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

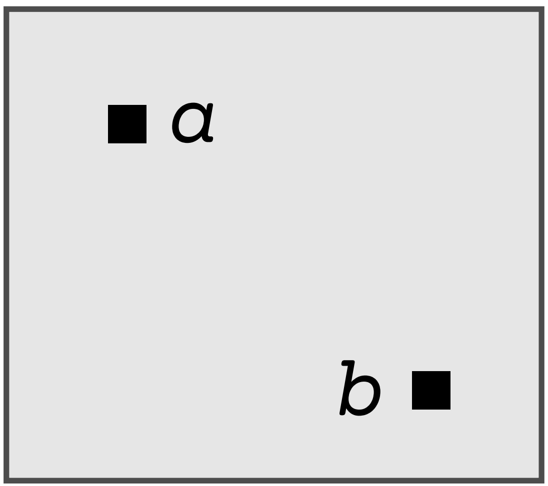
Do homotopy theory in type theory

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

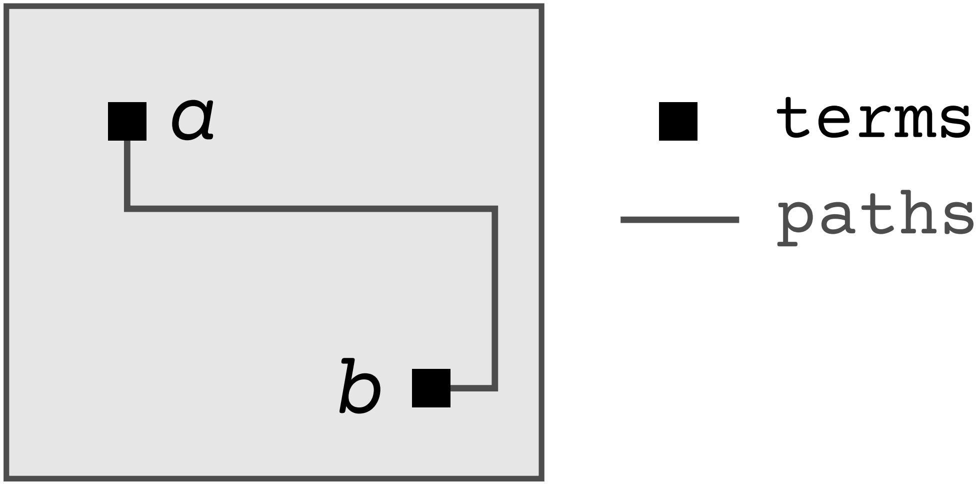
   synthetic homotopy theory

Every type is an ∞-groupoid

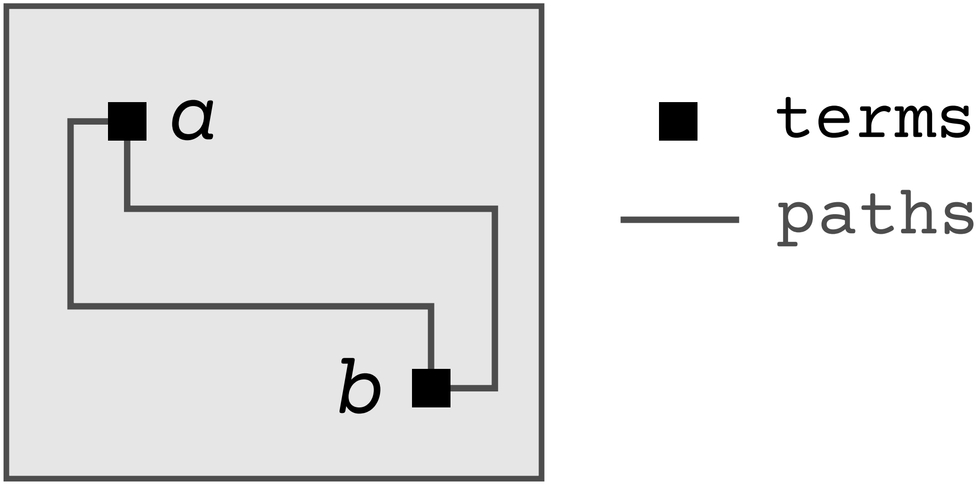


■ terms

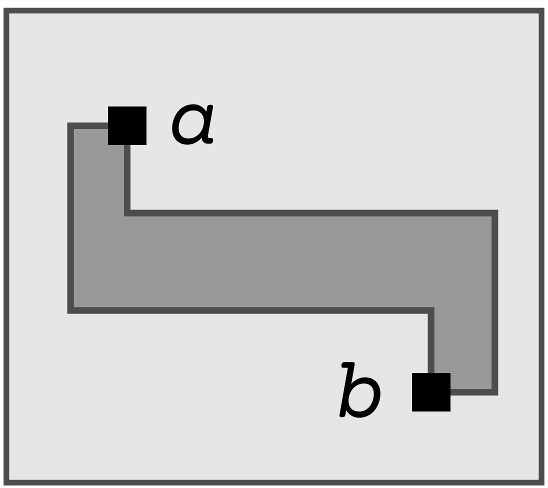
Every type is an ∞-groupoid



Every type is an ∞-groupoid



Every type is an ∞-groupoid

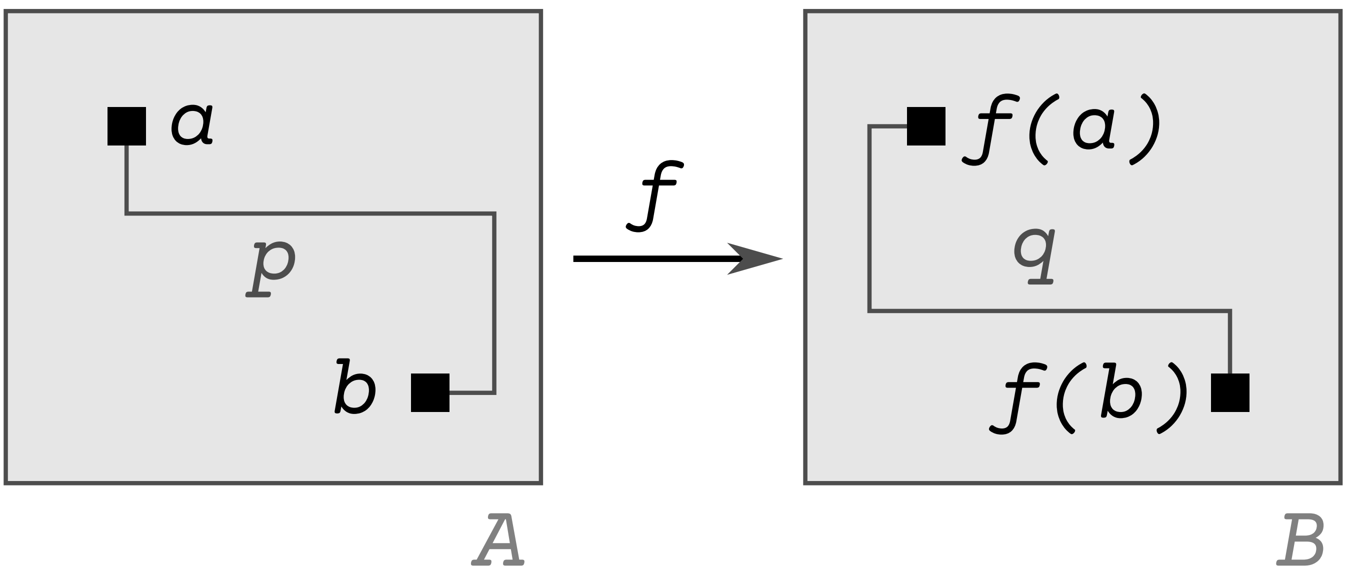


terms

paths

