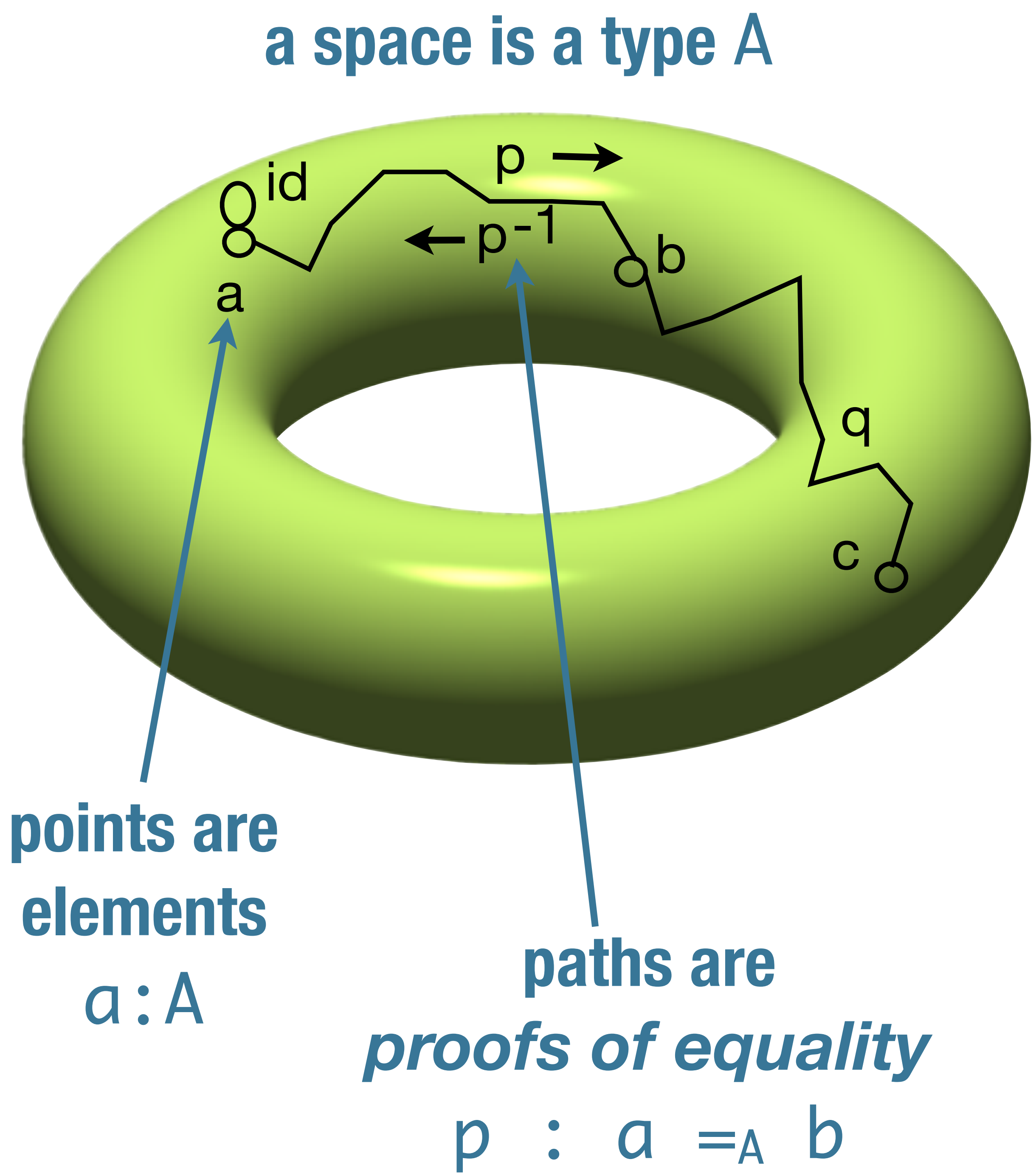
Spaces as types



path operations

$$\mathtt{id} : a = a \ (\mathrm{refl})$$
$$p^{-1} : b = a \ (\mathrm{sym})$$

Spaces as types



path operations

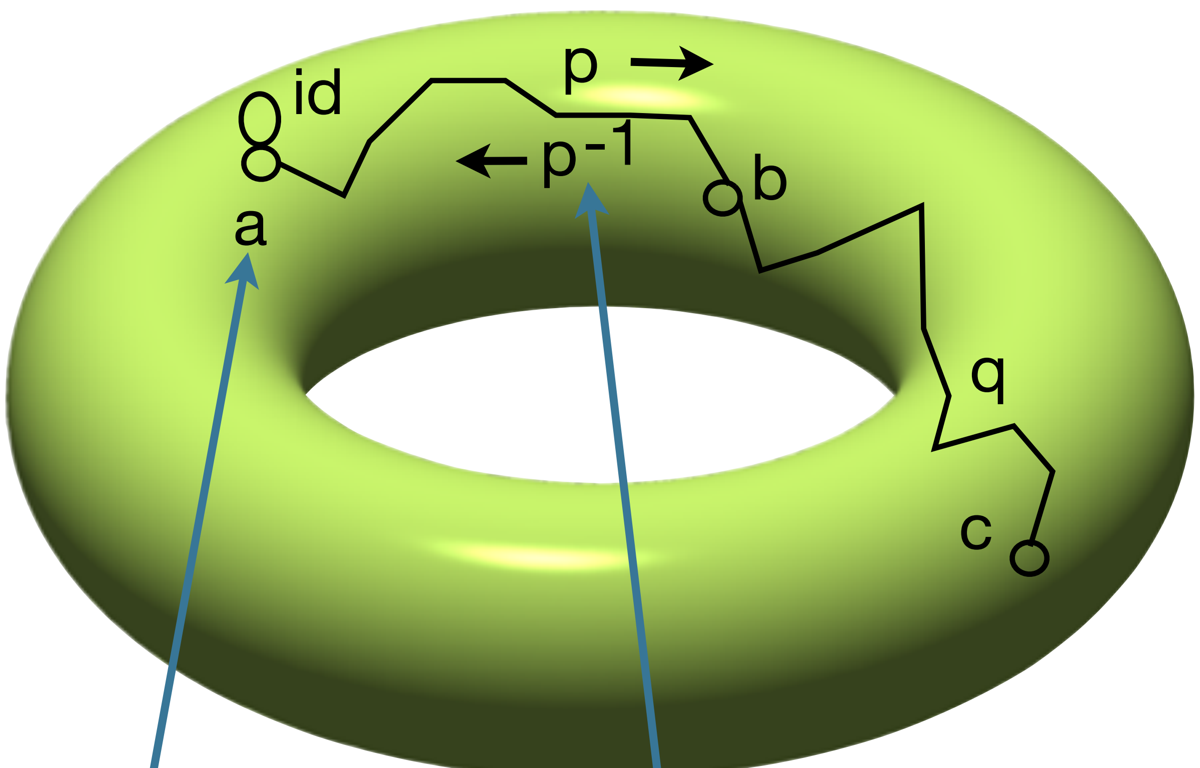
id : a = a (refl)

p^{-1} : b = a (sym)

q o p : a = c (trans)

Spaces as types

a space is a type A



points are elements

$a : A$

paths are ***proofs of equality***

$p : a =_A b$

path operations

$id : a = a$ (refl)

$p^{-1} : b = a$ (sym)

$q \circ p : a = c$ (trans)

homotopies

$id \circ p = p$

$p^{-1} \circ p = id$

$r \circ (q \circ p) = (r \circ q) \circ p$

Spaces as types

a space is a type A

points are elements
$a : A$

paths are *proofs of equality*
$p : a =_A b$

path operations

$\mathrm{id} : a = a$ (refl)

$p^{-1} : b = a$ (sym)

$q \circ p : a = c$ (trans)

homotopies

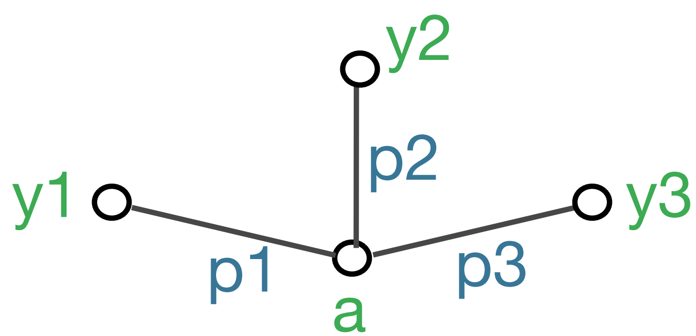
$\mathrm{id} \circ p = p$

$p^{-1} \circ p = \mathrm{id}$

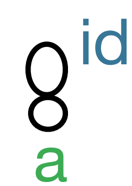
$r \circ (q \circ p) = (r \circ q) \circ p$

Path induction

Type of paths
from a to somewhere

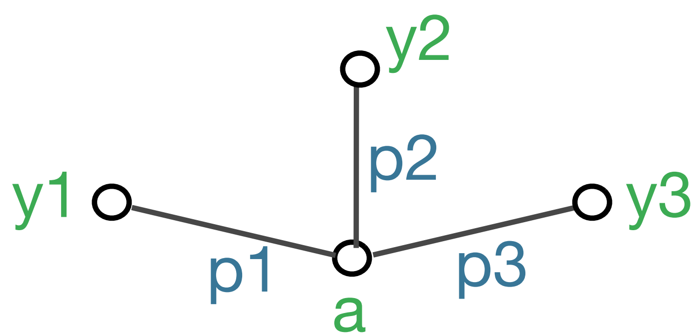


is inductively
generated by



Path induction

Type of paths from a to somewhere



is inductively generated by

Fix a type A with element $a:A$.

For a family of types $C(y:A,\ p:a=y)$,

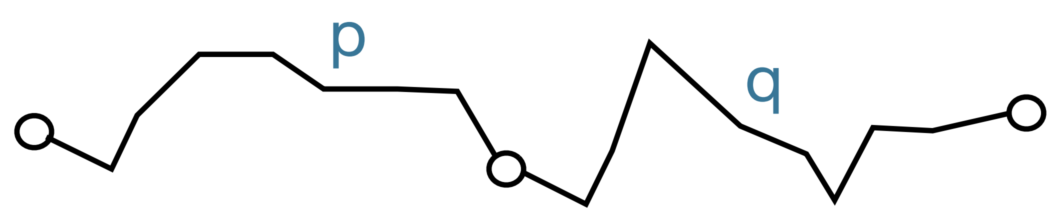
to give an element of

$$C(y,p) \text{ for all } y \text{ and } p:a=y,$$

suffices to give an element of

$$C(a,id)$$

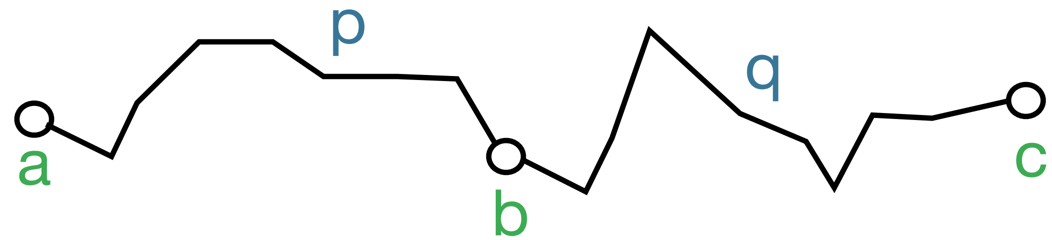
Composition, analytically



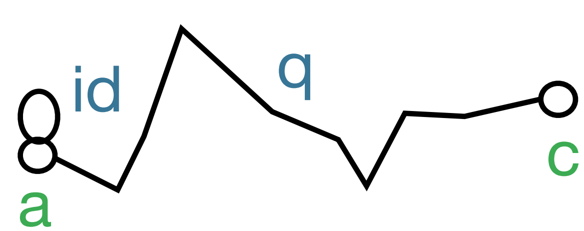
Given paths p and q `: [0,1] → X` where `p(1) = q(0)`
define **composition** by:

```
(q o p)(x) = p(2x)        if 0 ≤ x ≤ 1/2
           | q(2x - 1) if 1/2 ≤ x ≤ 1
```

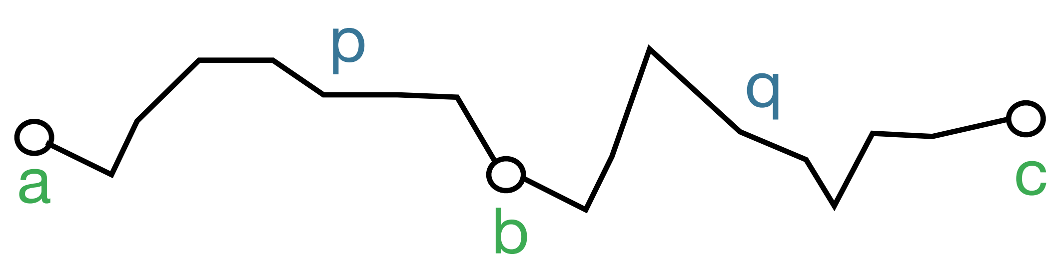
Composition, synthetically



Given paths `p : a=b` and `q : b=c`
define **composition** `(q o p)` by path induction:



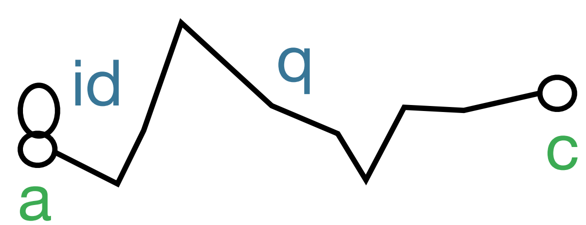
Composition, synthetically



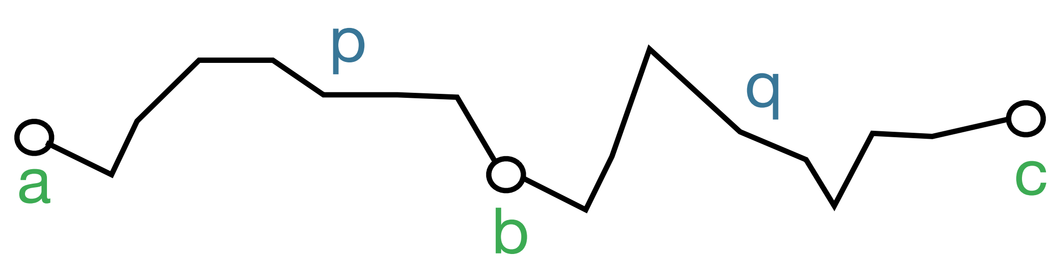
Given paths $\mathtt{p}:a=b$ and $\mathtt{q}:b=c$

define **composition** $(\mathtt{q} \circ \mathtt{p})$ by path induction:

- Suffices to consider case where b is a, and $\mathtt{p}$ is $\mathtt{id}$

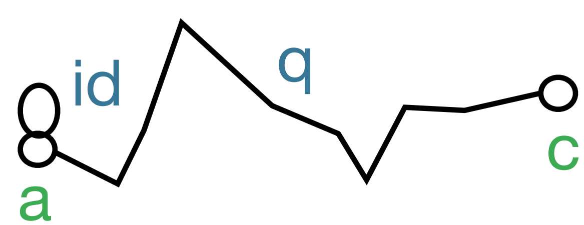


Composition, synthetically



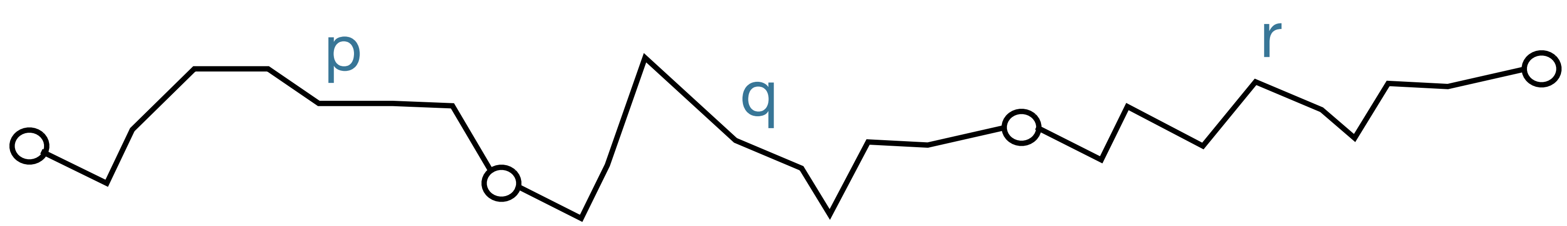
Given paths `p`:`a`=`b` and `q`:`b`=`c`
define **composition** `(q o p)` by path induction:

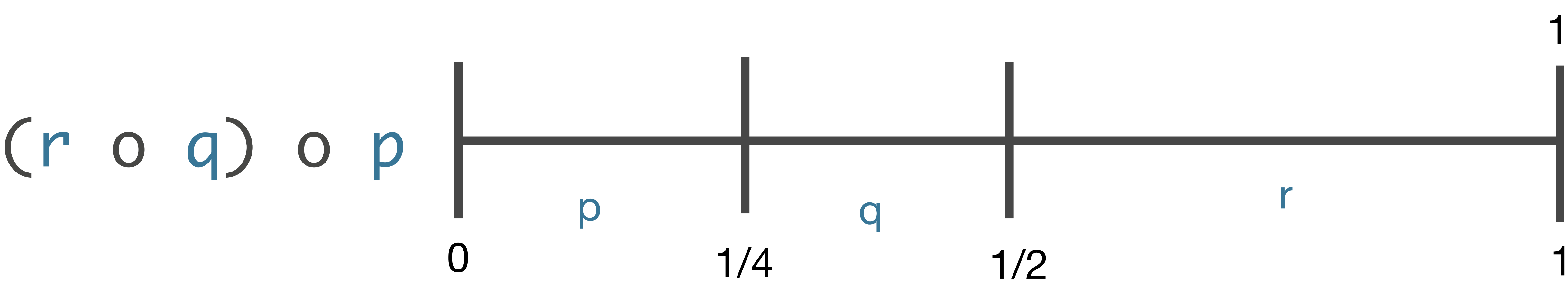
- Suffices to consider case where `b` is `a`, and `p` is `id`



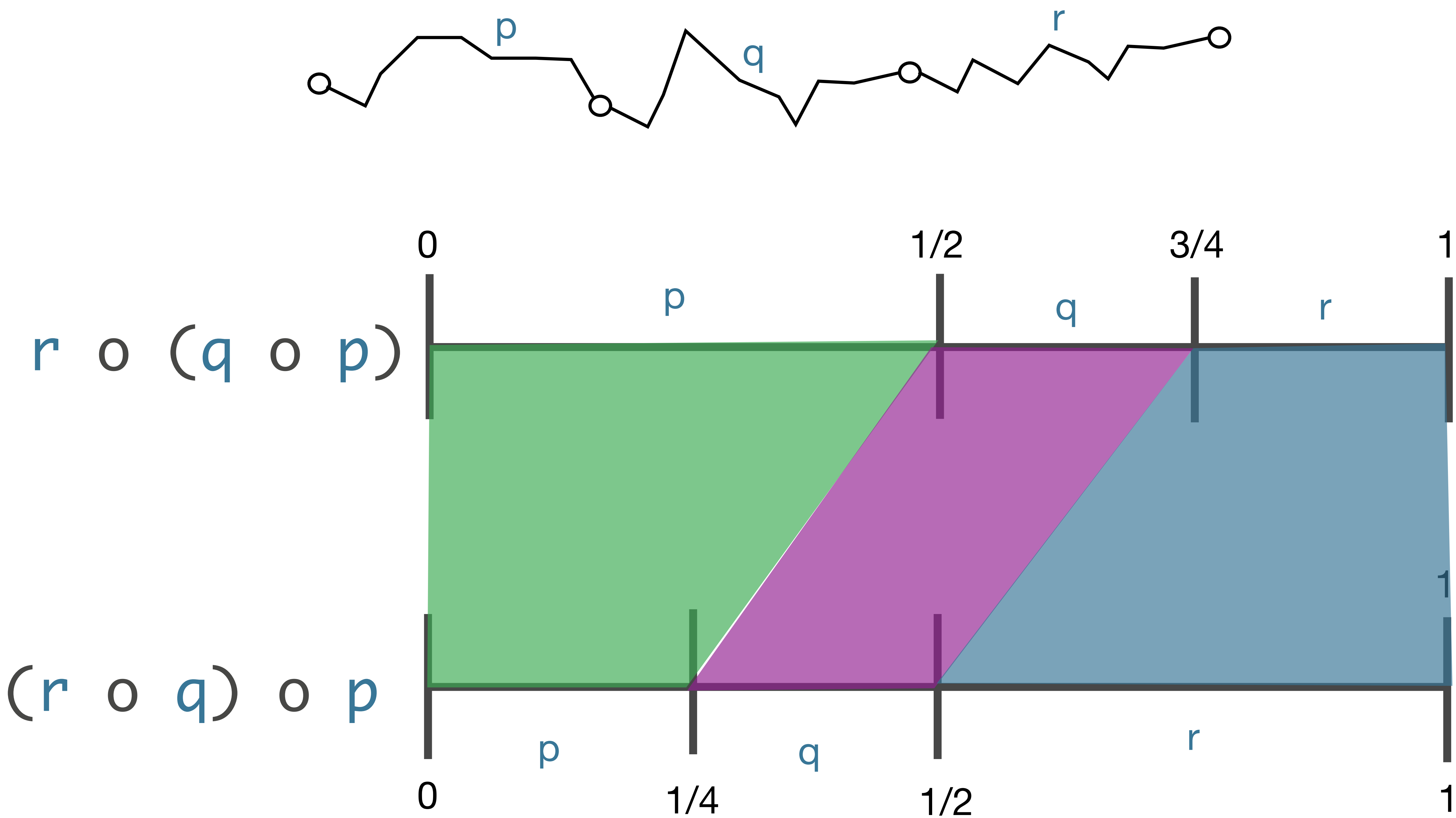
- In this case the composite is `q`

Associativity, analytically

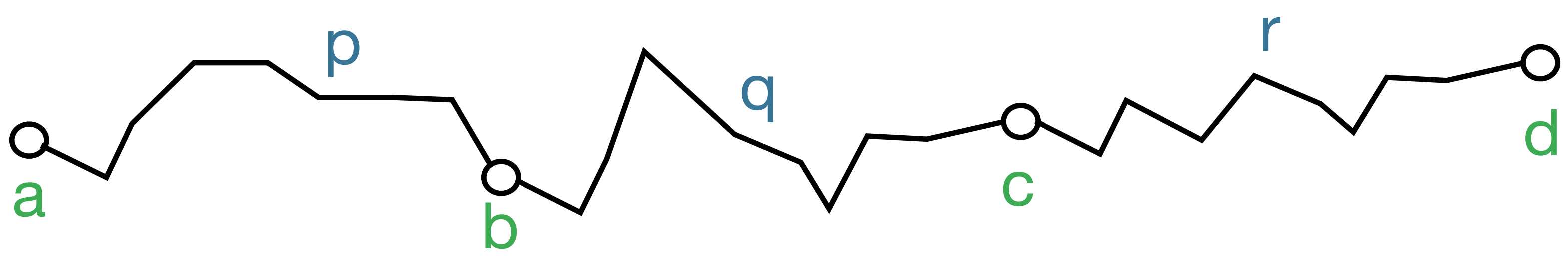




Associativity, analytically

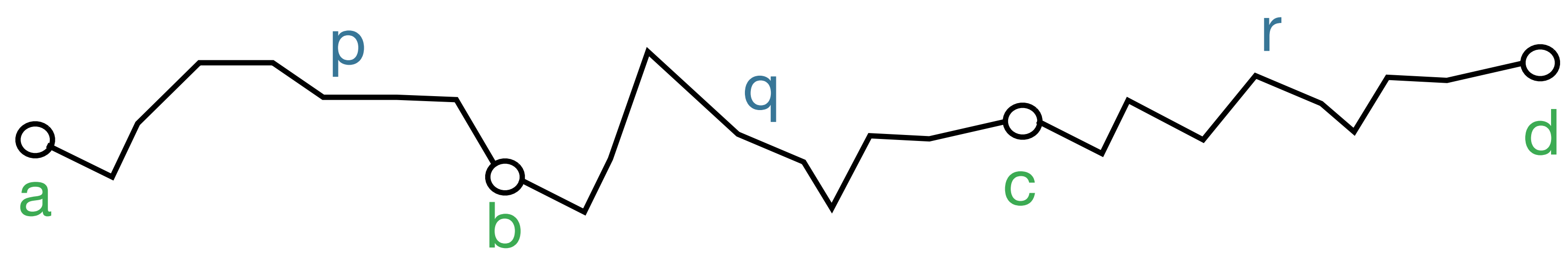


Associativity, synthetically



$$\forall\ a,b,c,d,\ p:a=b,\ q:b=c,\ r:c=d.$$
$$r \circ (q \circ p) = (r \circ q) \circ p$$

Associativity, synthetically



$$\forall\ a,b,c,d,\ p:a=b,\ q:b=c,\ r:c=d.$$
$$r \circ (q \circ p) = (r \circ q) \circ p$$

- By path induction, suffices to consider case where all points are a and all paths are `id`:
  `id o (id o id) = (id o id) o id`

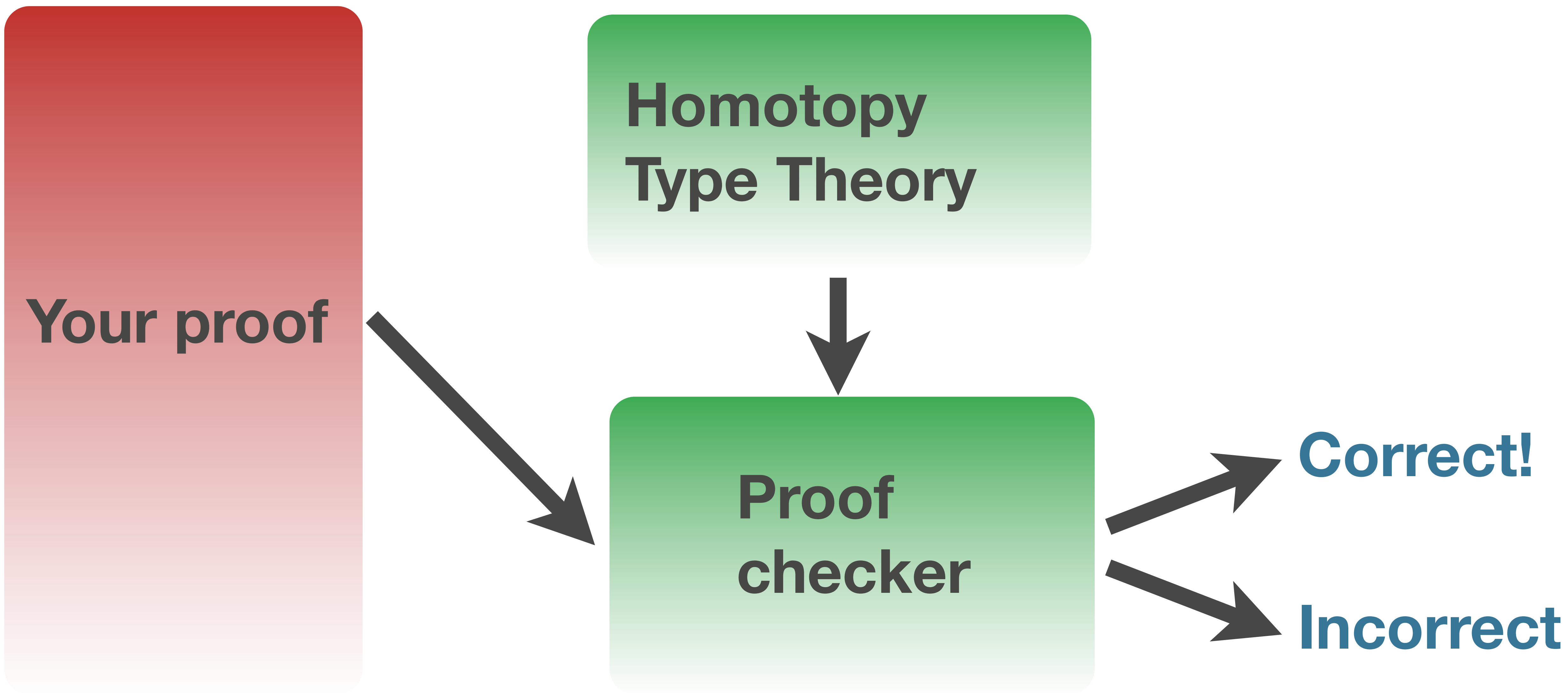
Associativity, synthetically

$$\forall\ a,b,c,d,\ p:a=b,\ q:b=c,\ r:c=d.$$
$$r \circ (q \circ p) = (r \circ q) \circ p$$

- By path induction, suffices to consider case where all points are a and all paths are `id`:
  $\mathrm{id} \circ (\mathrm{id} \circ \mathrm{id}) = (\mathrm{id} \circ \mathrm{id}) \circ \mathrm{id}$
- By definition of o, both sides equal `id`

Type theory is
a **logic** of
homotopy theory

Computer-checked proofs



Computer-checked proofs

```
_∘_  : {A : Type} {a b c : A}
     → Path b c → Path a b → Path a c
q ∘ id = q


∘-assoc  : {A : Type} {a b c d : A}
    (p : Path a b) (q : Path b c) (r : Path c d)
  → Path (r ∘ (q ∘ p)) ((r ∘ q) ∘ p)
∘-assoc id id id = id
```

We can do
computer-checked proofs
in **synthetic** homotopy theory

We can do
computer-checked proofs
in **synthetic** homotopy theory

- Proofs are constructive*: can run them

We can do **computer-checked proofs** in **synthetic** homotopy theory

- Proofs are constructive*: can run them
- Results apply in a variety of settings, from simplicial sets (hence topological spaces) to Quillen model categories and ∞-topoi*

We can do **computer-checked proofs** in **synthetic** homotopy theory

- Proofs are constructive*: can run them
- Results apply in a variety of settings, from simplicial sets (hence topological spaces) to Quillen model categories and ∞-topoi*
- New type-theoretic proofs/methods

We can do **computer-checked proofs** in **synthetic** homotopy theory

- Proofs are constructive*: can run them
- Results apply in a variety of settings, from simplicial sets (hence topological spaces) to Quillen model categories and ∞-topoi*
- New type-theoretic proofs/methods

*work in progress

Homotopy in HoTT

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

$\pi_{k<n}(S^n) = 0$

Hopf fibration

$\pi_2(S^2) = \mathbb{Z}$

$\pi_3(S^2) = \mathbb{Z}$

James Construction

$\pi_4(S^3) = \mathbb{Z}_?$

Freudenthal

$\pi_n(S^n) = \mathbb{Z}$

K(G,n)

Cohomology axioms

Blakers-Massey

Van Kampen

Covering spaces

Whitehead for n-types

[Brunerie, Finster, Hou, Licata, Lumsdaine, Shulman]

Homotopy in HoTT

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

$\pi_{k<n}(S^n) = 0$

Hopf fibration

$\pi_2(S^2) = \mathbb{Z}$

$\pi_3(S^2) = \mathbb{Z}$

James Construction

$\pi_4(S^3) = \mathbb{Z}_?$

Freudenthal

$\pi_n(S^n) = \mathbb{Z}$

K(G,n)

Cohomology axioms

Blakers-Massey

Van Kampen

Covering spaces

Whitehead for n-types

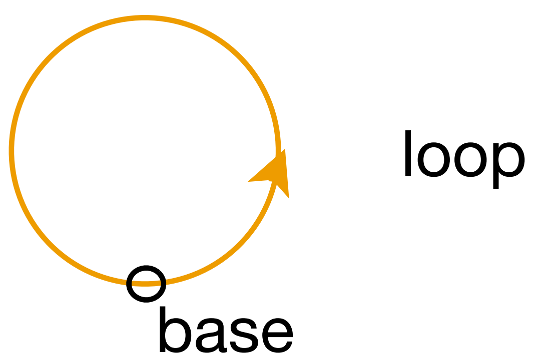
[Brunerie, Finster, Hou, Licata, Lumsdaine, Shulman]

Example:

The Fundamental Group of the Circle

The Circle

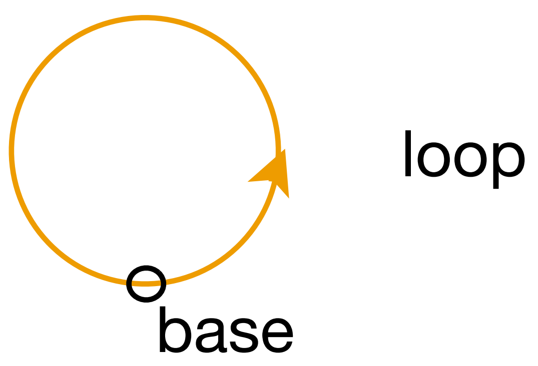
Circle S^1 is ***inductively generated*** by



The Circle

Circle S^1 is ***inductively generated*** by

```
base : S¹
loop : base = base
```

