

# **The Seifert-van Kampen Theorem in Homotopy Type Theory**

[ CSL 2016 ]

\* **Favonia**, Carnegie Mellon University, USA

**Michael Shulman**, University of San Diego, USA

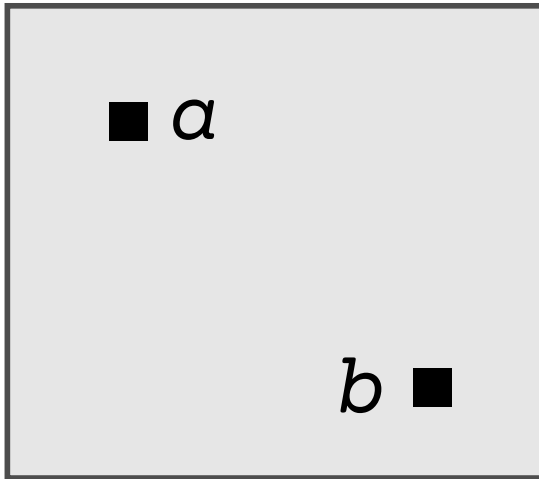
# Homotopy Type Theory

Do homotopy theory in type theory

Hopf fibrations, Eilenberg-Mac Lane spaces, homotopy groups of spheres, Mayer-Vietoris sequences, Blakers-Massey... [HoTT book; Cavallo 14; Hou (Favonia), Finster, Licata & Lumsdaine 16; ...]

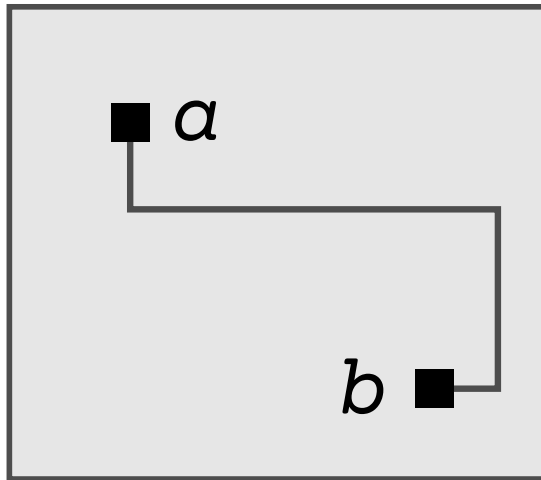
1. Mechanization
2. Translations to other models  
synthetic homotopy theory