paths of paths

Every function is a functor



Types and Spaces

A	Type	Space
$a : A$	Term	Point
$f : A \to B$	Function	Continuous Mapping
$C : A \to Type$	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

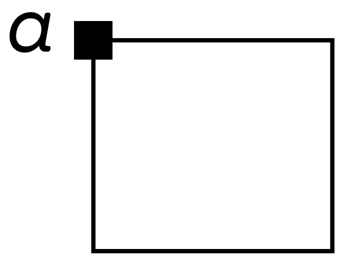
[subject of study]

Fundamental groups of pushouts

sets of loops at some point

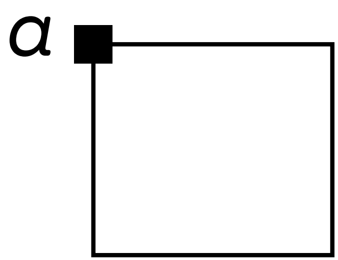
two spaces glued together

Fundamental Group



Ways to travel
from a to a

Fundamental Group

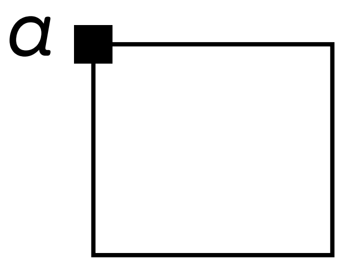


(circle)