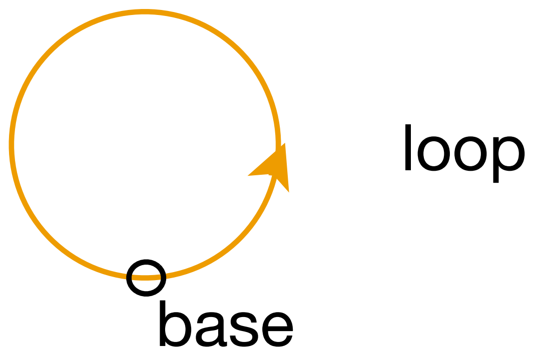


The Circle

Circle S^1 is ***inductively generated*** by

point `base : S¹`

`loop : base = base`

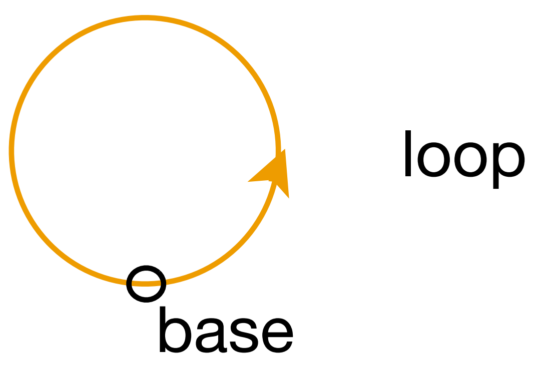


The Circle

Circle S^1 is ***inductively generated*** by

point `base : S¹`

path `loop : base = base`

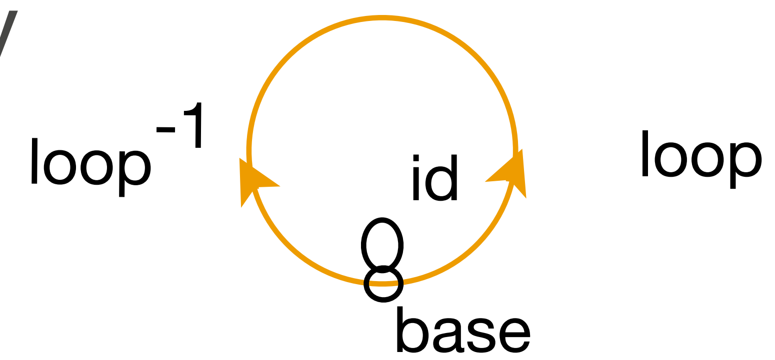


The Circle

Circle S^1 is ***inductively generated*** by

point `base : S¹`

path `loop : base = base`



Free type: equipped with structure by path induction

`id` `inv : loop o loop⁻¹ = id`

`loop⁻¹` `...`

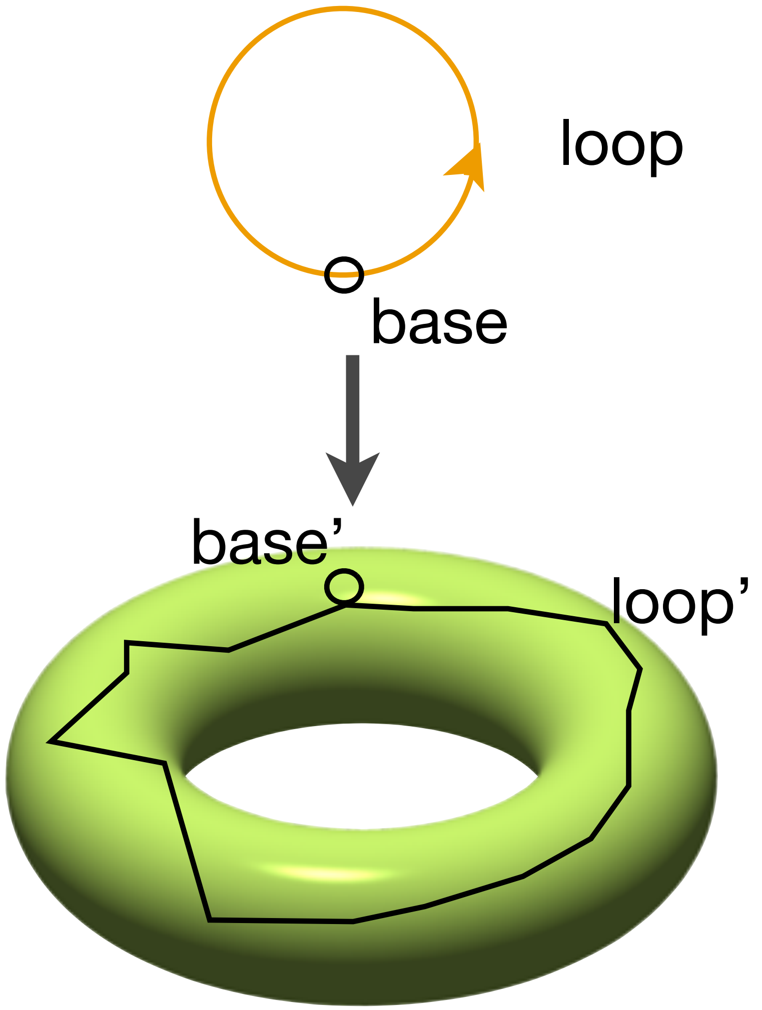
`loop o loop`

The Circle

Circle recursion:

function $S^1 \to X$ determined by

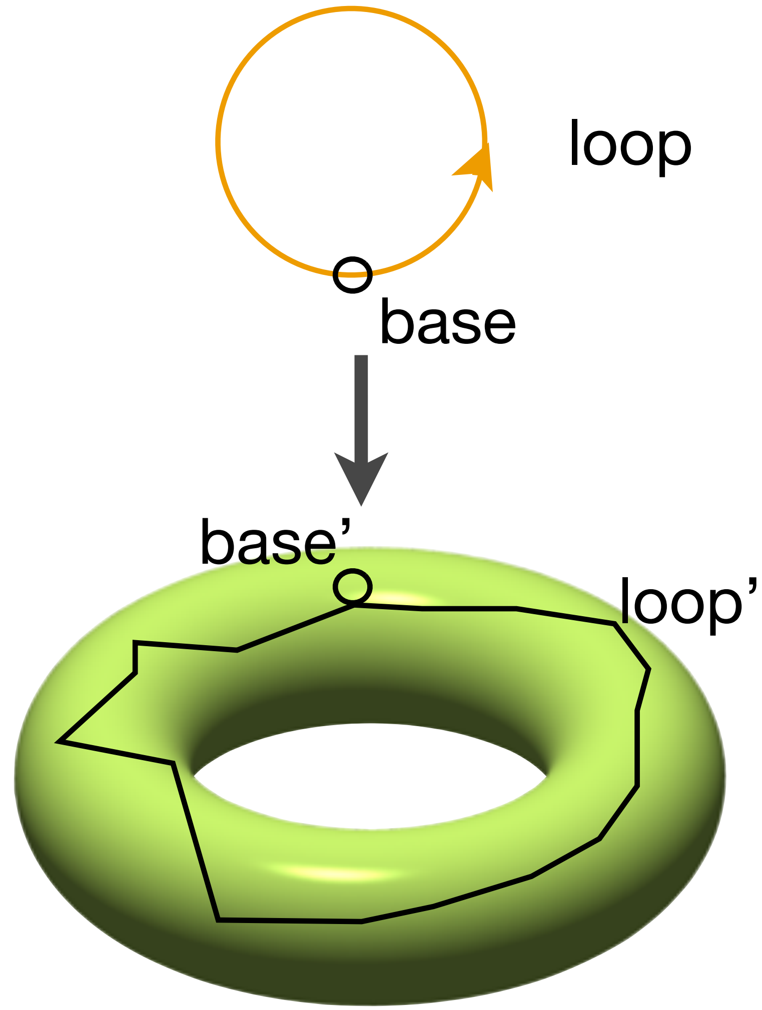
```
base' : X
loop' : base' = base'
```



The Circle

Circle recursion:
function $S^1 \to X$ determined by

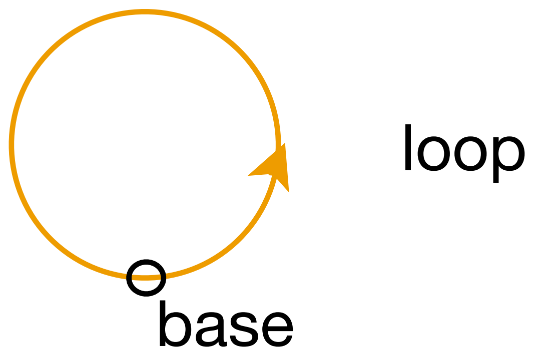
```
base' : X
loop' : base' = base'
```



Circle induction: To prove a predicate P for all points on the circle, suffices to prove P(`base`), continuously in the loop

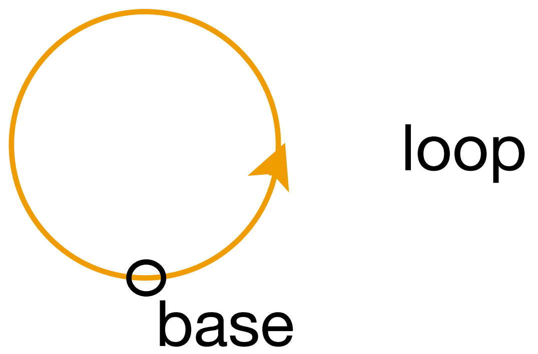
Fundamental group of circle

How many different loops are there on the circle, up to homotopy?



Fundamental group of circle

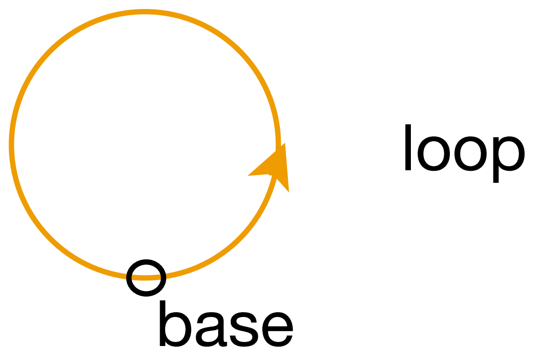
How many different loops are there on the circle, up to homotopy?



`id`

Fundamental group of circle

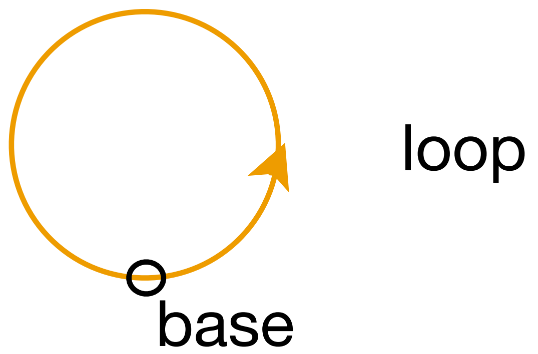
How many different loops are there on the circle, up to homotopy?




```
id
loop
```

Fundamental group of circle

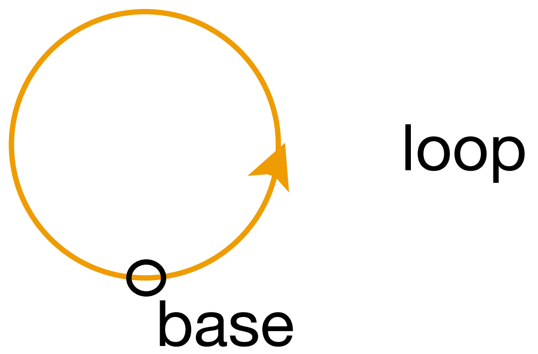
How many different loops are there on the circle, up to homotopy?




```
id
loop
loop⁻¹
```

Fundamental group of circle

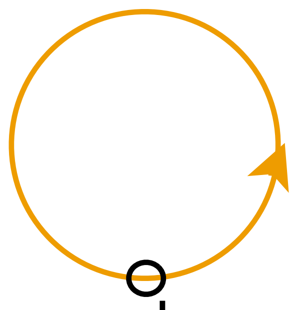
How many different loops are there on the circle, up to homotopy?




```
id
loop
loop⁻¹
loop o loop
```

Fundamental group of circle

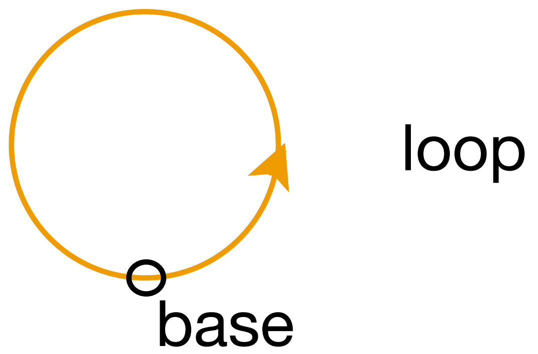
How many different loops are there on the circle, up to homotopy?




```
id
loop
loop⁻¹
loop o loop
loop⁻¹ o loop⁻¹
```

Fundamental group of circle

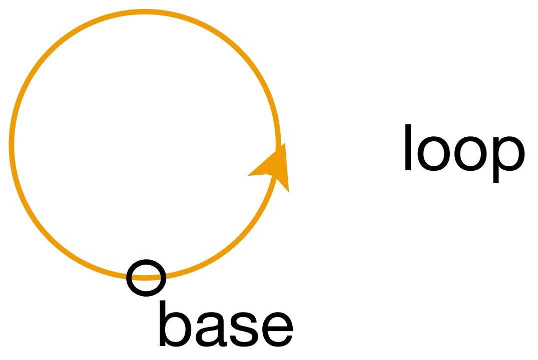
How many different loops are there on the circle, up to homotopy?




```
id
loop
loop⁻¹
loop o loop
loop⁻¹ o loop⁻¹
loop o loop⁻¹
```

Fundamental group of circle

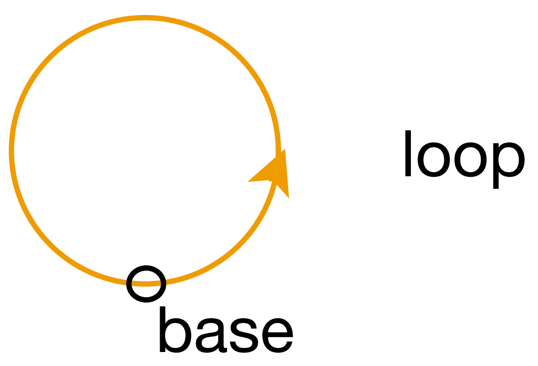
How many different loops are there on the circle, up to homotopy?




```
id
loop
loop⁻¹
loop o loop
loop⁻¹ o loop⁻¹
loop o loop⁻¹  = id
```

Fundamental group of circle

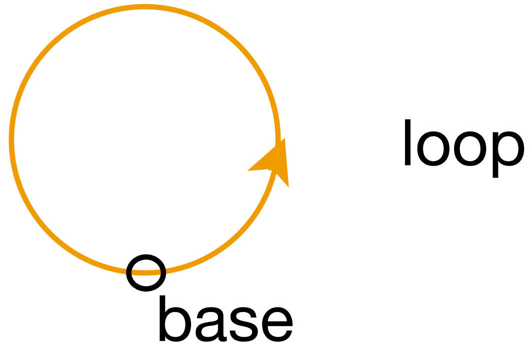
How many different loops are there on the circle, up to homotopy?




```
id                      0
loop
loop⁻¹
loop o loop
loop⁻¹ o loop⁻¹
loop o loop⁻¹   = id
```

Fundamental group of circle

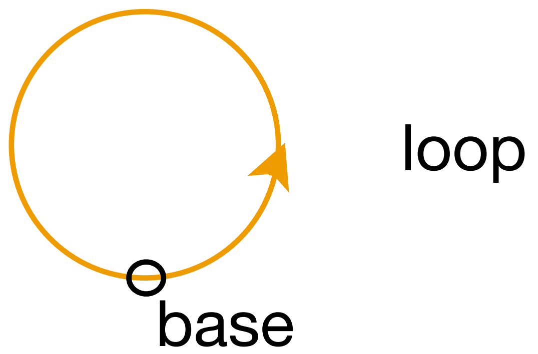
How many different loops are there on the circle, up to homotopy?



`id`	`0`
`loop`	`1`
`loop`$^{-1}$	
`loop o loop`	
`loop`$^{-1}$ `o loop`$^{-1}$	
`loop o loop`$^{-1}$ `= id`	

Fundamental group of circle

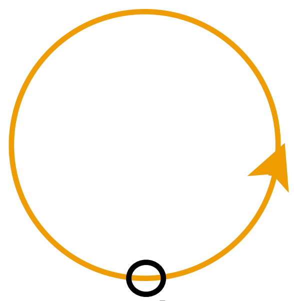
How many different loops are there on the circle, up to homotopy?



`id`	`0`
`loop`	`1`
`loop`$^{-1}$	`-1`
`loop o loop`	
`loop`$^{-1}$ `o loop`$^{-1}$	
`loop o loop`$^{-1}$ `= id`	

Fundamental group of circle

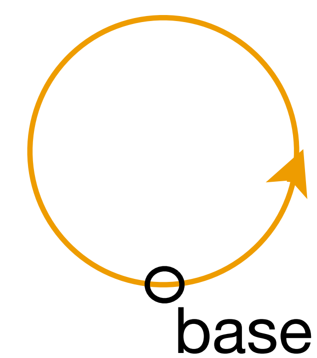
How many different loops are there on the circle, up to homotopy?