# Every type is an $\infty$ -groupoid



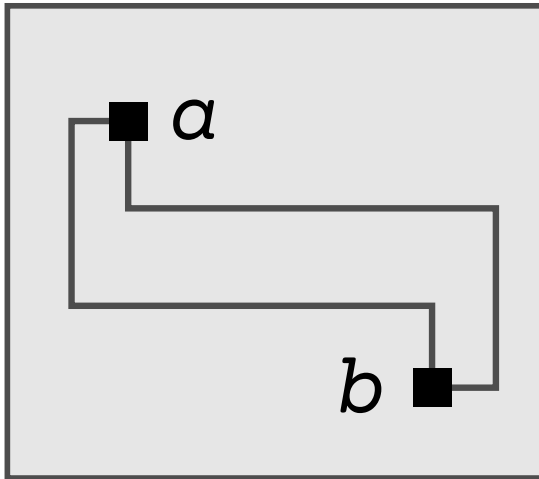
■ terms

# Every type is an $\infty$ -groupoid



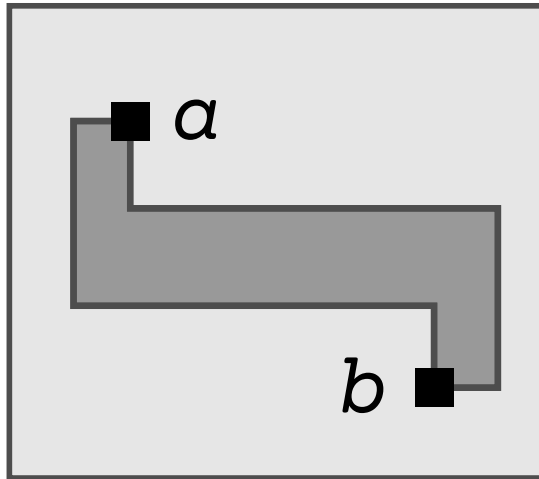
■ terms  
— paths

# Every type is an $\infty$ -groupoid



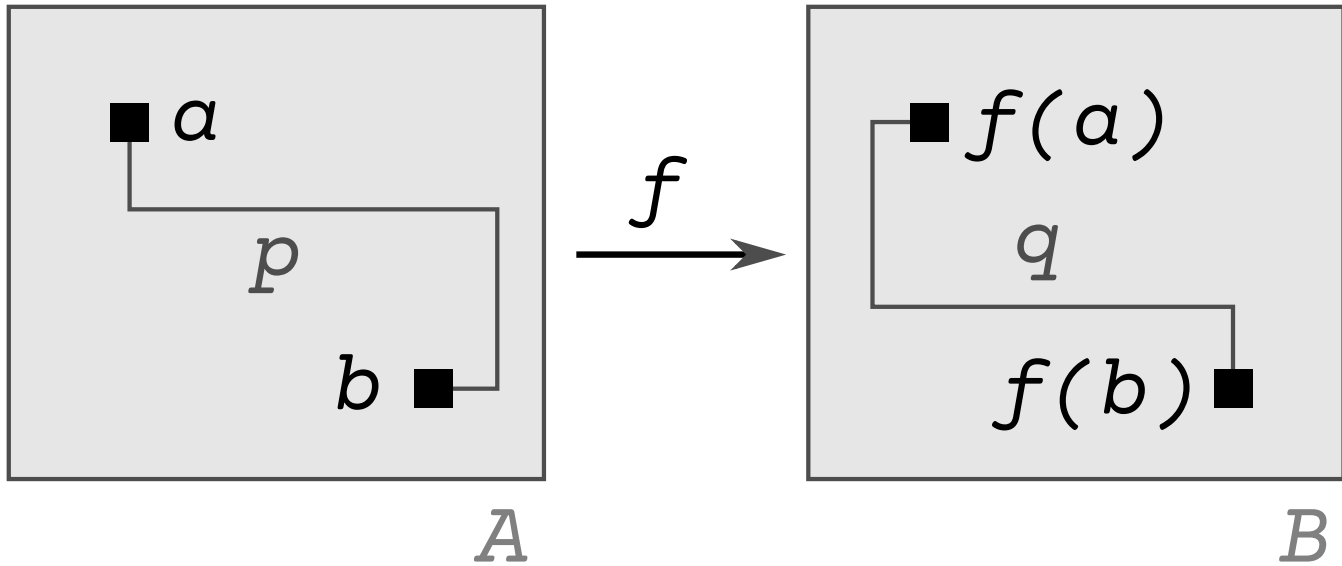
■ terms  
— paths

# Every type is an $\infty$ -groupoid



- terms
- paths
- paths of paths

# Every function is a functor



# Types and Spaces

$A$	Type	Space
$a : A$	Term	Point
$f : A \rightarrow B$	Function	Continuous Mapping
$C : A \rightarrow \text{Type}$	Dependent Type	Fibration
$C(a)$		Fiber
$p : a =_A b$	Identification	Path



[ subject of study ]

**Fundamental groups of pushouts**

[ subject of study ]

## **Fundamental groups of pushouts**

sets of loops  
at some point

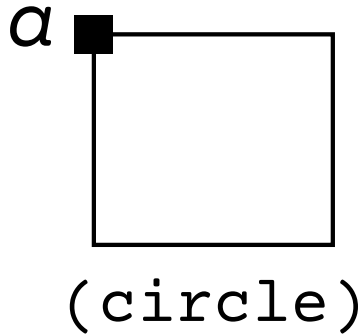
[ subject of study ]

**Fundamental groups of pushouts**

sets of loops  
at some point

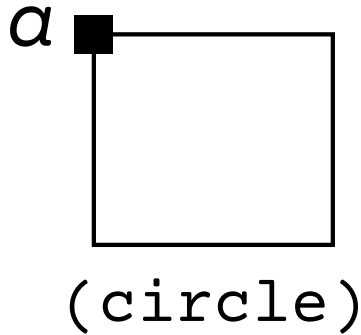
two spaces  
glued together

# Fundamental Group




Ways to travel  
from  $a$  to  $a$

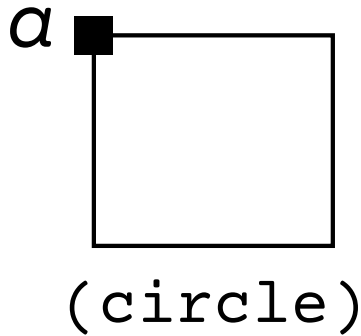
# Fundamental Group



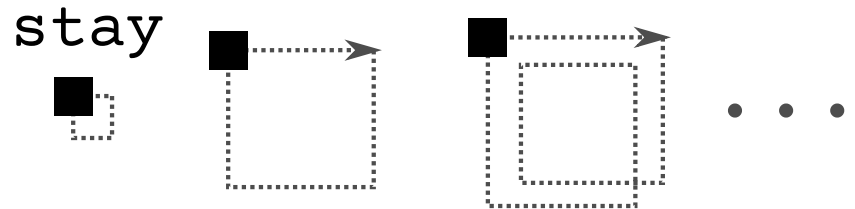
Ways to travel  
from  $a$  to  $a$

stay  


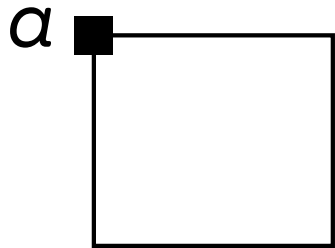
# Fundamental Group



Ways to travel  
from  $a$  to  $a$

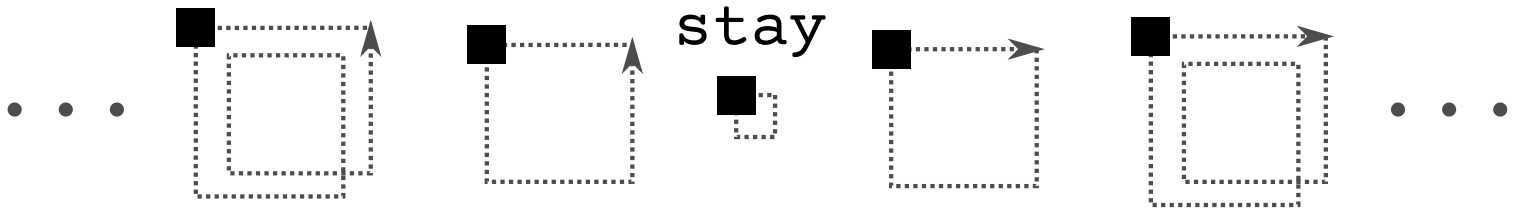


# Fundamental Group

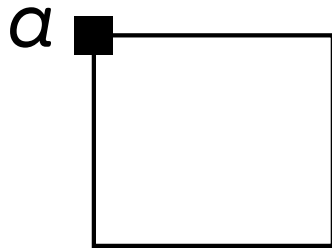


(circle)

Ways to travel  
from  $a$  to  $a$

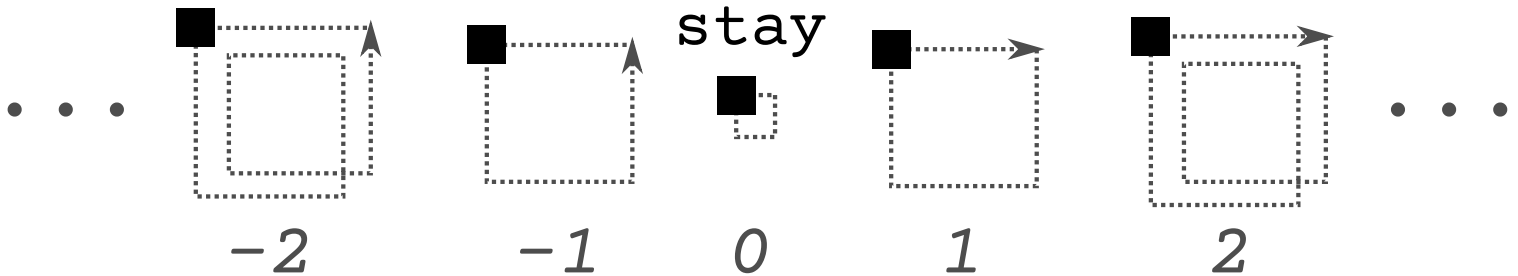


# Fundamental Group



(circle)