Ways to travel from a to a

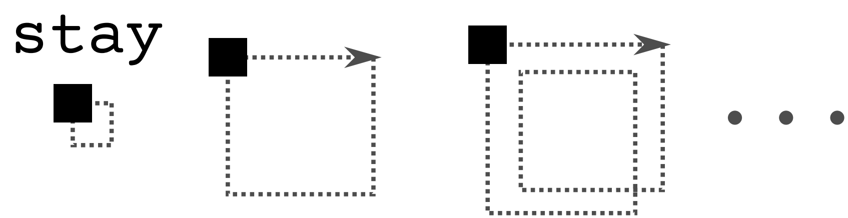
stay

Fundamental Group

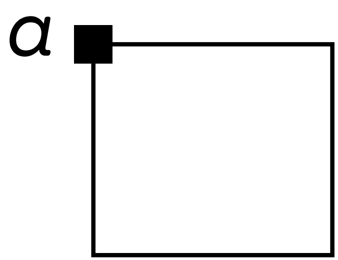


(circle)

Ways to travel from α to α

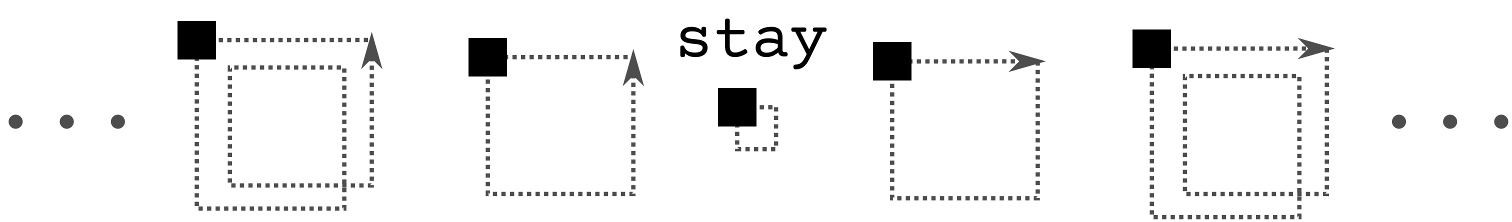


Fundamental Group

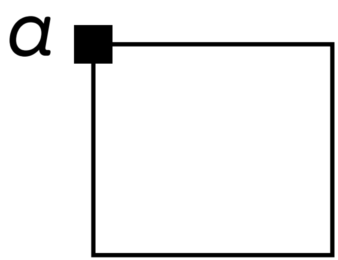


(circle)

Ways to travel from a to a

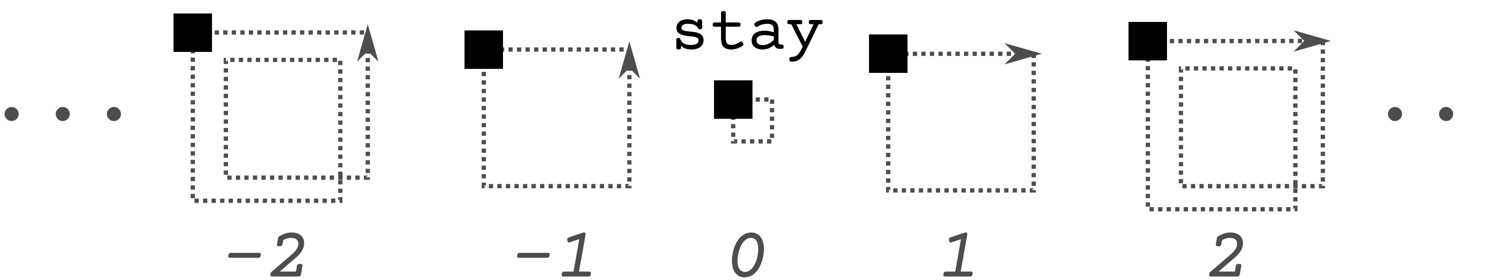


Fundamental Group



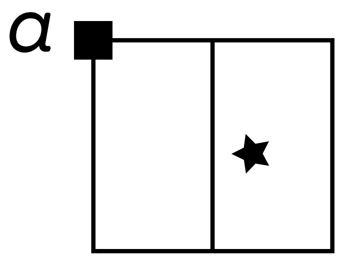
(circle)