`id`	`0`
`loop`	`1`
`loop`$^{-1}$	`-1`
`loop o loop`	`2`
`loop`$^{-1}$ `o loop`$^{-1}$	
`loop o loop`$^{-1}$ `= id`	

Fundamental group of circle

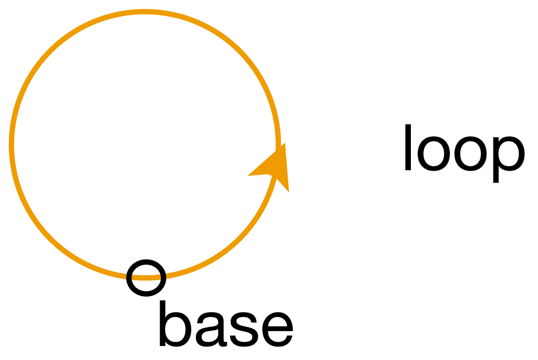
How many different loops are there on the circle, up to homotopy?




```
id                      0
loop                    1
loop⁻¹                  -1
loop o loop             2
loop⁻¹ o loop⁻¹         -2
loop o loop⁻¹   = id
```

Fundamental group of circle

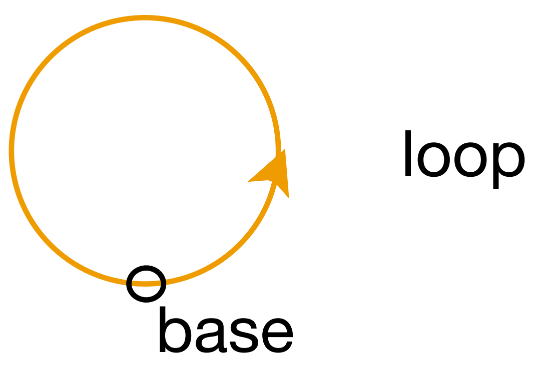
How many different loops are there on the circle, up to homotopy?



`id`	`0`
`loop`	`1`
`loop`$^{-1}$	`-1`
`loop o loop`	`2`
`loop`$^{-1}$ `o loop`$^{-1}$	`-2`
`loop o loop`$^{-1}$ `= id`	`0`

Fundamental group of circle

How many different loops are there on the circle, up to homotopy?



`id`	`0`
`loop`	`1`
`loop`$^{-1}$	`-1`
`loop o loop`	`2`
`loop`$^{-1}$ `o loop`$^{-1}$	`-2`
`loop o loop`$^{-1}$ `= id`	`0`

integers are "codes" for paths on the circle

Fundamental group of circle

Definition. $\Omega(S^1)$ is the **type** of loops at base
i.e. the type $(\mathit{base} = \mathit{base})$

Fundamental group of circle

Definition. $\Omega(S^1)$ is the **type** of loops at base
i.e. the type `(base = base)`

Theorem. $\Omega(S^1)$ is equivalent to $\mathbb{Z}$,
by a map that sends o to +

Fundamental group of circle

Definition. $\Omega(S^1)$ is the **type** of loops at base
i.e. the type `(base = base)`

Theorem. $\Omega(S^1)$ is equivalent to $\mathbb{Z}$,
by a map that sends o to +

Corollary: Fundamental group
of the circle is isomorphic to $\mathbb{Z}$

Fundamental group of circle

Definition. $\Omega(S^1)$ is the **type** of loops at base
i.e. the type `(base = base)`

Theorem. $\Omega(S^1)$ is equivalent to $\mathbb{Z}$,
by a map that sends o to +

Corollary: Fundamental group
of the circle is isomorphic to $\mathbb{Z}$

0-truncation (set of connected components) of $\Omega(S^1)$

Fundamental group of circle

Theorem. $\Omega(S^1)$ is equivalent to $\mathbb{Z}$

Proof: two mutually inverse functions

```
wind  : Ω(S¹) → ℤ

loop⁻ : ℤ → Ω(S¹)
```

Fundamental group of circle

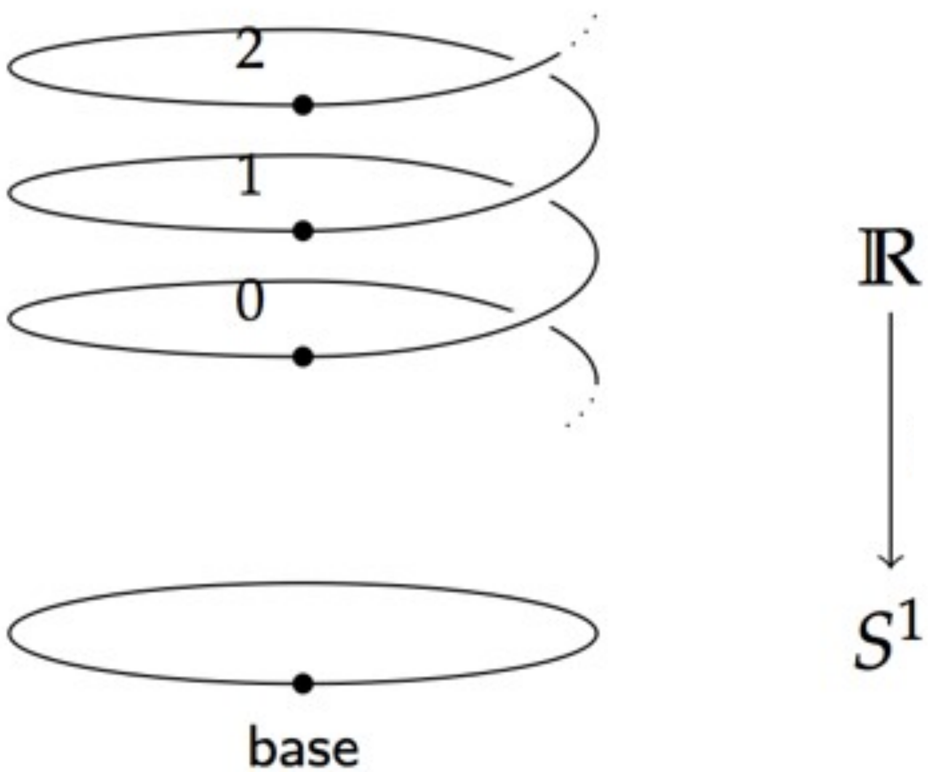
Theorem. $\Omega(S^1)$ is equivalent to $\mathbb{Z}$

Proof: two mutually inverse functions

```
wind   : Ω(S¹) → ℤ

loop⁻  : ℤ → Ω(S¹)
loop⁰   = id
loop⁺ⁿ  = loop o loop o … loop        (n times)
loop⁻ⁿ  = loop⁻¹ o loop⁻¹ o … loop⁻¹  (n times)
```

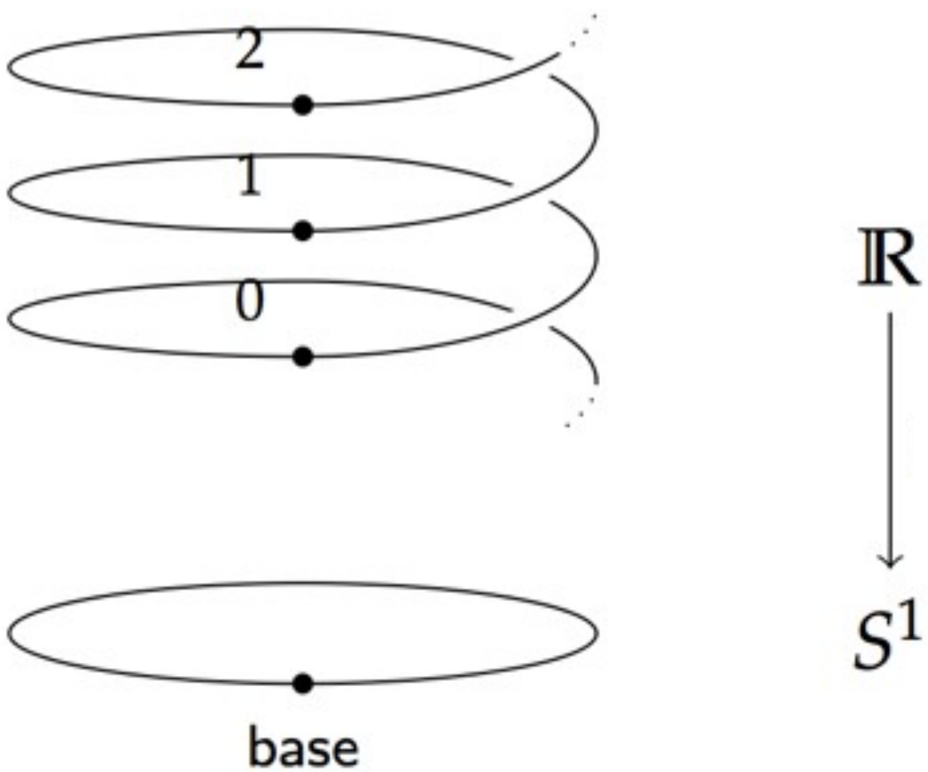
Universal Cover



$\texttt{wind} : \Omega(S^1) \to \mathbb{Z}$

defined by **lifting** a loop to the cover, and giving the other endpoint of 0

Universal Cover

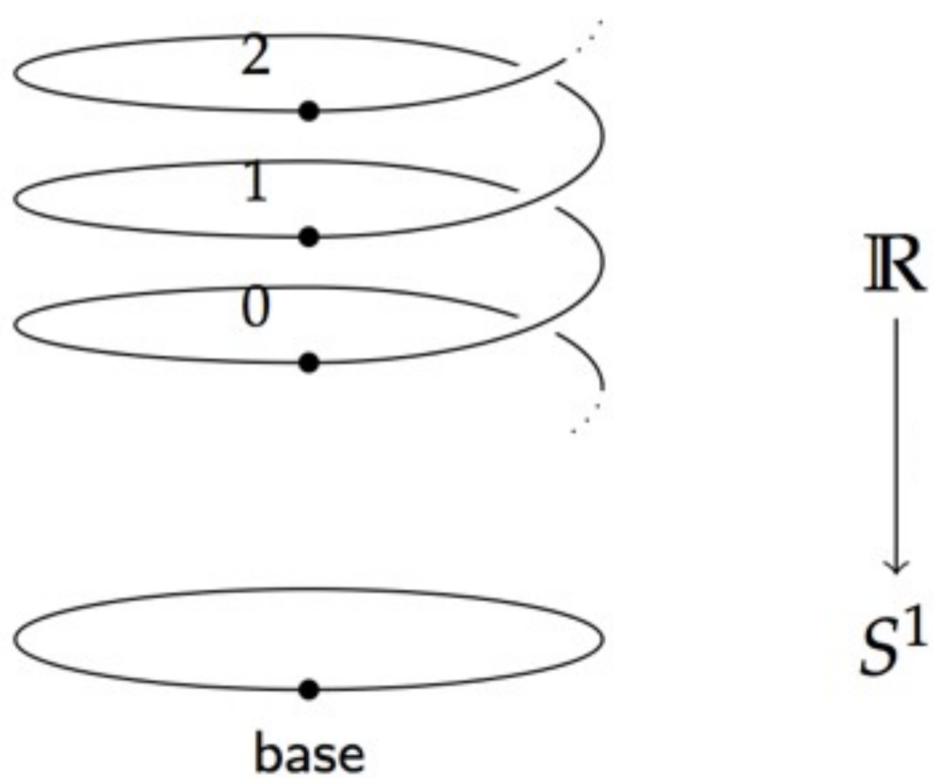


lifting is functorial

`wind : Ω(S¹) → ℤ`

defined by **lifting** a loop to the cover, and giving the other endpoint of 0

Universal Cover



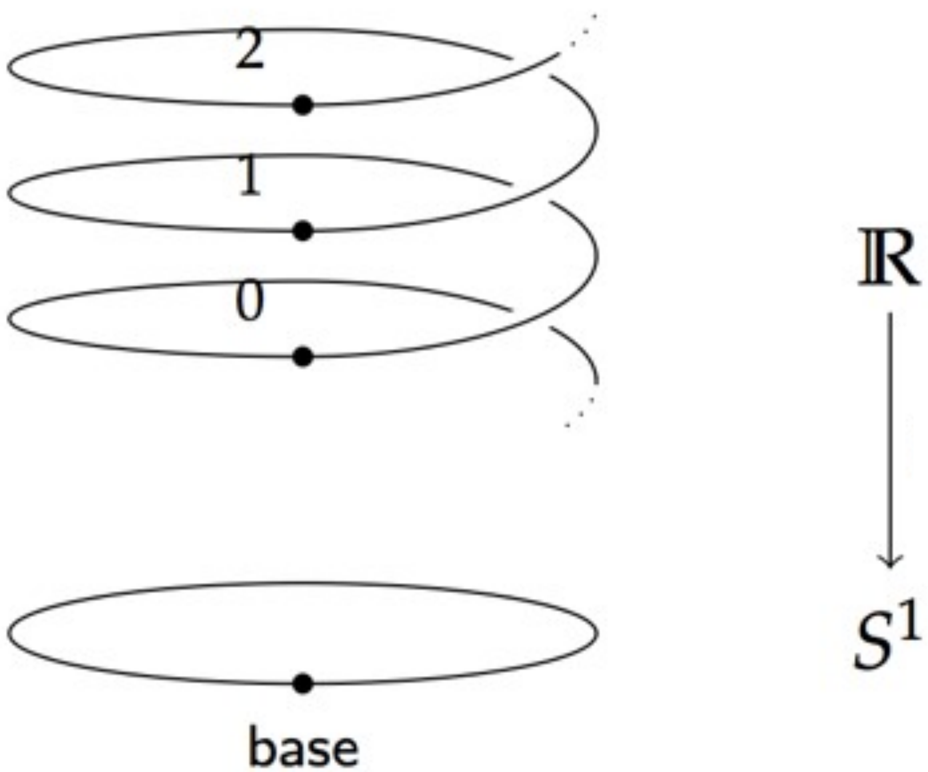
`wind` : $\Omega(S^1) \to \mathbb{Z}$

defined by **lifting** a loop to the cover, and giving the other endpoint of 0

lifting is functorial

lifting `loop` adds 1

Universal Cover



$\mathtt{wind} : \Omega(S^1) \to \mathbb{Z}$

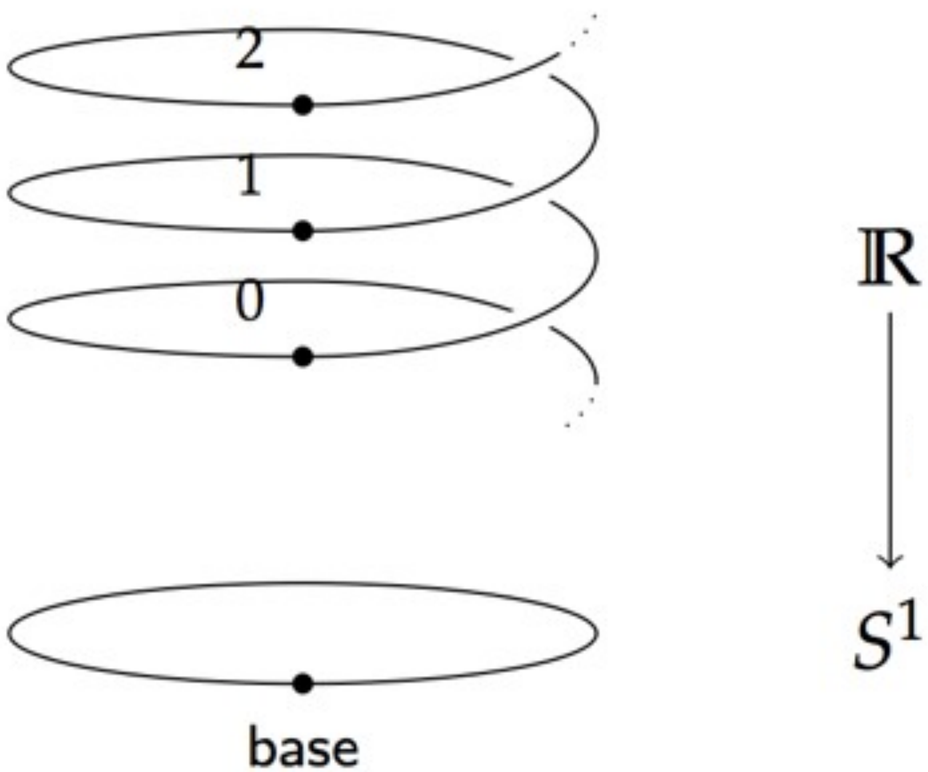
defined by **lifting** a loop to the cover, and giving the other endpoint of 0

lifting is functorial

lifting `loop` adds 1

lifting $\mathtt{loop}^{-1}$ subtracts 1

Universal Cover



`wind : Ω(S¹) → ℤ`

defined by **lifting** a loop to the cover, and giving the other endpoint of 0

lifting is functorial

lifting `loop` adds 1

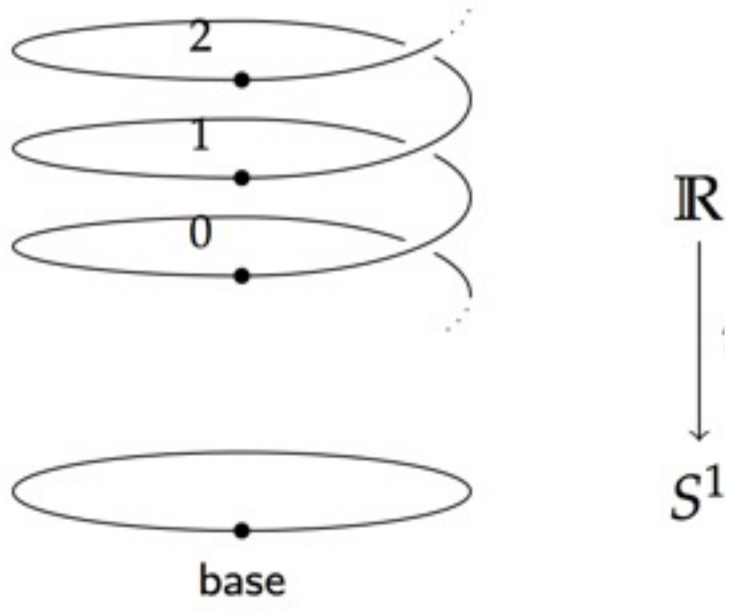
lifting `loop`$^{-1}$ subtracts 1

Example:

$$\mathtt{wind}(\mathtt{loop} \circ \mathtt{loop}^{-1})$$
$$= 0 + 1 - 1$$
$$= 0$$