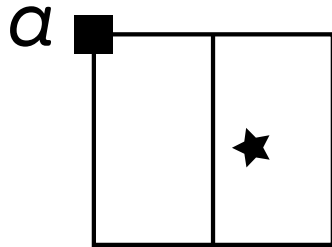
Ways to travel  
from  $a$  to  $a$



Here they correspond to integers



# Fundamental Group

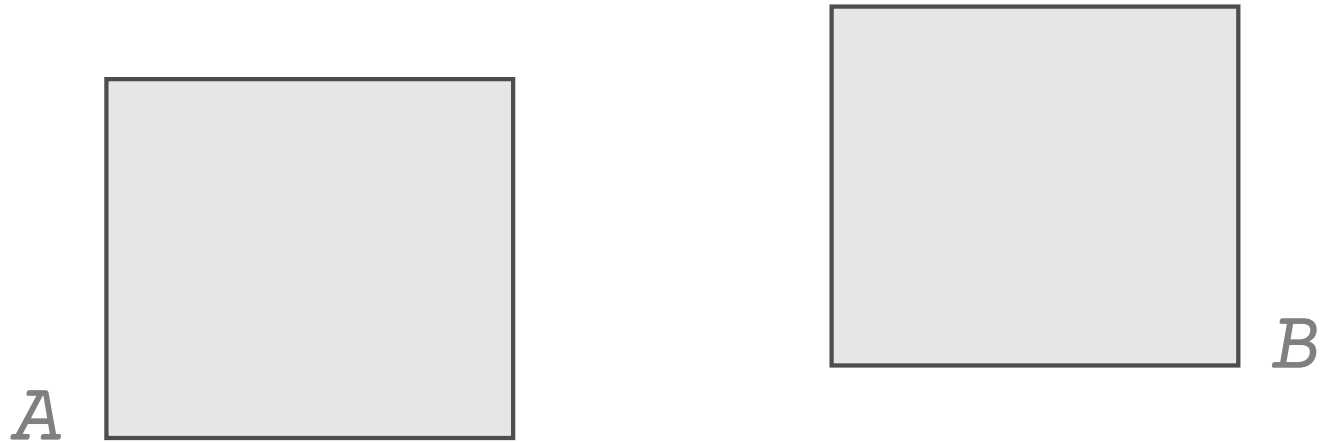


Ways to travel  
from  $a$  to  $a$

Much more if a new  
path ( $\star$ ) is added

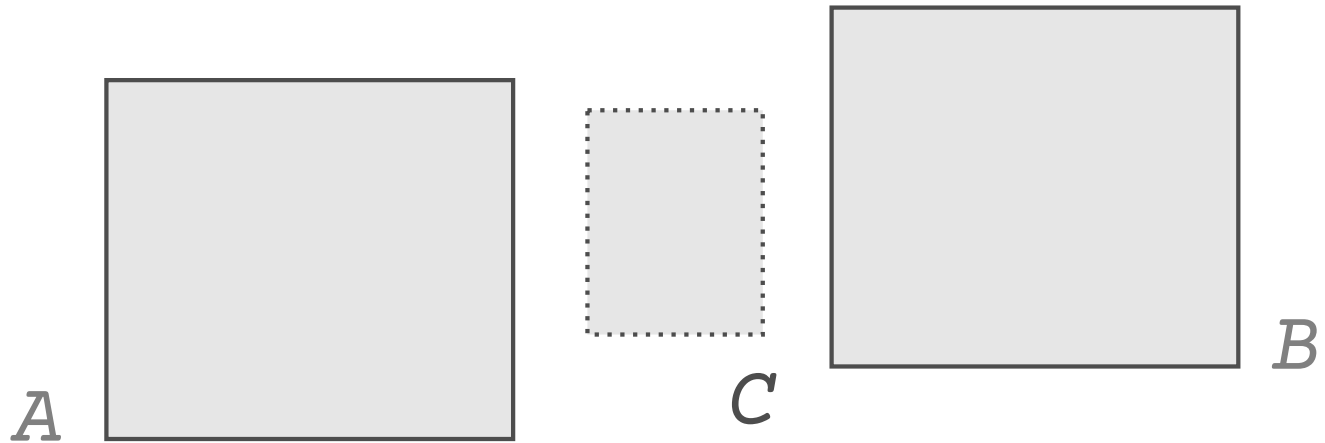
# Pushout

two spaces glued together