Ways to travel from a to a



Here they correspond to integers

Fundamental Group

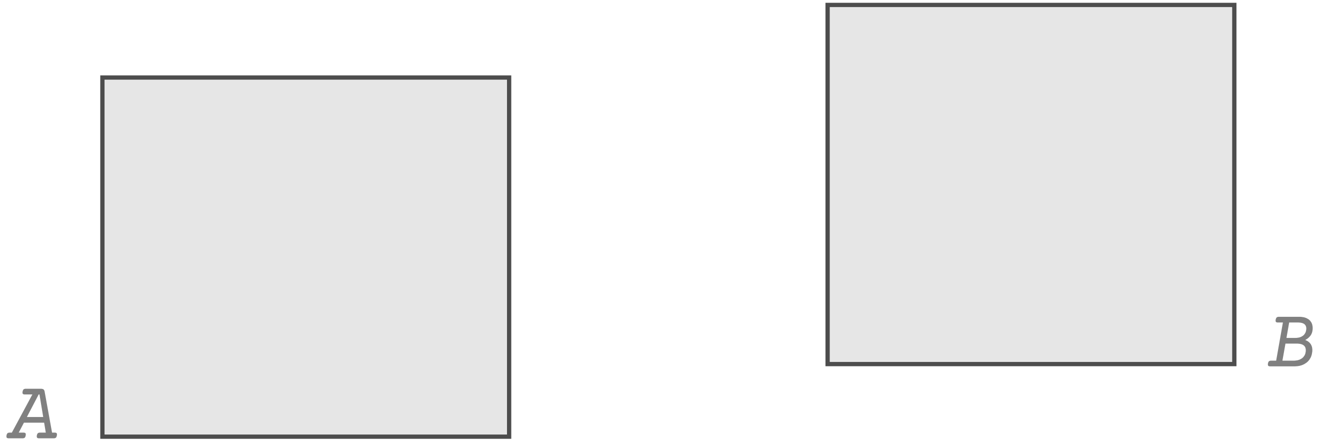


Ways to travel from a to a

Much more if a new path (★) is added

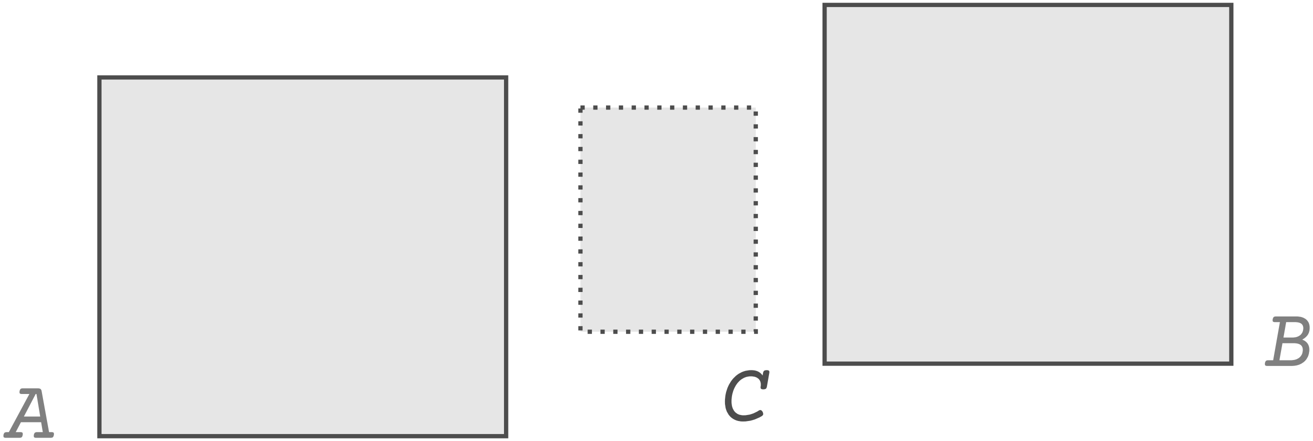
Pushout

two spaces glued together



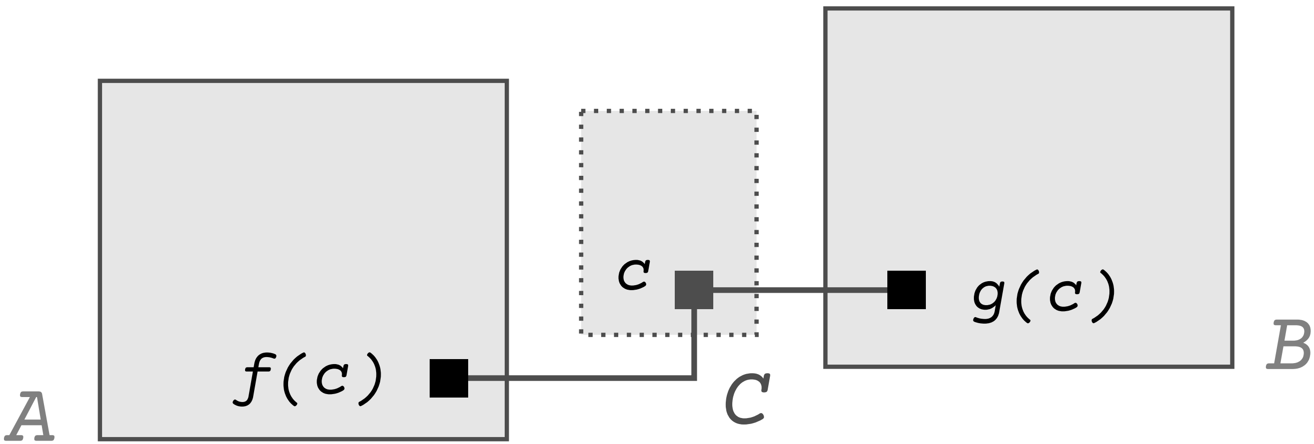
Pushout

two spaces glued together



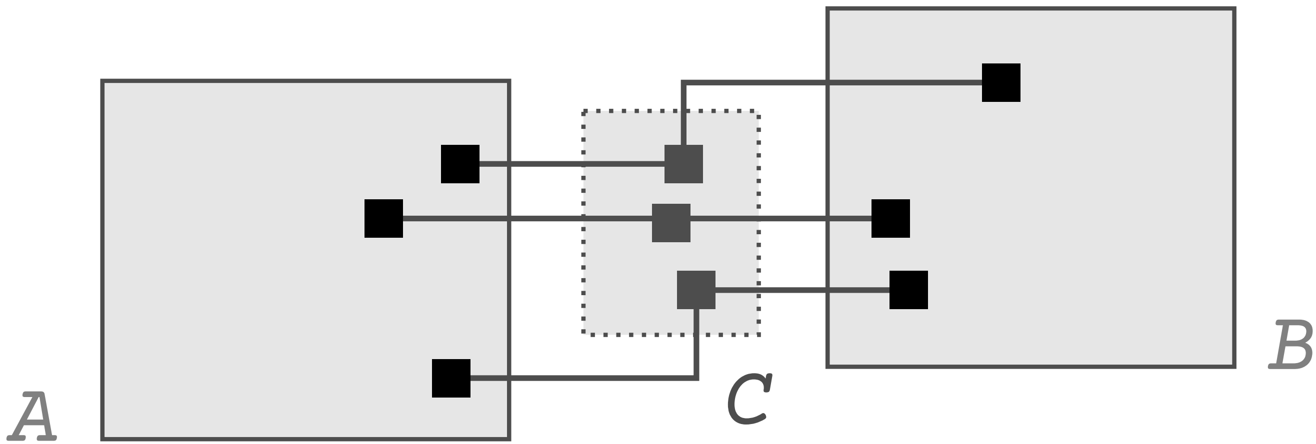
Pushout

two spaces glued together



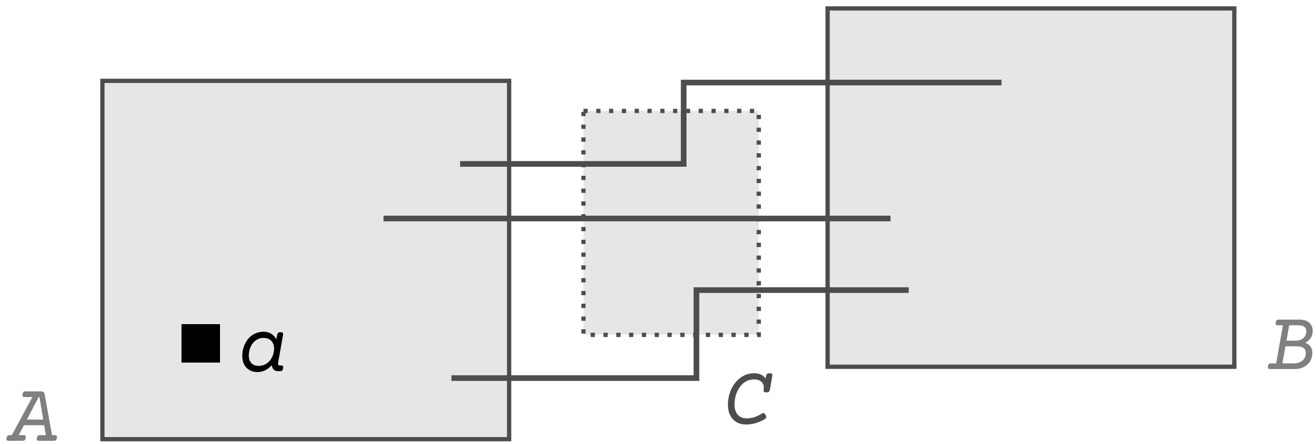
Pushout