Fibration = Family of types

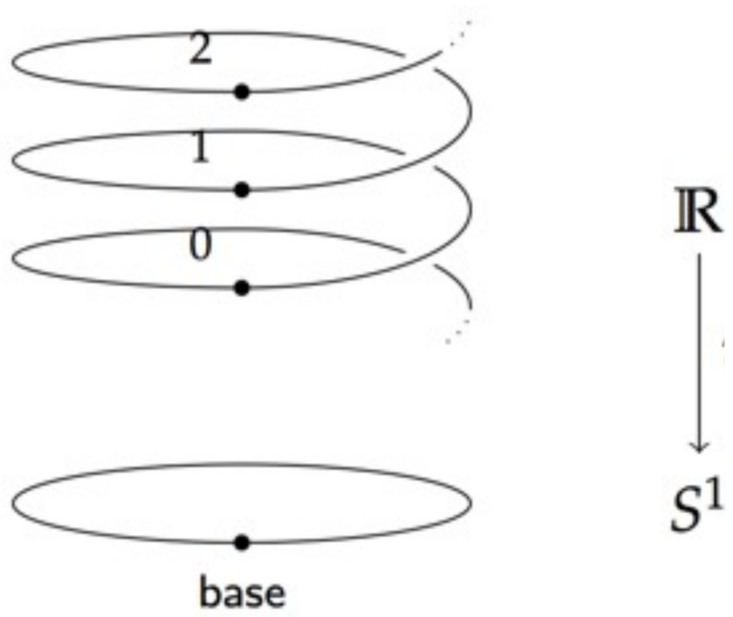
Fibration (classically):
map $p: E \to B$ such that
any path from $p(e)$ to y
lifts to a path in E from e
to some point in $p^{-1}(y)$



Fibration = Family of types

Fibration (classically):

map $p: E \to B$ such that any path from $p(e)$ to y lifts to a path in E from e to some point in $p^{-1}(y)$



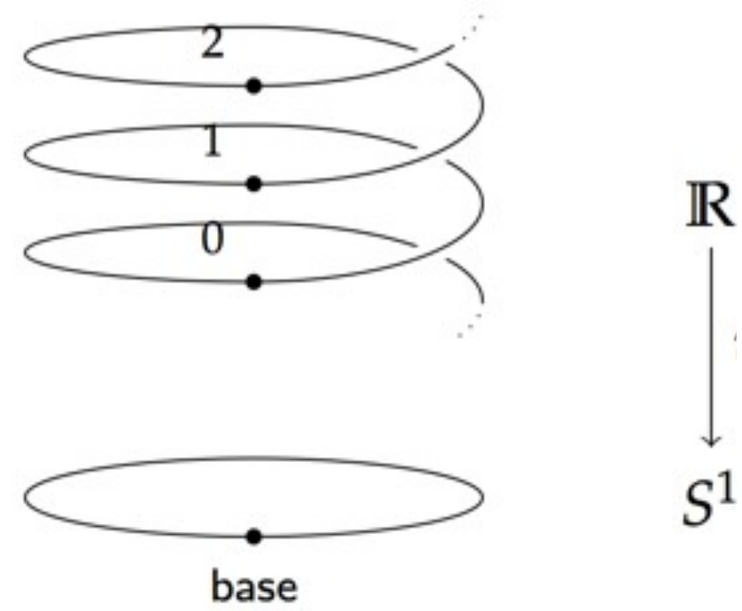
Family of types $(E(x))_{x:B}$

- *Fibers:* $E(b)$ is a type for all $b:B$
- *transport:* equivalence $E(b_1) \simeq E(b_2)$ for all $p: b_1 =_B b_2$

Fibration = Family of types

Fibration (classically):

map $p : E \to B$ such that any path from $p(e)$ to y lifts to a path in E from e to some point in $p^{-1}(y)$



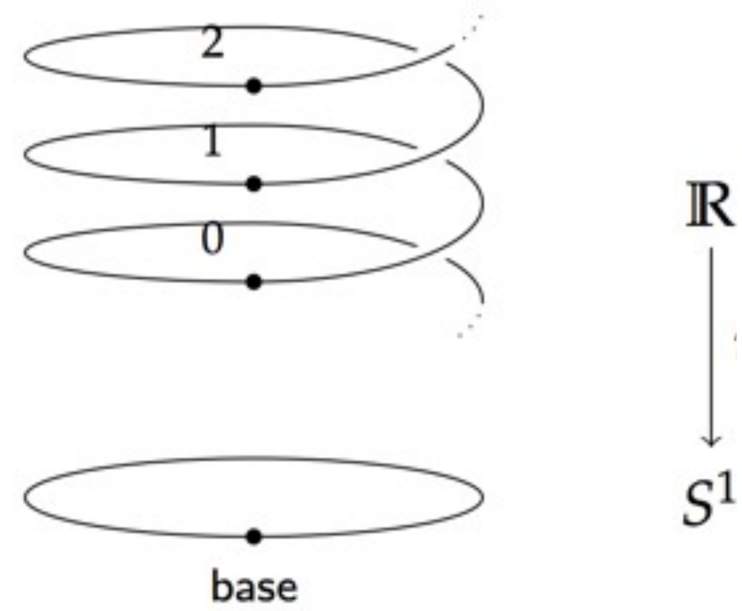
Family of types $(E(x))_{x:B}$

$p^{-1}(b)$

- *Fibers:* $E(b)$ is a type for all $b : B$
- *transport:* equivalence $E(b_1) \simeq E(b_2)$ for all $p : b_1 =_B b_2$

Fibration = Family of types

Fibration (classically):
map $p: E \to B$ such that
any path from $p(e)$ to y
lifts to a path in E from e
to some point in $p^{-1}(y)$



Family of types $(E(x))_{x:B}$

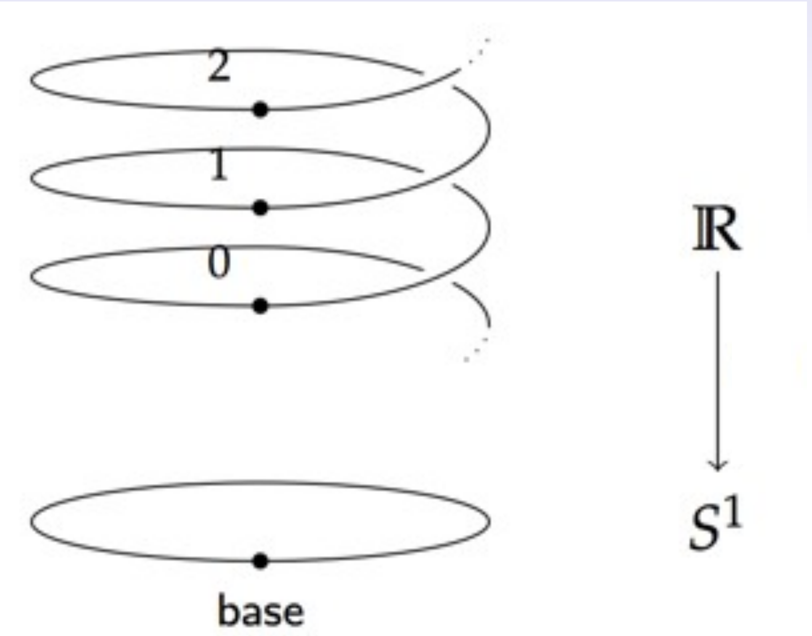
$p^{-1}(b)$

- *Fibers:* $E(b)$ is a type for all $b:B$
- *transport:* equivalence $E(b_1) \xrightarrow{\sim} E(b_2)$ for all $p : b_1 =_B b_2$

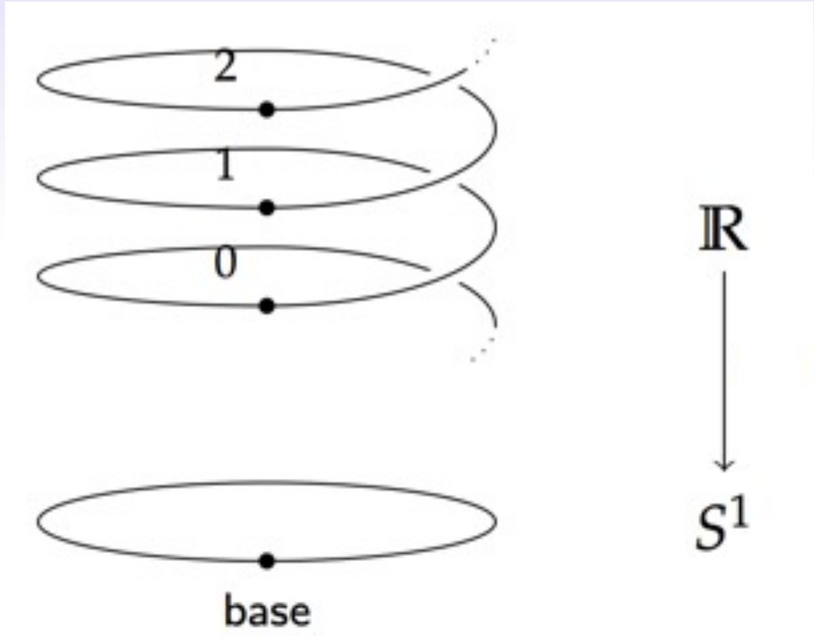
sends $e \in E(x)$ to other endpoint of lifting of p

Universal Cover

family of types $(\texttt{Cover(x)})_{\texttt{x:S1}}$



Universal Cover

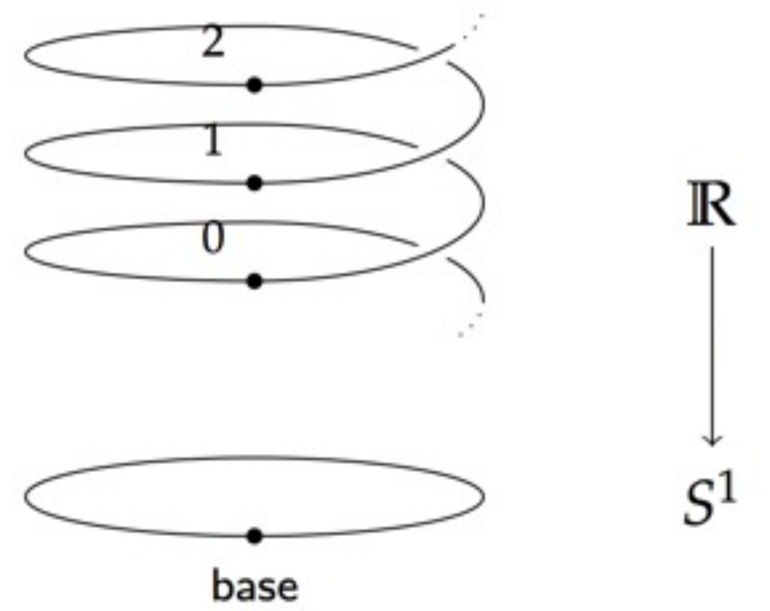


family of types $(\mathtt{Cover(x)})_{\mathtt{x:S1}}$

By circle recursion, it suffices to give

- Fiber over `base`: the type $\mathbb{Z}$
- Equivalence $\mathbb{Z} \simeq \mathbb{Z}$ as lifting of `loop`: `successor`

Universal Cover



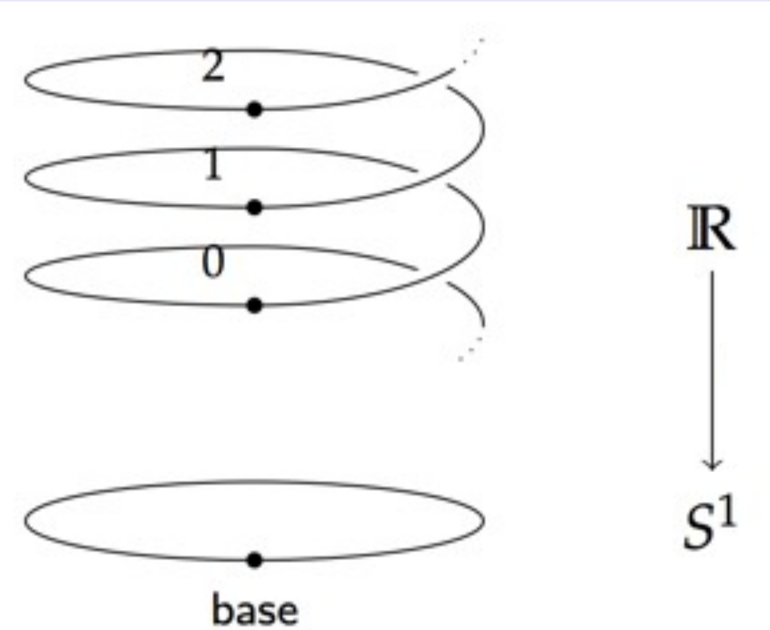
family of types $(\texttt{Cover(x)})_{\texttt{x:S1}}$

By circle recursion, it suffices to give

- Fiber over `base`: the type $\mathbb{Z}$
- Equivalence $\mathbb{Z} \simeq \mathbb{Z}$ as lifting of `loop`: `successor`

uses univalence

Universal Cover



family of types $(\mathtt{Cover(x)})_{\mathtt{x:S1}}$

By circle recursion, it suffices to give

- Fiber over `base`: the type $\mathbb{Z}$
- Equivalence $\mathbb{Z} \simeq \mathbb{Z}$ as lifting of `loop`: `successor`

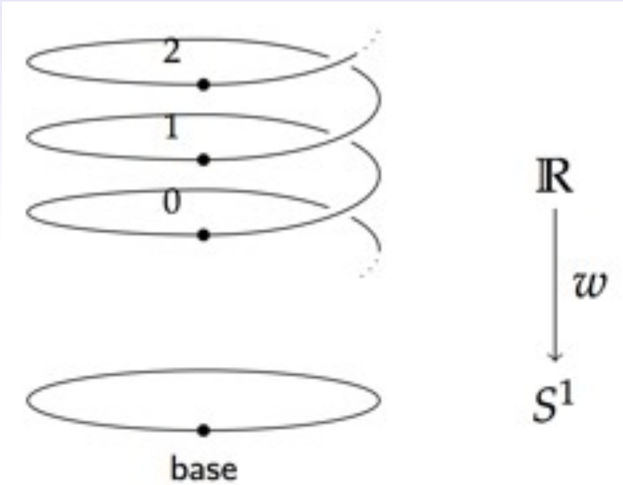
uses univalence

Defining equations:

$\mathtt{Cover(base)} := \mathbb{Z}$

$\mathtt{transport}_{\mathtt{Cover}}\mathtt{(loop)} := \mathtt{successor}$

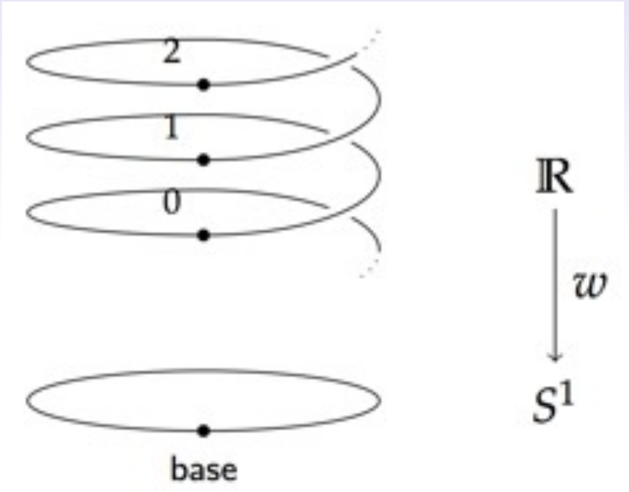
Winding number



$$\texttt{wind} : \Omega(S^1) \to \mathbb{Z}$$
$$\texttt{wind}(p) = \texttt{transport}_{\texttt{Cover}}(p,0)$$

lift p to cover, starting at 0

Winding number



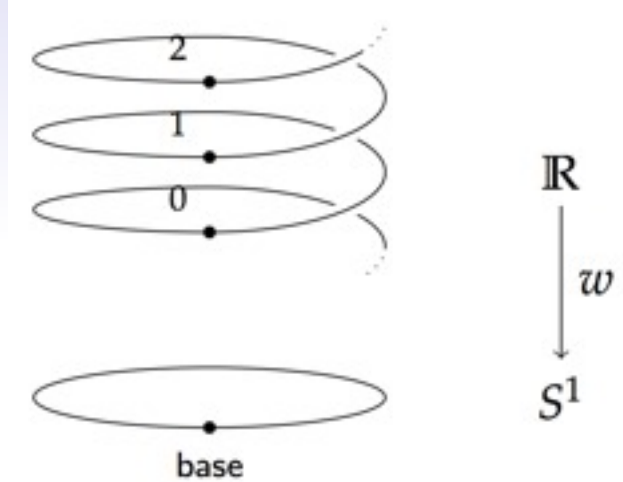
$\texttt{wind} : \Omega(S^1) \to \mathbb{Z}$