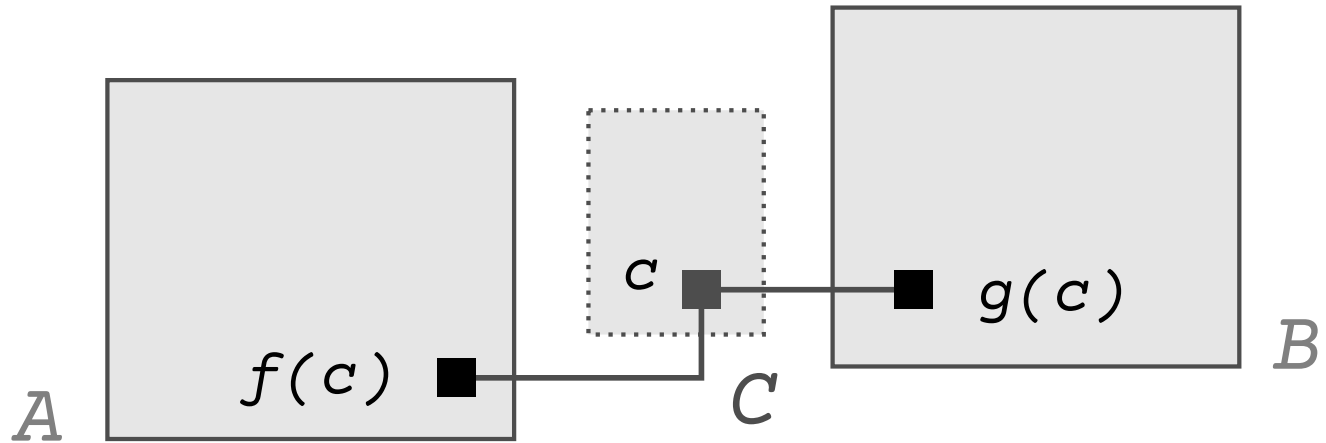
# Pushout

two spaces glued together



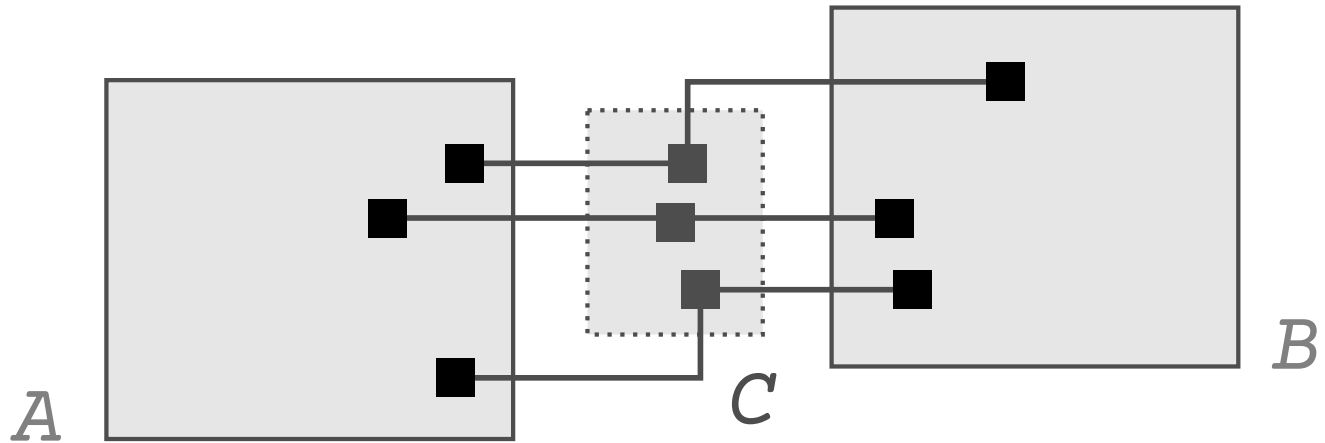
# Pushout

two spaces glued together



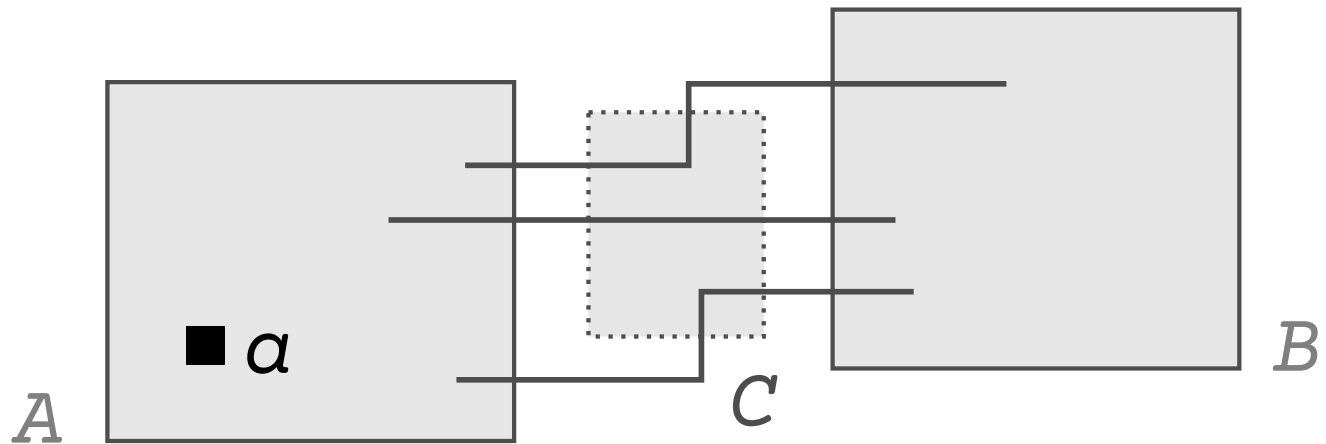
# Pushout

two spaces glued together



# Pushout

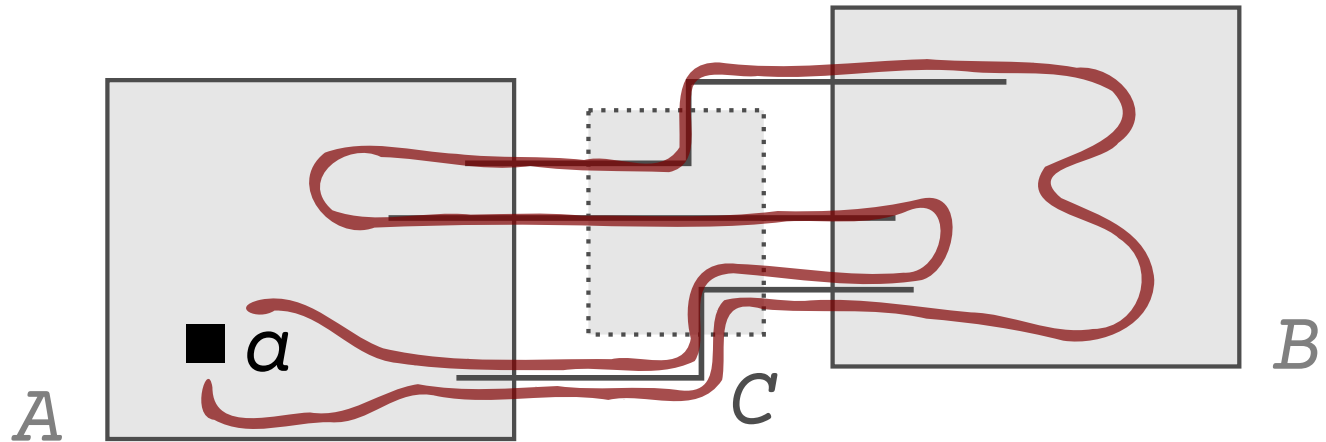
two spaces glued together



ways to travel from  $a$  to  $a$ ?