two spaces glued together



Pushout

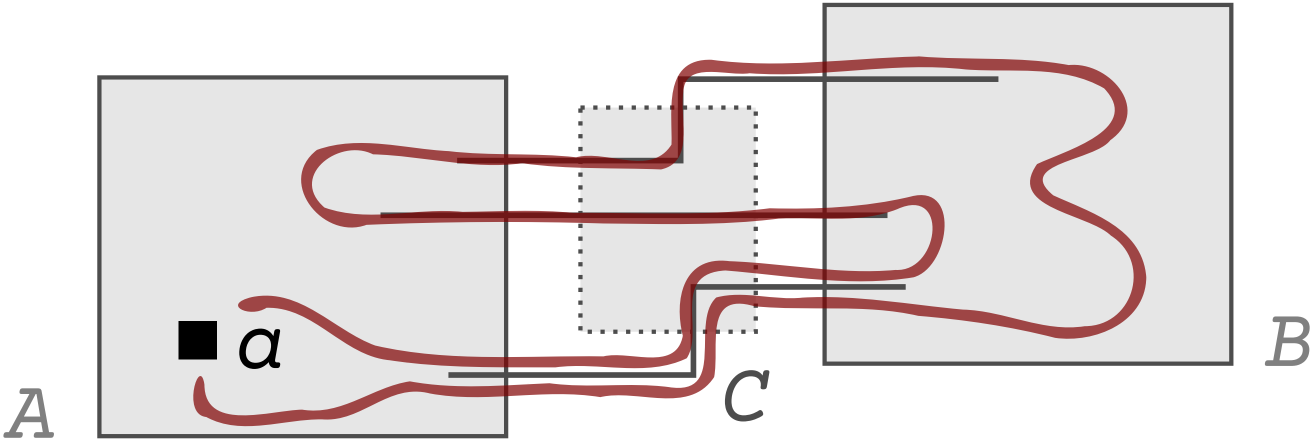
two spaces glued together



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!

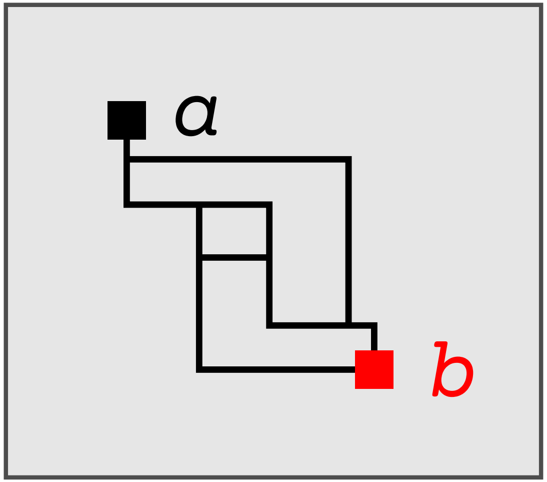
Theorem (drafted)

```
for any A, B, C, f and g,
fund-group(pushout)
        ~= ?(??(A), ??(B), C)
```

??: paths between any two points

?: "seqs of alternative elems"

Fundamental Groupoid



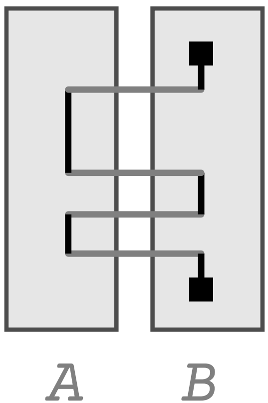
Ways to travel from a to b