$\texttt{wind}(p) = \texttt{transport}_{\texttt{Cover}}(p, 0)$

lift p to cover, starting at 0

$\texttt{wind}(\texttt{loop}^{-1} \circ \texttt{loop})$

Winding number



$\texttt{wind} : \Omega(S^1) \to \mathbb{Z}$

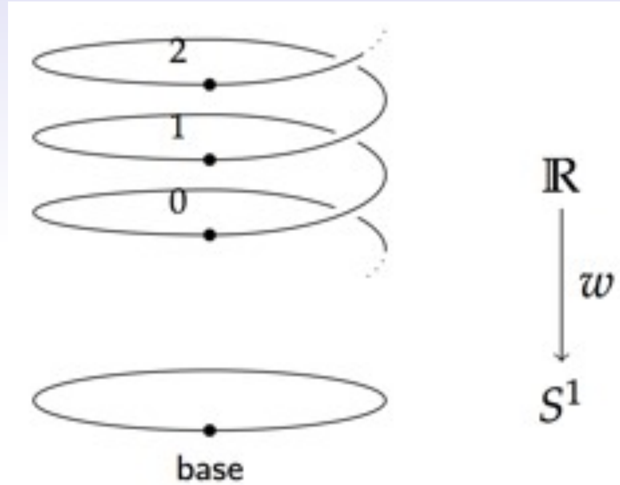
$\texttt{wind(p)} = \texttt{transport}_{\texttt{Cover}}\texttt{(p,0)}$

lift p to cover, starting at 0

$\texttt{wind(loop}^{-1}\texttt{ o loop)}$

$= \texttt{transport}_{\texttt{Cover}}\texttt{(loop}^{-1}\texttt{ o loop, 0)}$

Winding number

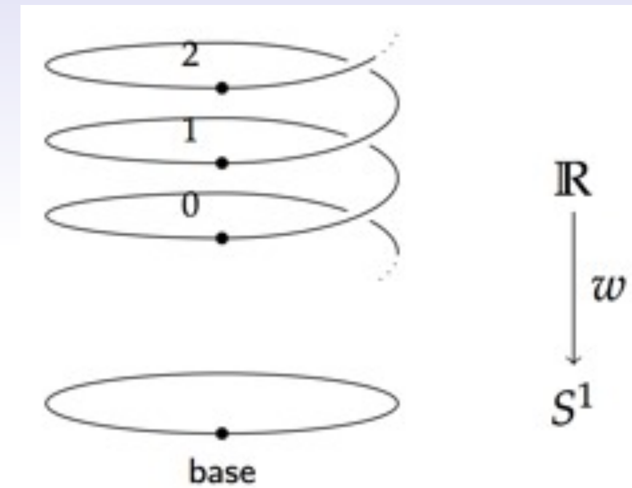


$$\text{wind} : \Omega(S^1) \to \mathbb{Z}$$
$$\text{wind}(p) = \text{transport}_{\text{Cover}}(p, 0)$$

lift p to cover, starting at 0

$$\text{wind}(\text{loop}^{-1} \circ \text{loop})$$
$$= \text{transport}_{\text{Cover}}(\text{loop}^{-1} \circ \text{loop}, 0)$$
$$= \text{transport}_{\text{Cover}}(\text{loop}^{-1}, \text{transport}_{\text{Cover}}(\text{loop}, 0))$$

Winding number

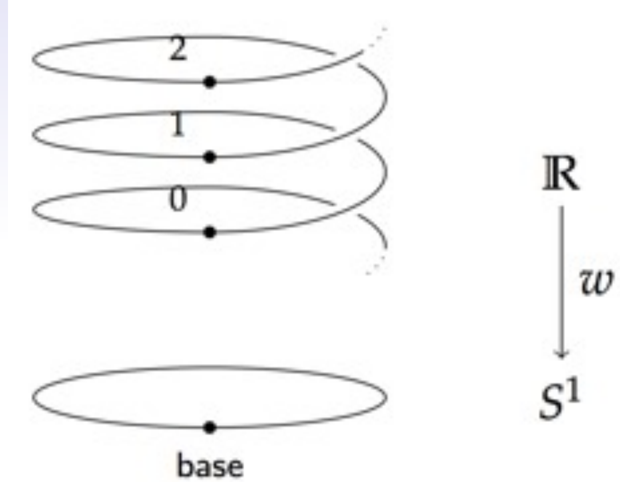


$$\text{wind} : \Omega(S^1) \to \mathbb{Z}$$
$$\text{wind}(p) = \text{transport}_{\text{Cover}}(p, 0)$$

lift p to cover, starting at 0

$$\begin{aligned} &\text{wind}(\text{loop}^{-1} \circ \text{loop}) \\ = \; &\text{transport}_{\text{Cover}}(\text{loop}^{-1} \circ \text{loop}, 0) \\ = \; &\text{transport}_{\text{Cover}}(\text{loop}^{-1}, \text{transport}_{\text{Cover}}(\text{loop}, 0)) \\ = \; &\text{transport}_{\text{Cover}}(\text{loop}^{-1}, 1) \end{aligned}$$

Winding number



$\mathtt{wind} : \Omega(S^1) \to \mathbb{Z}$

$\mathtt{wind(p)} = \mathtt{transport}_{\mathtt{Cover}}\mathtt{(p,0)}$

lift p to cover, starting at 0

$\mathtt{wind(loop^{-1}\ o\ loop)}$

$= \mathtt{transport}_{\mathtt{Cover}}\mathtt{(loop^{-1}\ o\ loop,\ 0)}$

$= \mathtt{transport}_{\mathtt{Cover}}\mathtt{(loop^{-1},\ transport}_{\mathtt{Cover}}\mathtt{(loop,0))}$

$= \mathtt{transport}_{\mathtt{Cover}}\mathtt{(loop^{-1},\ 1)}$

$= 0$

So far

Theorem. $\Omega(S^1)$ is equivalent to $\mathbb{Z}$

Proof: two functions

```
wind    : Ω(S¹) → ℤ
wind(p) = transport_Cover(p,0)

loop⁻  : ℤ → Ω(S¹)
loop⁰   = id
loop⁺ⁿ  = loop o loop o … loop          (n times)
loop⁻ⁿ  = loop⁻¹ o loop⁻¹ o … loop⁻¹   (n times)
```

So far

Theorem. $\Omega(S^1)$ is equivalent to $\mathbb{Z}$

Proof: two **mutually inverse** functions

```
wind    : Ω(S¹) → ℤ
wind(p) = transport_Cover(p,0)

loop⁻  : ℤ → Ω(S¹)
loop⁰    = id
loop⁺ⁿ  = loop o loop o … loop
loop⁻ⁿ  = loop⁻¹ o loop⁻¹ o … loop⁻¹
```

$$\mathtt{wind} : \Omega(S^1) \to \mathbb{Z}$$
$$\mathtt{wind(p)} = \mathtt{transport}_{\mathtt{Cover}}\mathtt{(p,0)}$$

$$\mathtt{loop}^{-} : \mathbb{Z} \to \Omega(S^1)$$
$$\mathtt{loop}^{0} = \mathtt{id}$$
$$\mathtt{loop}^{+n} = \mathtt{loop} \circ \mathtt{loop} \circ \ldots \mathtt{loop} \quad \text{(n times)}$$
$$\mathtt{loop}^{-n} = \mathtt{loop}^{-1} \circ \mathtt{loop}^{-1} \circ \ldots \mathtt{loop}^{-1} \quad \text{(n times)}$$

Composite #1

Lemma. $\forall n.$ `wind(loop`^{n}`) = n`

Proof: induction on n. E.g.

Composite #1

Lemma. $\forall n.$ `wind(loop`^{n}`) = n`

Proof: induction on n. E.g.

`wind(loop`$^{n+1}$`)`

Composite #1

Lemma. $\forall n.\ \mathtt{wind}(\mathtt{loop}^n) = n$

Proof: induction on n. E.g.

$$
\begin{aligned}
&\mathtt{wind}(\mathtt{loop}^{n+1}) \\
= \;&\mathtt{wind}(\mathtt{loop}\ \mathtt{o}\ \mathtt{loop}^n) && [\text{def. } \mathtt{loop}^{-}]
\end{aligned}
$$

Composite #1

Lemma. $\forall n.\ \texttt{wind}(\texttt{loop}^n) = n$

Proof: induction on n. E.g.

$$
\begin{aligned}
& \texttt{wind}(\texttt{loop}^{n+1}) \\
&= \texttt{wind}(\texttt{loop o loop}^n) && [\texttt{def. loop}^-] \\
&= \texttt{transport}_{\texttt{Cover}}(\texttt{loop o loop}^n\texttt{,0}) && [\texttt{def. wind}]
\end{aligned}
$$

Composite #1

Lemma. $\forall n.\ \mathtt{wind}(\mathtt{loop}^n) = n$

Proof: induction on n. E.g.

$$
\begin{aligned}
& \mathtt{wind}(\mathtt{loop}^{n+1}) \\
= \ & \mathtt{wind}(\mathtt{loop} \circ \mathtt{loop}^n) && [\mathtt{def.\ loop}^-] \\
= \ & \mathtt{transport}_{\mathtt{Cover}}(\mathtt{loop} \circ \mathtt{loop}^n, 0) && [\mathtt{def.\ wind}] \\
= \ & \mathtt{transport}_{\mathtt{Cover}}(\mathtt{loop}, \\
& \qquad \mathtt{transport}_{\mathtt{Cover}}(\mathtt{loop}^n, 0)) && [\mathtt{functorial}]
\end{aligned}
$$

Composite #1

Lemma. $\forall n.\ \mathtt{wind}(\mathtt{loop}^n) = n$

Proof: induction on n. E.g.

$$
\begin{aligned}
& \mathtt{wind}(\mathtt{loop}^{n+1}) \\
&= \mathtt{wind}(\mathtt{loop} \circ \mathtt{loop}^{n}) && [\text{def. } \mathtt{loop}^{-}] \\
&= \mathtt{transport}_{\mathtt{Cover}}(\mathtt{loop} \circ \mathtt{loop}^{n}, 0) && [\text{def. } \mathtt{wind}] \\
&= \mathtt{transport}_{\mathtt{Cover}}(\mathtt{loop}, \\
&\qquad \mathtt{transport}_{\mathtt{Cover}}(\mathtt{loop}^{n}, 0)) && [\text{functorial}] \\
&= \mathtt{transport}_{\mathtt{Cover}}(\mathtt{loop}, n) && [\text{IH}]
\end{aligned}
$$

Composite #1

Lemma. $\forall$n. `wind(loop`^{n}`) = n`

Proof: induction on n. E.g.

$$
\begin{aligned}
& \mathtt{wind(loop}^{n+1}\mathtt{)} \\
&= \mathtt{wind(loop\ o\ loop}^{n}\mathtt{)} && \mathtt{[def.\ loop}^{-}\mathtt{]} \\
&= \mathtt{transport}_{\mathtt{Cover}}\mathtt{(loop\ o\ loop}^{n}\mathtt{,0)} && \mathtt{[def.\ wind]} \\
&= \mathtt{transport}_{\mathtt{Cover}}\mathtt{(loop,} \\
&\qquad \mathtt{transport}_{\mathtt{Cover}}\mathtt{(loop}^{n}\mathtt{,0))} && \mathtt{[functorial]} \\
&= \mathtt{transport}_{\mathtt{Cover}}\mathtt{(loop,n)} && \mathtt{[IH]} \\
&= \mathtt{n+1} && \mathtt{[def.\ Cover]}
\end{aligned}
$$

Composite #1

```
wind-loop^ : (n : Int) → Path (wind (loop^ n)) n
wind-loop^ Zero = id
wind-loop^ (Pos One) = ap≃ transport-Cover-loop
wind-loop^ (Pos (S n)) =
  transport Cover (loop ∙ loop^ (Pos n)) Zero        ≃⟨ ap≃ (transport-∙ Cover loop (loop^ (Pos n))) ⟩
  transport Cover loop
           (transport Cover (loop^ (Pos n)) Zero)  ≃⟨ ap (transport Cover loop) (wind-loop^ (Pos n))  ⟩
  transport Cover loop (Pos n)                      ≃⟨ ap≃ transport-Cover-loop ⟩
  succ (Pos n) ∎
wind-loop^ (Neg One) = ap≃ transport-Cover-!loop
wind-loop^ (Neg (S n)) =
  transport Cover (! loop ∙ loop^ (Neg n)) Zero      ≃⟨ ap≃ (transport-∙ Cover (! loop) (loop^ (Neg n))) ⟩
  transport Cover (! loop)
           (transport Cover (loop^ (Neg n)) Zero)   ≃⟨ ap≃ transport-Cover-!loop ⟩
  pred (transport Cover (loop^ (Neg n)) Zero)        ≃⟨ ap pred (wind-loop^ (Neg n)) ⟩
  pred (Neg n) ∎
```

Composite #2

Lem. $\forall p{:}\mathtt{base}{=}\mathtt{base}.\ \mathtt{loop}^{\mathtt{wind}(p)} = p$

Composite #2

Lem. $\forall$`p:base=base. loop`$^{\texttt{wind(p)}}$ `= p`

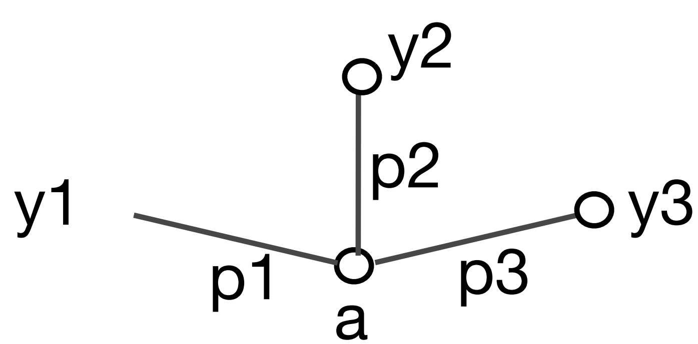
Proof: want to apply path induction
but path induction does not apply to loops

Composite #2

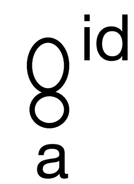
Lem. $\forall p{:}\mathtt{base}{=}\mathtt{base}.\ \mathtt{loop}^{\mathtt{wind}(p)} = p$

Proof: want to apply path induction
but path induction does not apply to loops

Type of paths from a to somewhere



is inductively generated by



Composite #2

Lem. $\forall y:S^1,\ p:\mathsf{base}=y.\ \mathsf{loop}^{\mathsf{wind}(p)} = p$

Composite #2

Lem. $\forall y{:}S^1,\ p{:}\mathtt{base}{=}y.\ \mathtt{loop}^{\mathtt{wind}(p)} = p$

Proof:

$\mathtt{wind} \quad : \mathtt{base}{=}\mathtt{base} \to \mathbb{Z}$

$\mathtt{wind}(p) = \mathtt{transport}_{\mathtt{Cover}}(p,0)$

$\mathtt{loop}^{-} \quad : \mathbb{Z} \to \Omega(S^1)$

Composite #2

Lem. $\forall y{:}S^1,\ p{:}\mathtt{base}{=}y.\ \mathtt{loop}^{\mathtt{wind}(p)} = p$

Proof: need to generalize wind

$$\mathtt{wind} : \mathtt{base}{=}\mathtt{base} \to \mathbb{Z}$$

$$\mathtt{wind}(p) = \mathtt{transport}_{\mathtt{Cover}}(p, 0)$$

$$\mathtt{loop}^{-} : \mathbb{Z} \to \Omega(S^1)$$

Composite #2

Lem. $\forall y{:}S^1,\ p{:}\mathrm{base}{=}y.\ \mathtt{loop}^{\mathtt{wind}(p)} = p$

Proof:

$$\mathtt{encode} \;:\; \forall y{:}S^1.\ \mathtt{base}{=}y \to \mathtt{Cover}(y)$$
$$\mathtt{encode}(p) = \mathtt{transport}_{\mathtt{Cover}}(p,0)$$

$$\mathtt{loop}^{-} \;:\; \mathbb{Z} \to \Omega(S^1)$$

Composite #2

Lem. $\forall y{:}S^1,\ p{:}\mathrm{base}{=}y.\ \mathrm{loop}^{\mathrm{wind}(p)} = p$

Proof: need to generalize loop^{-}

$$\mathrm{encode} : \forall y{:}S^1.\ \mathrm{base}{=}y \to \mathrm{Cover}(y)$$
$$\mathrm{encode}(p) = \mathrm{transport}_{\mathrm{Cover}}(p,0)$$

$$\mathrm{loop}^{-} : \mathbb{Z} \to \Omega(S^1)$$

Composite #2

Lem. $\forall y{:}S^1,\ p{:}\mathtt{base}{=}y.\ \mathtt{loop}^{\mathtt{wind}(p)} = p$