# Pushout

two spaces glued together



ways to travel from  $a$  to  $a$ ?  
alternative paths in  $A$  and  $B$ !

# Theorem (drafted)

for any  $A, B, C, f$  and  $g,$

$\text{fund-group}(\text{pushout})$

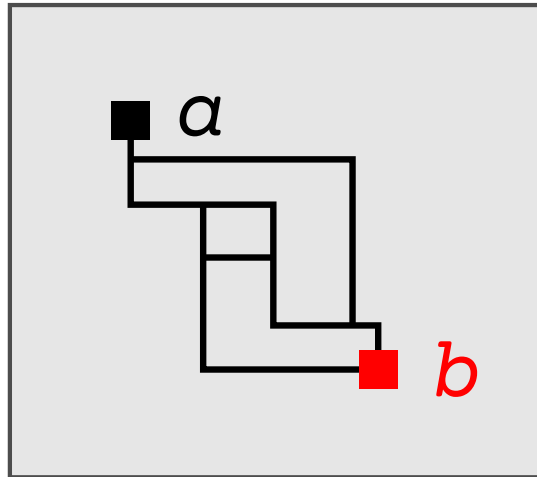
$$\sim = ?(??(A), ??(B), C)$$

??: paths between any two points

?: "seqs of alternative elems"



# Fundamental Groupoid



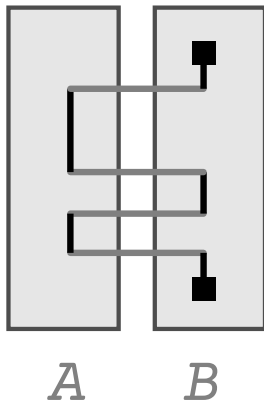
Ways to travel from  $a$  to  $b$

# Theorem (revised)

for any  $A, B, C, f$  and  $g$ ,  
 $\text{fund-groupoid}(\text{pushout})$   
 $\sim = ?(\text{fund-groupoid}(A),$   
 $\text{fund-groupoid}(B), C)$

?: "seqs of alternative elems"

# Alternative Sequences



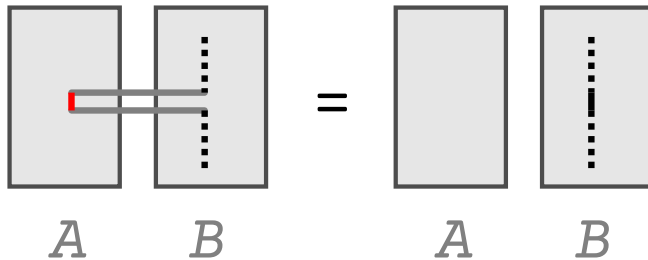
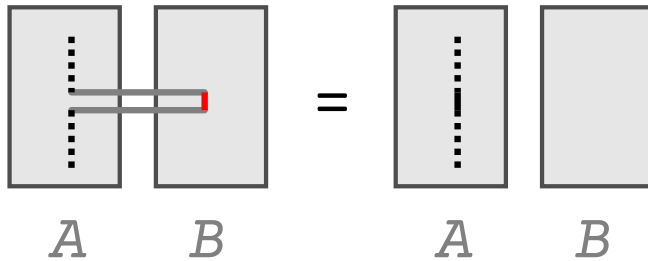
$[p_1, p_2, \dots, p_n]$

consider four cases:

A to A, A to B,

B to A, B to B

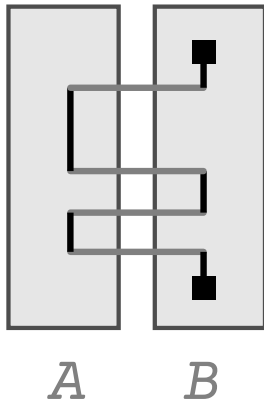
# Alternative Sequences



**quotients** by  
squashing  
superfluous  
trivial paths

going back immediately = not going at all

# Alternative Sequences



$[p_1, p_2, \dots, p_n]$

consider four cases:

A to A, A to B,

B to A, B to B

each case is a quotient  
of alternative sequences

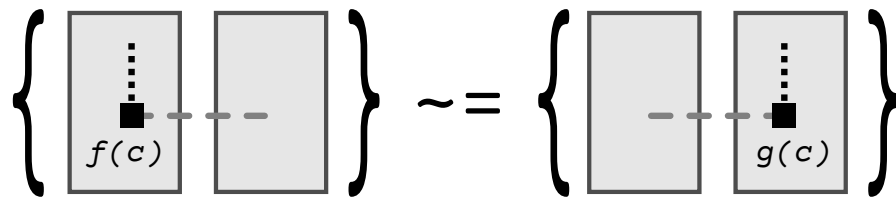
# Alternative Sequences

next: unify four cases into  
one type family "alt-seq"

# Alternative Sequences

next: unify four cases into  
one type family "alt-seq"

show that it respects bridges, ex:

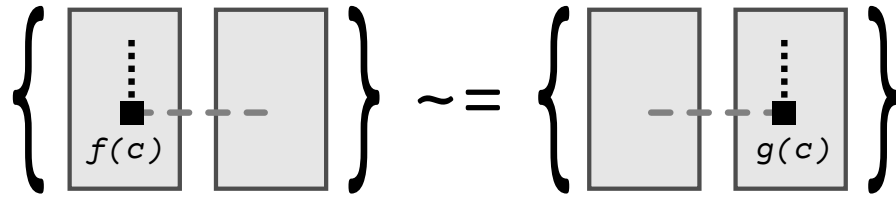


alt-seq a (f c)  $\sim =$  alt-seq a (g c)