Theorem (revised)

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

?: "seqs of alternative elems"

Alternative Sequences



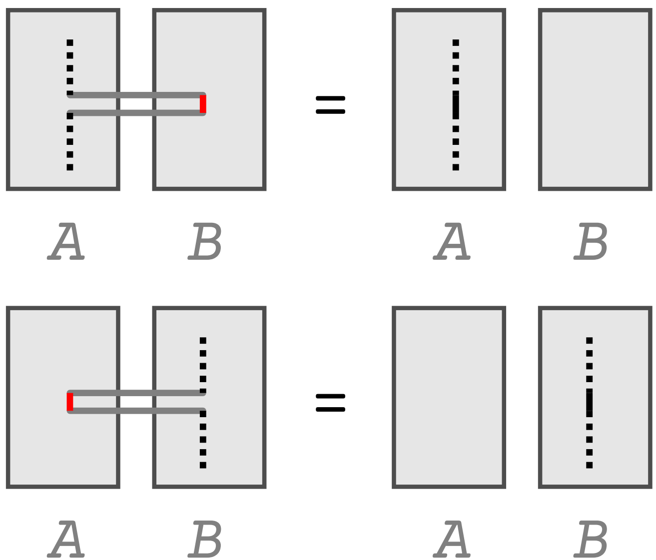
[p1, p2, ..., pn]

consider four cases:

A to *A*, *A* to *B*,

B to *A*, *B* to *B*

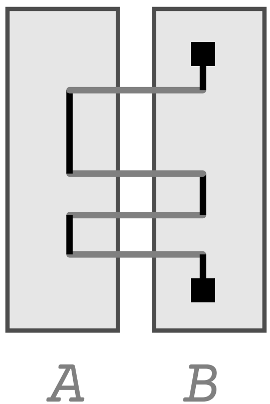
Alternative Sequences



quotients by squashing superfluous trivial paths

going back immediately = not going at all

Alternative Sequences



[p1, p2, ..., pn]

consider four cases:

A to *A*, *A* to *B*,

B to *A*, *B* to *B*

each case is a quotient
of alternative sequences

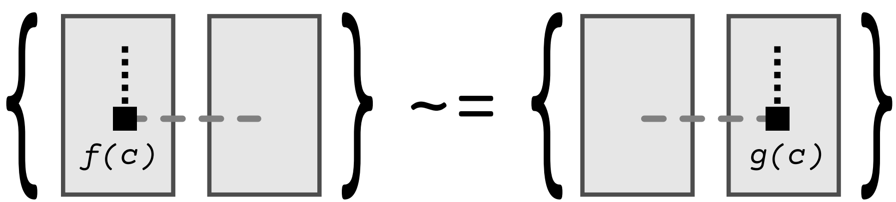
Alternative Sequences

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

Alternative Sequences

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

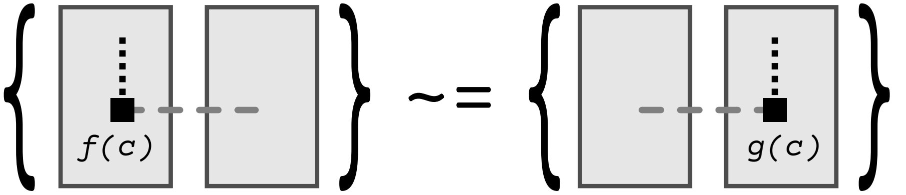
show that it respects bridges, ex:

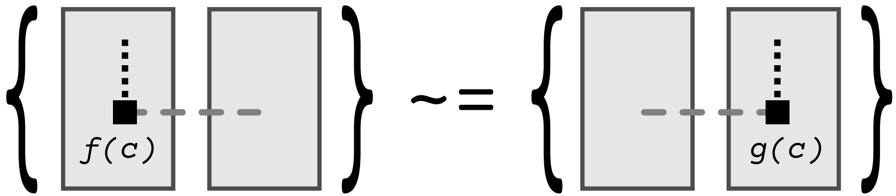


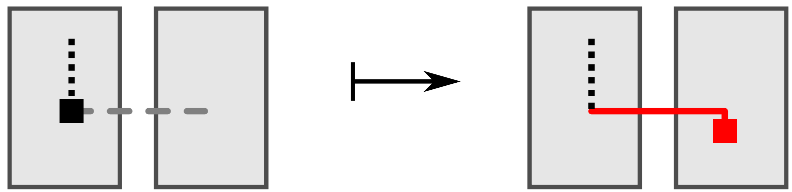
alt-seq a (f c) ~= alt-seq a (g c)

Recipe of Equivalences

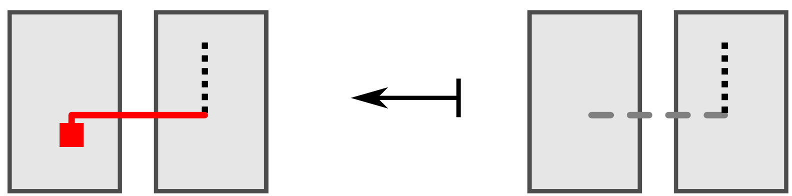
* two functions back and forth
* round-trips are identity


f(c)
~=
g(c)