Proof: need to generalize wind and loop^{-}

$$\mathtt{encode} \ : \ \forall y{:}S^1.\ \mathtt{base}{=}y \to \mathtt{Cover}(y)$$
$$\mathtt{encode}(p) = \mathtt{transport}_{\mathtt{Cover}}(p,0)$$

$$\mathtt{decode} \ : \ \forall y{:}S^1.\ \mathtt{Cover}(y) \to \mathtt{base}{=}y$$

Composite #2

Lem. $\forall y{:}S^1,\ p{:}\mathrm{base}{=}y.\ \mathrm{decode}_y(\mathrm{encode}_y(p)) = p$

Proof: need to generalize wind and loop$^-$

$$\mathrm{encode} \quad : \forall y{:}S^1.\ \mathrm{base}{=}y \to \mathrm{Cover}(y)$$
$$\mathrm{encode}(p) = \mathrm{transport}_{\mathrm{Cover}}(p, 0)$$

$$\mathrm{decode} \quad : \forall y{:}S^1.\ \mathrm{Cover}(y) \to \mathrm{base}{=}y$$

Decode

$$\texttt{decode} \;:\; \forall y{:}S^1.\ \texttt{Cover}(y) \to \texttt{base}{=}y$$

Defined by circle induction:

```
decode(base) := loop⁻
decode(loop) := …
```

Decode

$$\texttt{decode} \;:\; \forall y{:}S^1.\ \texttt{Cover}(y) \to \texttt{base}=y$$

Defined by circle induction:

Cover(base) → base=base

```
decode(base) := loop⁻
decode(loop) := …
```

Decode

`decode : ∀y:S¹. Cover(y) → base=y`

Defined by circle induction:

Cover(base) → base=base

```
decode(base) := loop⁻
decode(loop) := …
```

"`loop⁻` is invariant under going around the `loop` in the fibration `Cover(y) → base=y`"

Decode

```
decode : {x : S¹} → Cover x → Path base x
decode {x} =
 S¹-induction
  (λ x' → Cover x' → Path base x')
  loop^
  loop^-respects-loop
  x where
    loop^-respects-loop : transport (λ x' →  Cover x' → Path base x') loop loop^ ≡ (λ n → loop^ n)
    loop^-respects-loop =
       (transport (λ x' →  Cover x' → Path base x') loop loop^  ≃〈 transport-→ Cover (Path base) loop loop^ 〉

          transport (λ x' → Path base x') loop
        o loop^
        o transport Cover (! loop)                             ≃〈 λ≃ (λ y → transport-Path-right loop (loop^ (transport Cover (! loop) y))) 〉

          (λ p → loop · p)
        o loop^
        o transport Cover (! loop)                             ≃〈 λ≃ (λ y → ap (λ x' → loop · loop^ x') (ap≃ transport-Cover-!loop)) 〉

          (λ p → loop · p)
        o loop^
        o pred                                                 ≃〈 id 〉

        (λ n → loop · (loop^ (pred n)))                        ≃〈 λ≃ (λ y → move-left-! _ loop (loop^ y) (loop^-preserves-pred y)) 〉

        (λ n → loop^ n)
        ∎)
```

Composite #2

Lem. $\forall y{:}S^1,\ p{:}base{=}y.\ \mathrm{decode}_y(\mathrm{encode}_y(p)) = p$

$\mathrm{encode} \ :\ \forall y{:}S^1.\ base{=}y \to \mathrm{Cover}(y)$

$\mathrm{encode}_y(p) = \mathrm{transport}_{\mathrm{Cover}}(p, 0)$

$\mathrm{decode} \ :\ \forall y{:}S^1.\ \mathrm{Cover}(y) \to base{=}y$

$\mathrm{decode}_{base}(n) = \mathrm{loop}^n$

Composite #2

Lem. $\forall y{:}S^1,\ p{:}\mathrm{base}{=}y.\ \ \mathrm{decode}_y(\mathrm{encode}_y(p)) = p$

Proof: By **path induction**, suffices to show

$$\mathrm{decode}_{\mathrm{base}}(\mathrm{encode}_{\mathrm{base}}(\mathrm{id}))$$

$$= \mathrm{id}$$

$\mathrm{encode} \ :\ \forall y{:}S^1.\ \mathrm{base}{=}y \to \mathrm{Cover}(y)$

$\mathrm{encode}_y(p) = \mathrm{transport}_{\mathrm{Cover}}(p,0)$

$\mathrm{decode} \ :\ \forall y{:}S^1.\ \mathrm{Cover}(y) \to \mathrm{base}{=}y$

$\mathrm{decode}_{\mathrm{base}}(n) = \mathrm{loop}^n$

Composite #2

Lem. $\forall y{:}S^1,\ p{:}base{=}y.\ \ decode_y(encode_y(p)) = p$

Proof: By **path induction**, suffices to show

$$decode_{base}(encode_{base}(id))$$

$$= decode_{base}(0)$$

$$= id$$

$encode\ :\ \forall y{:}S^1.\ base{=}y \rightarrow Cover(y)$

$encode_y(p) = transport_{Cover}(p,0)$

$decode\ :\ \forall y{:}S^1.\ Cover(y) \rightarrow base{=}y$

$decode_{base}(n) = loop^n$

Composite #2

Lem. $\forall y{:}S^1,\ p{:}\mathrm{base}{=}y.\ \ \mathrm{decode}_y(\mathrm{encode}_y(p)) = p$

Proof: By **path induction**, suffices to show

$$
\begin{aligned}
&\mathrm{decode}_{\mathrm{base}}(\mathrm{encode}_{\mathrm{base}}(\mathrm{id}))\\
&= \mathrm{decode}_{\mathrm{base}}(0)\\
&= \mathrm{loop}^0\\
&= \mathrm{id}
\end{aligned}
$$

$\mathrm{encode} \ : \ \forall y{:}S^1.\ \mathrm{base}{=}y \to \mathrm{Cover}(y)$

$\mathrm{encode}_y(p) = \mathrm{transport}_{\mathrm{Cover}}(p,0)$

$\mathrm{decode} \ : \ \forall y{:}S^1.\ \mathrm{Cover}(y) \to \mathrm{base}{=}y$

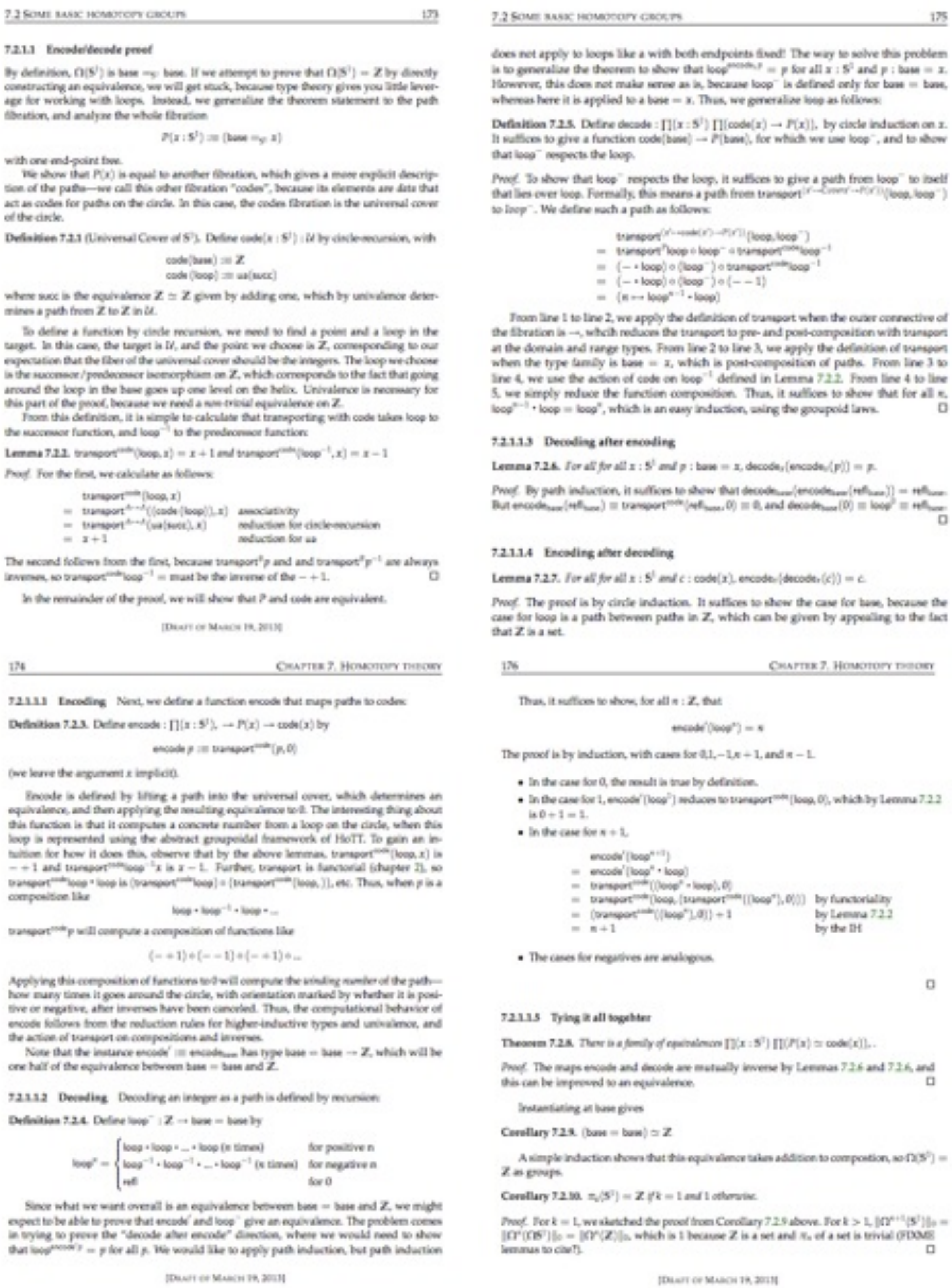
$\mathrm{decode}_{\mathrm{base}}(n) = \mathrm{loop}^n$

Composite #2

```
decode-encode  : {x : S¹} (α : Path base x)
                 → Path (decode (encode α)) α
decode-encode id = id
```

Fundamental group of the circle

The book



Computer-checked

Conclusion

We can do **computer-checked proofs** in **synthetic** homotopy theory

- Proofs are constructive*: can run them
- Results apply in a variety of settings, from simplicial sets (hence topological spaces) to Quillen model categories and ∞-topoi*
- New type-theoretic proofs/methods

*work in progress

Homotopy in HoTT

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

$\pi_{k<n}(S^n) = 0$

Hopf fibration

$\pi_2(S^2) = \mathbb{Z}$

$\pi_3(S^2) = \mathbb{Z}$

James Construction

$\pi_4(S^3) = \mathbb{Z}_?$

Freudenthal

$\pi_n(S^n) = \mathbb{Z}$

K(G,n)

Cohomology axioms

Blakers-Massey

Van Kampen

Covering spaces

Whitehead for n-types

[Brunerie, Finster, Hou, Licata, Lumsdaine, Shulman]

Homotopy in HoTT

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

$\pi_{k<n}(S^n) = 0$

Hopf fibration

$\pi_2(S^2) = \mathbb{Z}$

$\pi_3(S^2) = \mathbb{Z}$

James Construction

$\pi_4(S^3) = \mathbb{Z}_?$

Freudenthal

$\pi_n(S^n) = \mathbb{Z}$

K(G,n)

Cohomology axioms

Blakers-Massey

Van Kampen

Covering spaces

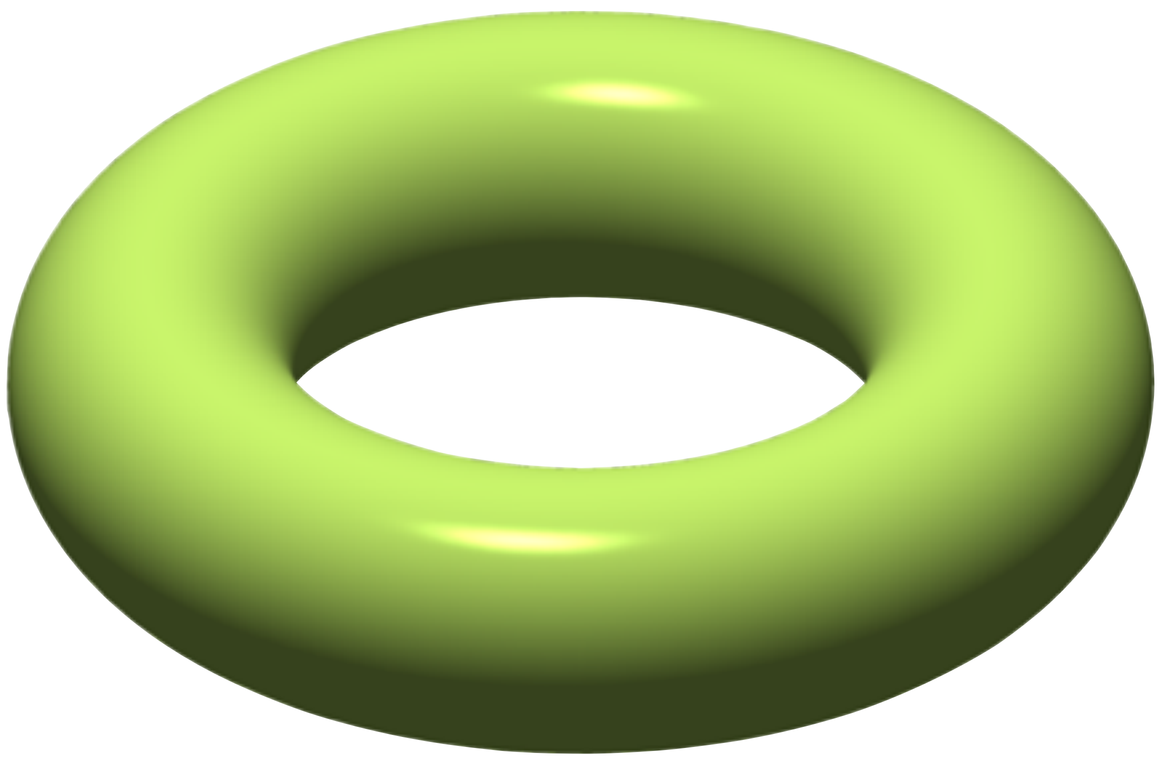
Whitehead for n-types

[Brunerie, Finster, Hou, Licata, Lumsdaine, Shulman]

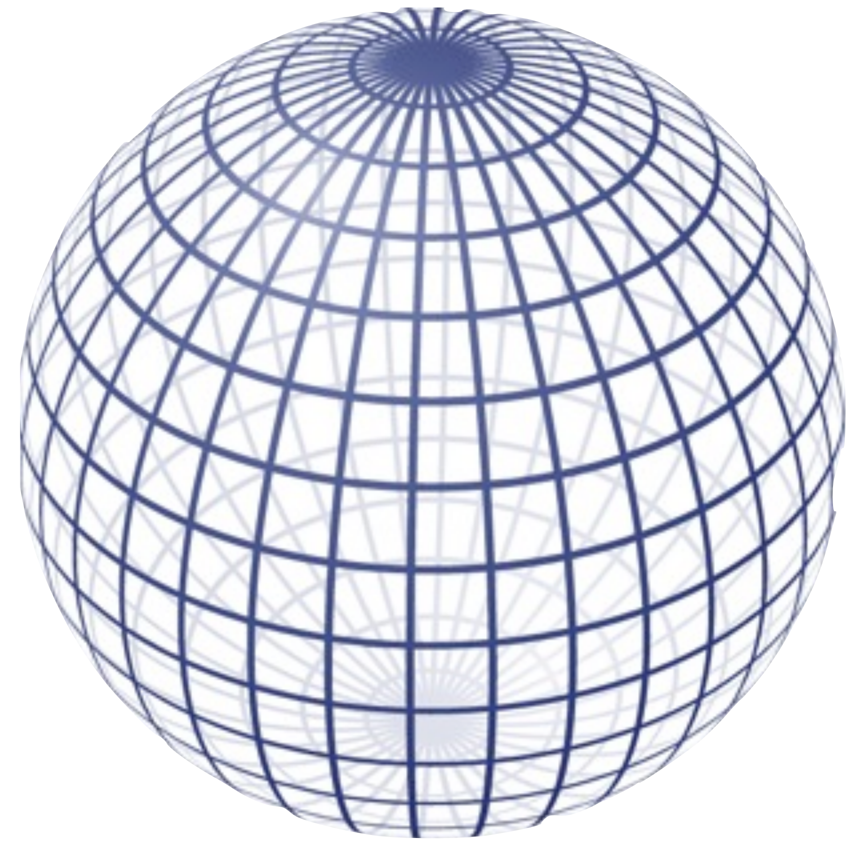
Reading list

- $\pi_1(S^1) = \mathbb{Z}$ [Licata and Shulman, LICS'13]
- Other results: forthcoming Homotopy Type Theory book
- Blog: `homotopytypetheory.org`
- Formalizations:
  `github.com/dlicata335/hott-agda`
  `github.com/hott/hott-agda`
  `github.com/hott/hott [Coq]`