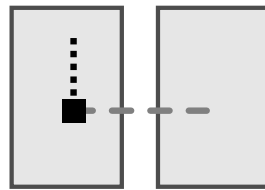
# **Recipe of Equivalences**

- \* two functions back and forth
- \* round-trips are identity

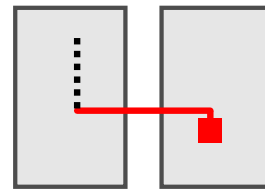




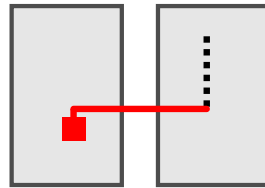
$$\left\{ \begin{array}{|c|c|} \hline \vdots \\ \hline \blacksquare \\ \hline f(c) \\ \hline \end{array} \quad \begin{array}{|c|} \hline \vdots \\ \hline \vdots \\ \hline \end{array} \right\} \sim = \left\{ \begin{array}{|c|} \hline \vdots \\ \hline \vdots \\ \hline \vdots \\ \hline \end{array} \quad \begin{array}{|c|} \hline \vdots \\ \hline \blacksquare \\ \hline g(c) \\ \hline \end{array} \right\}$$



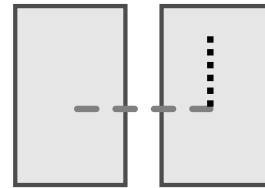
[..., p]



[..., p, trivial]

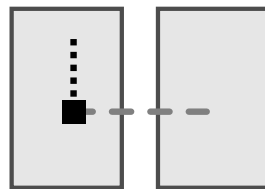


[..., p, trivial]

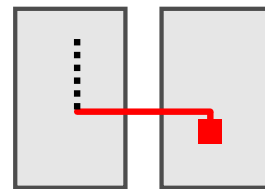
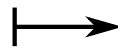


[..., p]

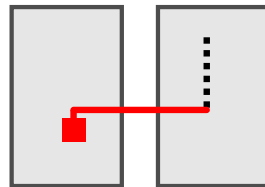
$$\left\{ \begin{array}{|c|c|} \hline \vdots \\ \hline \blacksquare \\ \hline f(c) \\ \hline \end{array} \quad \begin{array}{|c|c|} \hline \vdots \\ \hline \vdots \\ \hline \vdots \\ \hline \end{array} \right\} \sim = \left\{ \begin{array}{|c|c|} \hline \vdots \\ \hline \vdots \\ \hline \vdots \\ \hline g(c) \\ \hline \end{array} \quad \begin{array}{|c|c|} \hline \vdots \\ \hline \vdots \\ \hline \vdots \\ \hline \end{array} \right\}$$



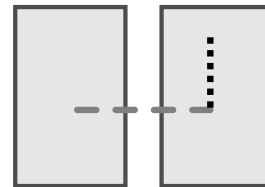
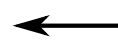
[..., p]



[..., p, trivial]



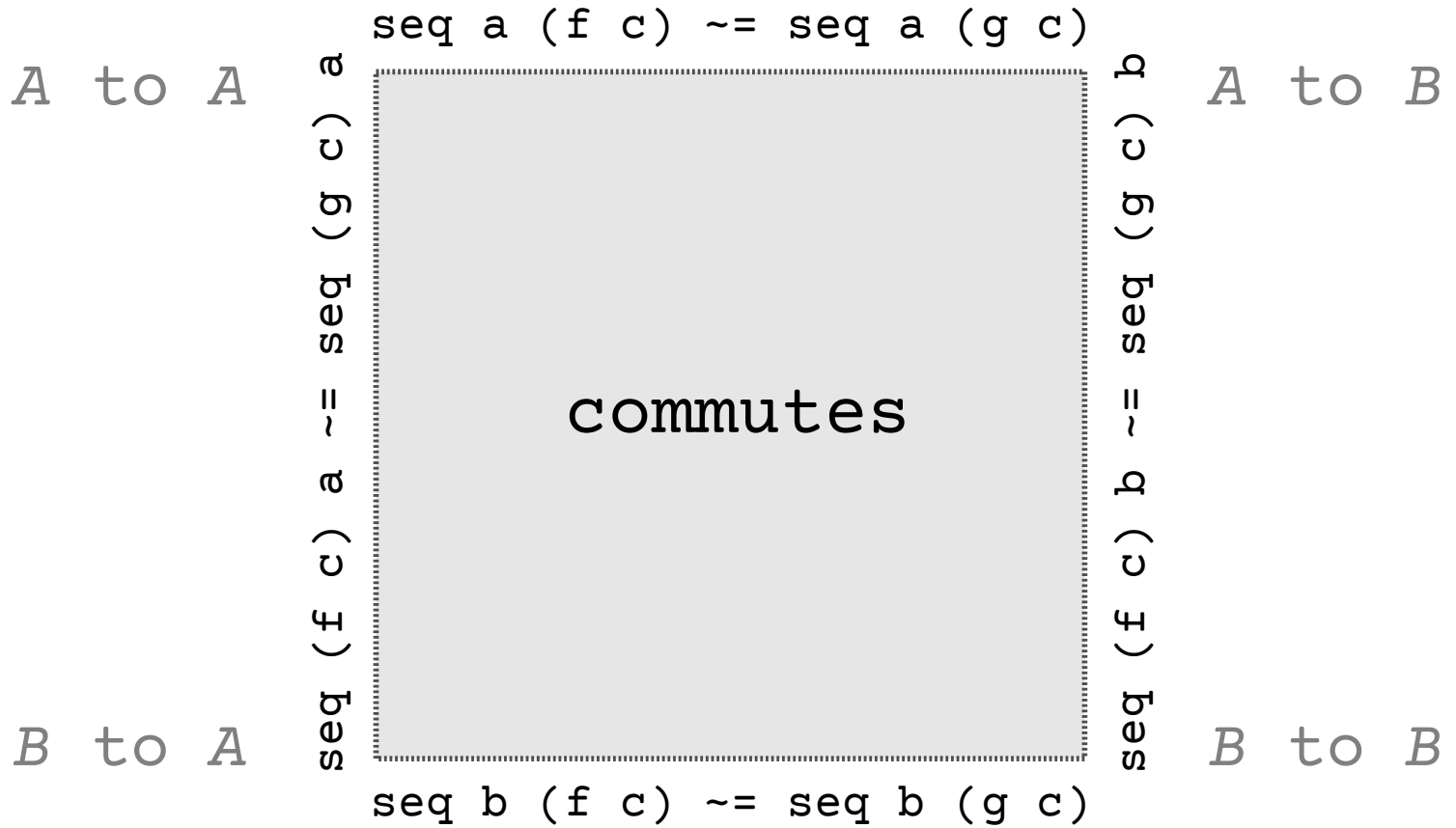
[..., p, trivial]



[..., p]

round-trips are identity due to quotient relation (squashing trivials)

# Alternative Sequences



# Theorem (final)

```
for any A, B, C, f and g,  
fund-groupoid(pushout)  
  ~ = alt-seq(fund-groupoid(A),  
              fund-groupoid(B), C)
```

(zero pages left before the proofs)

**fund-groupoid**  $\xrightarrow{\text{encode}}$  **alt-seqs**  
(all paths)

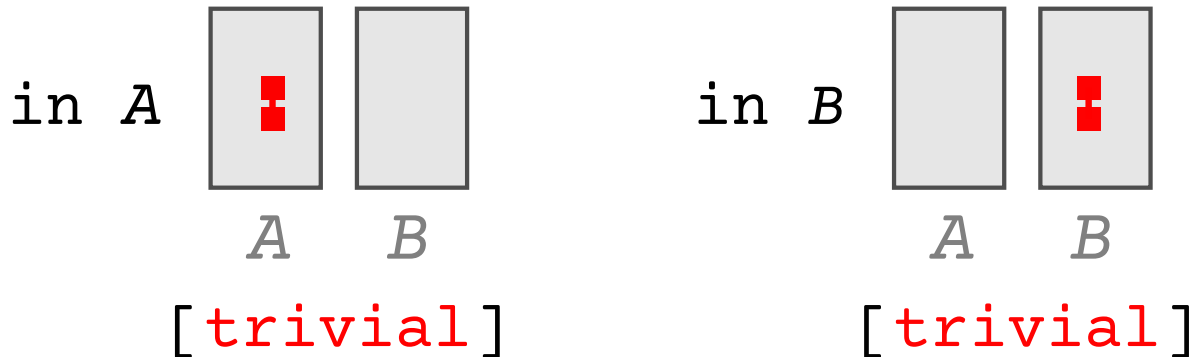
**fund-groupoid**  $\xrightarrow{\text{encode}}$  **alt-seqs**  
(all paths)

**path induction principle:**  
consider only trivial paths

for any point  $p$  in pushout  
find an alt-seq from  $p$  to  $p$   
representing the trivial path at  $p$

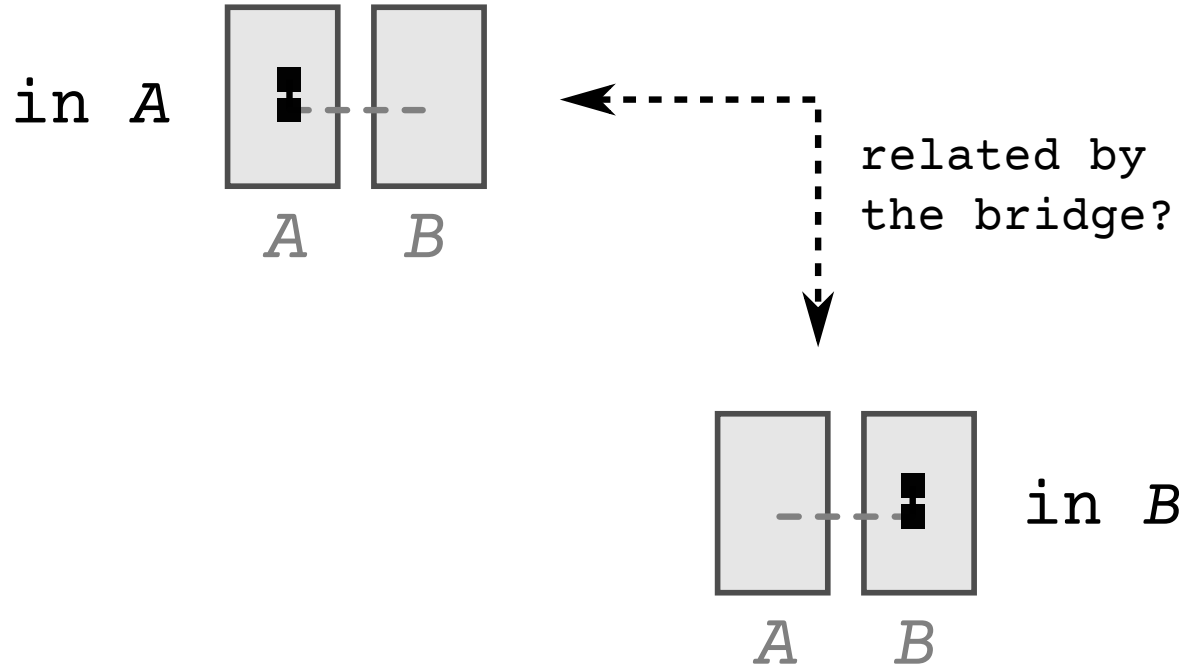
**fund-groupoid**  $\xrightarrow{\text{encode}}$  **alt-seqs**  
(all paths)