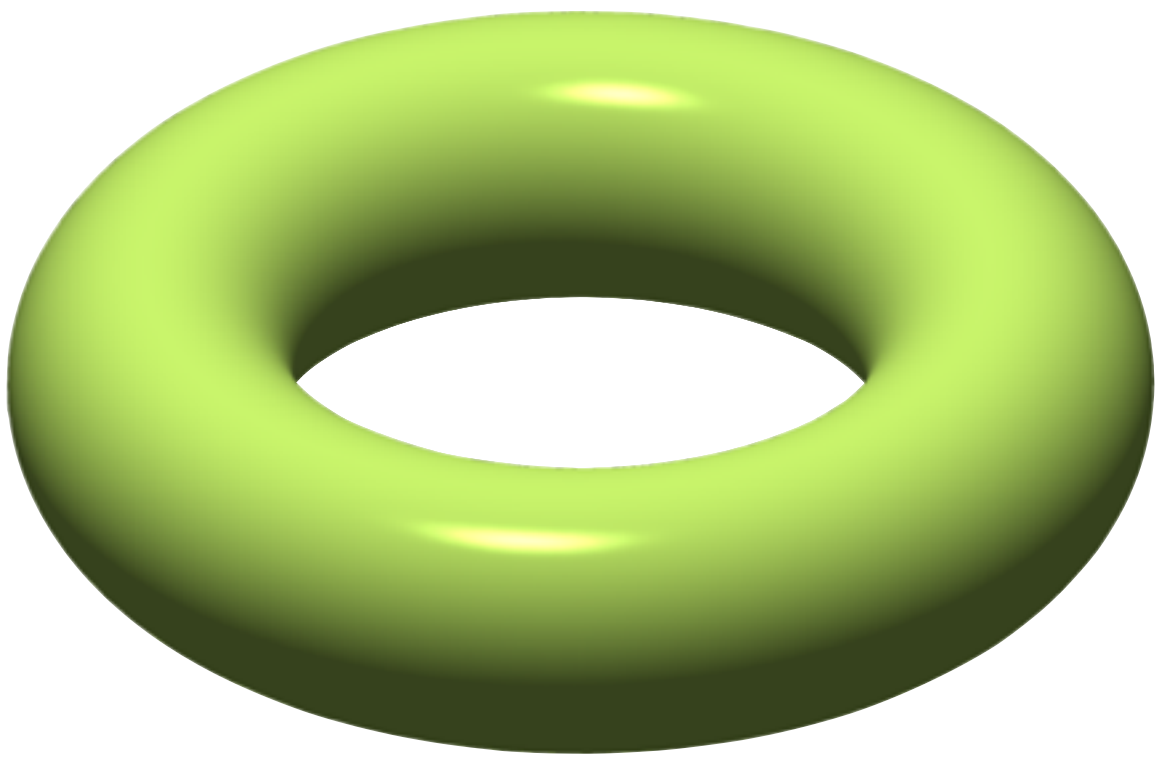
Algebraic invariants



$\neq$

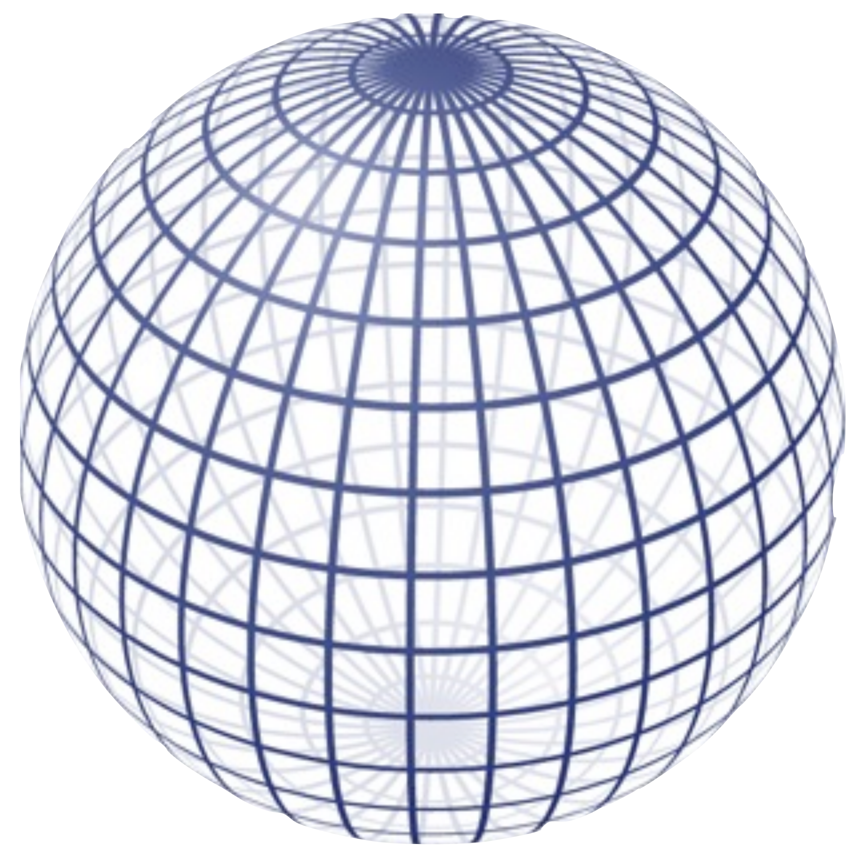


Algebraic invariants



fundamental group
is non-trivial ($\mathbb{Z} \times \mathbb{Z}$)

$\neq$



fundamental group
is trivial

Homotopy groups

Homotopy groups

- Fundamental group π_1: group of loops

Homotopy groups

- Fundamental group π_1: group of loops
- π_2: group of homotopies

Homotopy groups

- Fundamental group π_1: group of loops
- π_2: group of homotopies
- π_3: group of homotopies between homotopies

Homotopy groups

- Fundamental group π_1: group of loops
- π_2: group of homotopies
- π_3: group of homotopies between homotopies
- …

Homotopy groups

k^{th} homotopy group

n-dimensional sphere

	π_1	π_2	π_3	π_4	π_5	π_6	π_7	π_8	π_9	π_{10}	π_{11}	π_{12}	π_{13}	π_{14}	π_{15}
S^0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
S^1	Z	0	0	0	0	0	0	0	0	0	0	0	0	0	0
S^2	0	Z	Z	Z_2	Z_2	Z_{12}	Z_2	Z_2	Z_3	Z_{15}	Z_2	Z_2^2	$Z_{12}\times Z_2$	$Z_{84}\times Z_2^2$	Z_2^2
S^3	0	0	Z	Z_2	Z_2	Z_{12}	Z_2	Z_2	Z_3	Z_{15}	Z_2	Z_2^2	$Z_{12}\times Z_2$	$Z_{84}\times Z_2^2$	Z_2^2
S^4	0	0	0	Z	Z_2	Z_2	$Z\times Z_{12}$	Z_2^2	Z_2^2	$Z_{24}\times Z_3$	Z_{15}	Z_2	Z_2^3	$Z_{120}\times Z_{12}\times Z_2$	$Z_{84}\times Z_2^5$
S^5	0	0	0	0	Z	Z_2	Z_2	Z_{24}	Z_2	Z_2	Z_2	Z_{30}	Z_2	Z_2^3	$Z_{72}\times Z_2$
S^6	0	0	0	0	0	Z	Z_2	Z_2	Z_{24}	0	Z	Z_2	Z_{60}	$Z_{24}\times Z_2$	Z_2^3
S^7	0	0	0	0	0	0	Z	Z_2	Z_2	Z_{24}	0	0	Z_2	Z_{120}	Z_2^3
S^8	0	0	0	0	0	0	0	Z	Z_2	Z_2	Z_{24}	0	0	Z_2	$Z\times Z_{120}$

[image from wikipedia]

$\pi_k(S^n)$ in HoTT

k^{th} homotopy group

n-dimensional sphere

	π_1	π_2	π_3	π_4	π_5	π_6	π_7	π_8	π_9	π_{10}	π_{11}	π_{12}	π_{13}	π_{14}	π_{15}
S^0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
S^1	$\mathbb{Z}$	0	0	0	0	0	0	0	0	0	0	0	0	0	0
S^2	0	$\mathbb{Z}$	$\mathbb{Z}$	$\mathbb{Z}_2$	$\mathbb{Z}_2$	$\mathbb{Z}_{12}$	$\mathbb{Z}_2$	$\mathbb{Z}_2$	$\mathbb{Z}_3$	$\mathbb{Z}_{15}$	$\mathbb{Z}_2$	$\mathbb{Z}_2^2$	$\mathbb{Z}_{12}\times\mathbb{Z}_2$	$\mathbb{Z}_{84}\times\mathbb{Z}_2^2$	$\mathbb{Z}_2^2$
S^3	0	0	$\mathbb{Z}$	$\mathbb{Z}_2$	$\mathbb{Z}_2$	$\mathbb{Z}_{12}$	$\mathbb{Z}_2$	$\mathbb{Z}_2$	$\mathbb{Z}_3$	$\mathbb{Z}_{15}$	$\mathbb{Z}_2$	$\mathbb{Z}_2^2$	$\mathbb{Z}_{12}\times\mathbb{Z}_2$	$\mathbb{Z}_{84}\times\mathbb{Z}_2^2$	$\mathbb{Z}_2^2$
S^4	0	0	0	$\mathbb{Z}$	$\mathbb{Z}_2$	$\mathbb{Z}_2$									
S^5	0	0	0	0	$\mathbb{Z}$	$\mathbb{Z}_2$	$\mathbb{Z}_2$	$\mathbb{Z}_{24}$							
S^6	0	0	0	0	0	$\mathbb{Z}$	$\mathbb{Z}_2$	$\mathbb{Z}_2$	$\mathbb{Z}_{24}$	0					
S^7	0	0	0	0	0	0	$\mathbb{Z}$	$\mathbb{Z}_2$	$\mathbb{Z}_2$	$\mathbb{Z}_{24}$	0	0			
S^8	0	0	0	0	0	0	0	$\mathbb{Z}$	$\mathbb{Z}_2$	$\mathbb{Z}_2$	$\mathbb{Z}_{24}$	0	0	$\mathbb{Z}_2$	

[image from wikipedia]

$\pi_k(S^n)$ in HoTT

k^{th} homotopy group

n-dimensional sphere

	π_1	π_2	π_3	π_4	π_5	π_6	π_7	π_8	π_9	π_{10}	π_{11}	π_{12}	π_{13}	π_{14}	π_{15}
S^0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
S^1	$\mathbf{Z}$	0	0	0	0	0	0	0	0	0	0	0	0	0	0
S^2	0	$\mathbf{Z}$	$\mathbf{Z}$	Z_2	Z_2	Z_{12}	Z_2	Z_2	Z_3	Z_{15}	Z_2	Z_2^2	$Z_{12}\times Z_2$	$Z_{84}\times Z_2^2$	Z_2^2
S^3	0	0	$\mathbf{Z}$	Z_2	Z_2	Z_{12}	Z_2	Z_2	Z_3	Z_{15}	Z_2	Z_2^2	$Z_{12}\times Z_2$	$Z_{84}\times Z_2^2$	Z_2^2
S^4	0	0	0	$\mathbf{Z}$	Z_2	Z_2									
S^5	0	0	0	0	$\mathbf{Z}$	Z_2	Z_2	Z_{24}							
S^6	0	0	0	0	0	$\mathbf{Z}$	Z_2	Z_2	Z_{24}	0					
S^7	0	0	0	0	0	0	$\mathbf{Z}$	Z_2	Z_2	Z_{24}	0	0			
S^8	0	0	0	0	0	0	0	$\mathbf{Z}$	Z_2	Z_2	Z_{24}	0	0	Z_2	

[image from wikipedia]

$\pi_n(S^n) = \mathbb{Z}$ for $n \geq 1$

Proof: Induction on n

- Base case: $\pi_1(S^1) = \mathbb{Z}$
- Inductive step: $\pi_{n+1}(S^{n+1}) = \pi_n(S^n)$

$\pi_n(S^n) = \mathbb{Z}$ for $n \geq 1$

Proof: Induction on n

- Base case: $\pi_1(S^1) = \mathbb{Z}$
- Inductive step: $\pi_{n+1}(S^{n+1}) = \pi_n(S^n)$

  Key lemma: $\tau_n(S^n) = \tau_n(\Omega(S^{n+1}))$

$\pi_n(S^n) = \mathbb{Z}$ for $n \geq 1$

Proof: Induction on n

- Base case: $\pi_1(S^1) = \mathbb{Z}$
- Inductive step: $\pi_{n+1}(S^{n+1}) = \pi_n(S^n)$

  Key lemma: $\tau_n(S^n) = \tau_n(\Omega(S^{n+1}))$

n-truncation:
best approximation of a type such that all (n+1)-paths are equal

$\pi_n(S^n) = \mathbb{Z}$ for $n \geq 1$

Proof: Induction on n

- Base case: $\pi_1(S^1) = \mathbb{Z}$
- Inductive step: $\pi_{n+1}(S^{n+1}) = \pi_n(S^n)$

Key lemma: $\tau_n(S^n) = \tau_n(\Omega(S^{n+1}))$

n-truncation:
best approximation of a type such that all (n+1)-paths are equal

higher inductive type generated by
$\texttt{base}_n : S^n$
$\texttt{loop}_n : \Omega^n(S^n)$

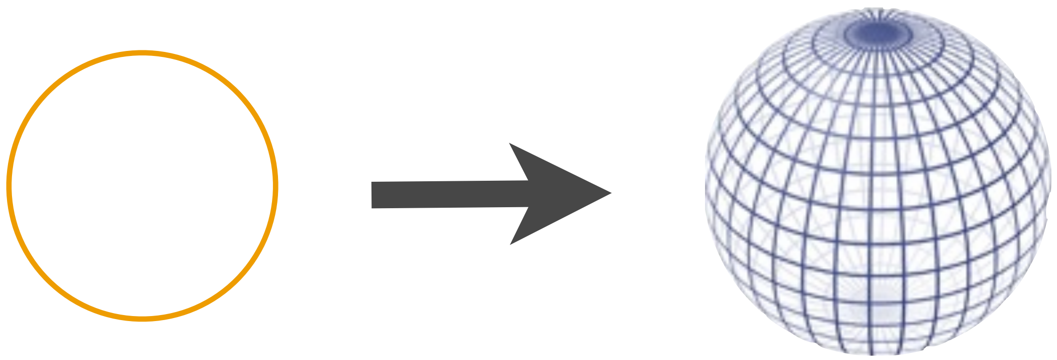
$\tau_n(S^n) = \tau_n(\Omega(S^{n+1}))$

n-truncation of S^n is the type of "codes" for loops on S^{n+1}

$\tau_n(S^n) = \tau_n(\Omega(S^{n+1}))$

n-truncation of S^n is the type of “codes” for loops on S^{n+1}

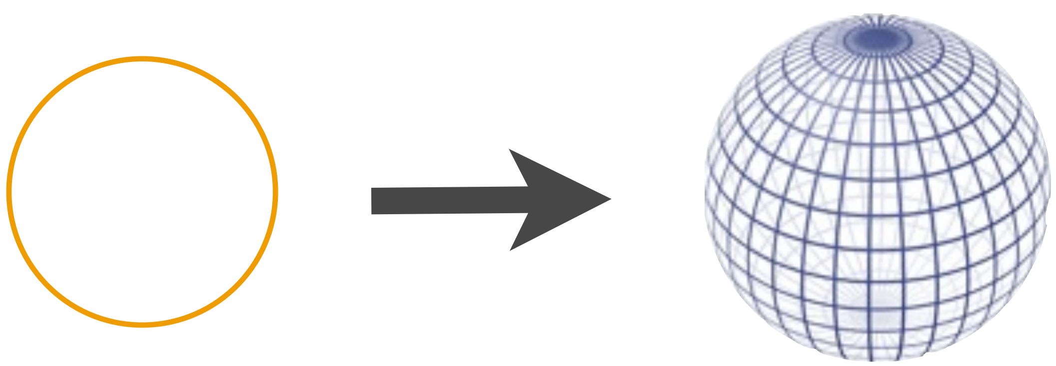
- Decode: promote n-dimensional loop on S^n to n+1-dimensional loop on S^{n+1}



$\tau_n(S^n) = \tau_n(\Omega(S^{n+1}))$

n-truncation of S^n is the type of “codes” for loops on S^{n+1}

- Decode: promote n-dimensional loop on S^n to n+1-dimensional loop on S^{n+1}



- Encode: define fibration $\texttt{Code}(x{:}S^{n+1})$ with
  $\texttt{Code}(\texttt{base}_{n+1}) := \tau_n(S^n)$
  $\texttt{Code}(\texttt{loop}_{n+1}) :=$ equivalence $\tau_n(S^n) \simeq \tau_n(S^n)$
  “rotating by $\texttt{loop}_n$”

$\pi_2(S^2)$: Hopf fibration