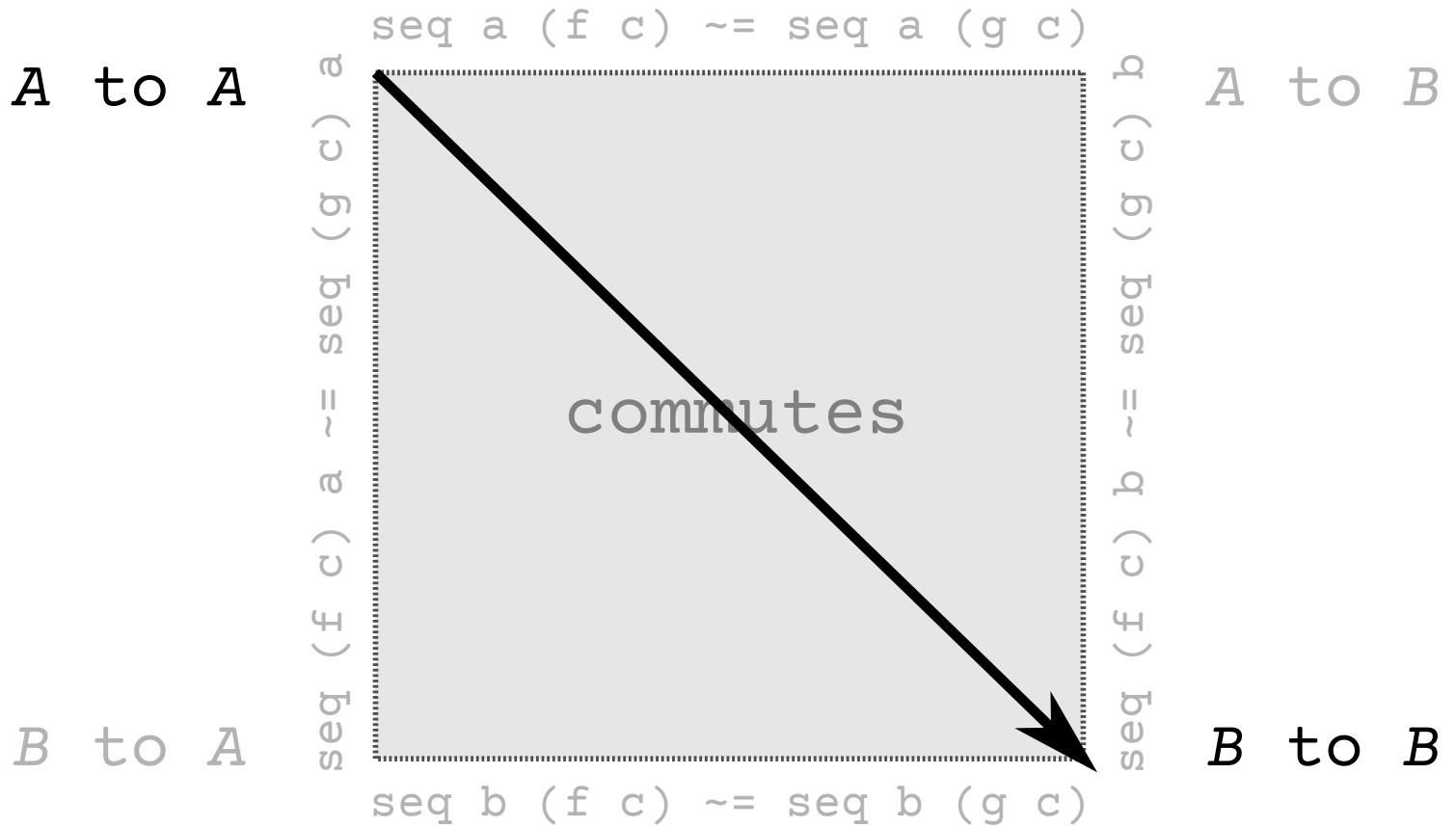
**path induction principle:**  
consider only trivial paths

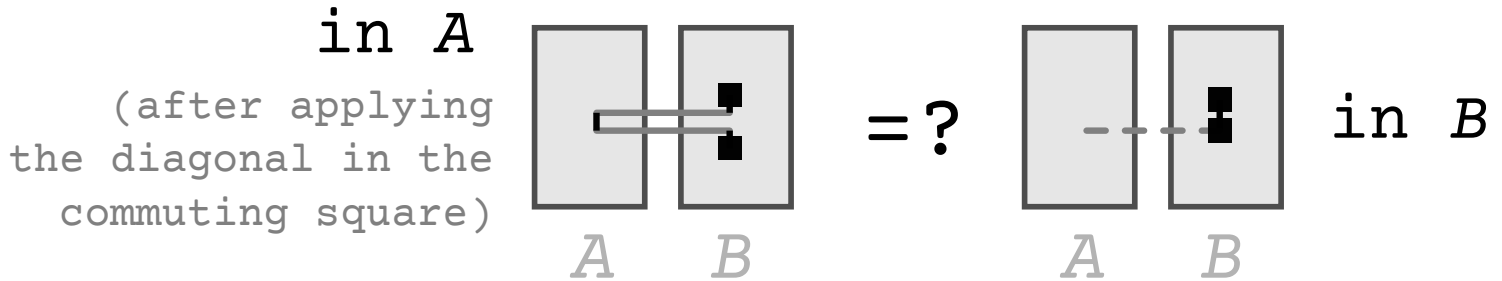


next: respecting bridges









witnessed by the quotient relation  
(squashing trivials)

**alt-seq**  $\xrightarrow{\text{decode}}$  **fund-groupoid**

just compositions!

**alt-seq**  $\xrightarrow{\text{decode}}$  **fund-groupoid**

just compositions!

**grp**  $\xrightarrow{\text{encode}}$  **seqs**  $\xrightarrow{\text{decode}}$  **grp**

again by path induction  
(similar to "encode")

**alt-seq**  $\xrightarrow{\text{decode}}$  **fund-groupoid**

just compositions!

**grp**d  $\xrightarrow{\text{encode}}$  **seqs**  $\xrightarrow{\text{decode}}$  **grp**d

again by path induction  
(similar to "encode")

**seqs**  $\xrightarrow{\text{decode}}$  **grp**d  $\xrightarrow{\text{encode}}$  **seqs**

induction on sequences

lemma:  $\text{encode}(\text{decode}[p_1, p_2, \dots])$   
 $= p_1 :: \text{encode}(\text{decode}[p_2, \dots])$

# Seifert-van Kampen

```
for any A, B, C, f and g,  
fund-groupoid(pushout)  
  ~ = alt-seq(fund-groupoid(A),  
              fund-groupoid(B), C)
```

# Final Notes

- \* Refined version: Can focus on just the set of base points of  $C$  covering its components.
- \* All mechanized in Agda  
[github.com/HoTT/HoTT-Agda/blob/1.0/Homotopy/VanKampen.agda](https://github.com/HoTT/HoTT-Agda/blob/1.0/Homotopy/VanKampen.agda)