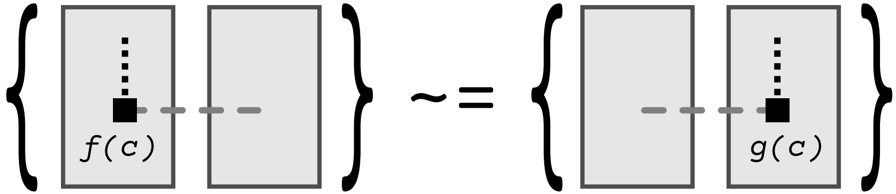


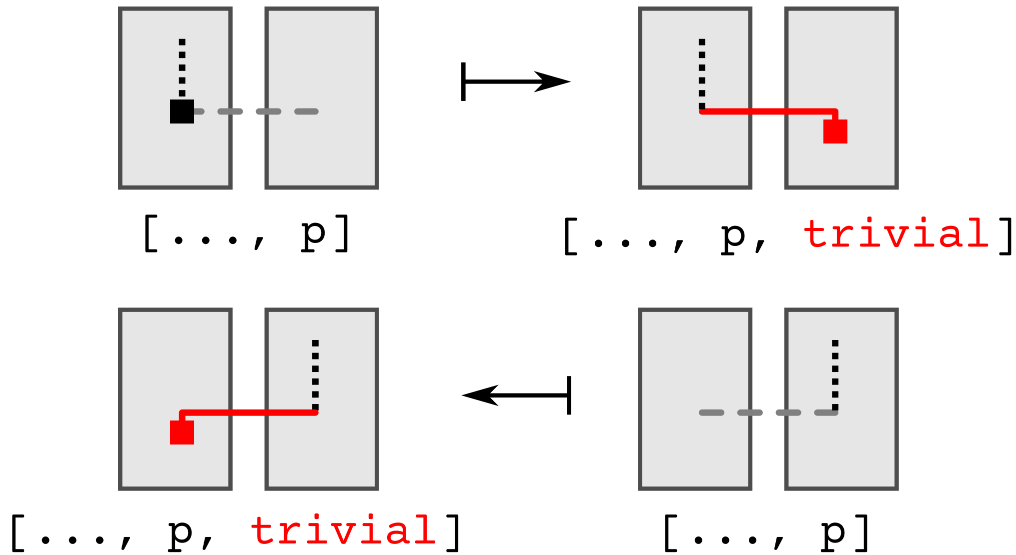


[..., p] [..., p, trivial]



[..., p, trivial] [..., p]





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

Alternative Sequences

`seq a (f c) ~= seq a (g c)`

A to *A*

A to *B*

`seq (f c) a ~= seq (g c) a`

commutes

`seq (f c) b ~= seq (g c) b`

B to *A*

B to *B*

`seq b (f c) ~= seq b (g c)`

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 $\xrightarrow[\texttt{encode}]{}$ alt-seqs

(all paths)

fund-groupoid $\underset{\text{encode}}{\texttt{->}}$ 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*

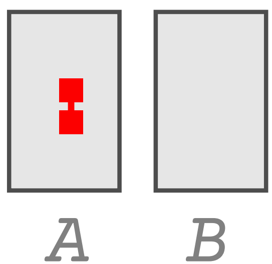
fund-groupoid -> alt-seqs
encode

(all paths)

path induction principle:

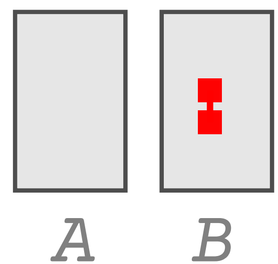
consider only trivial paths

in *A*



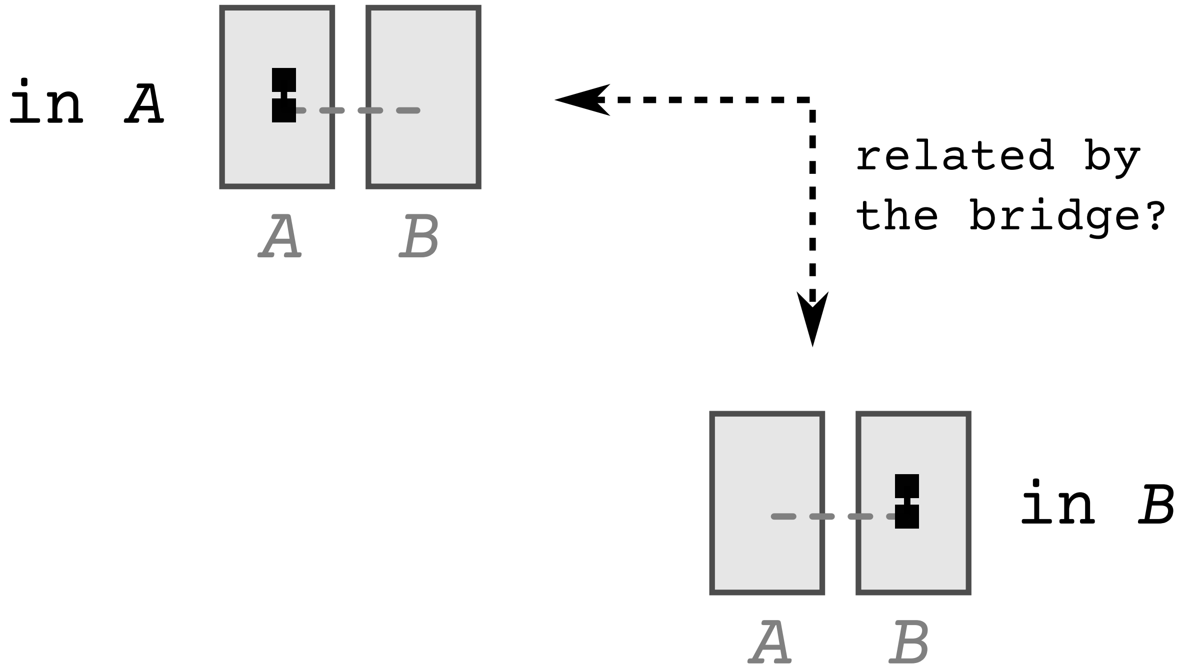
[trivial]

in *B*

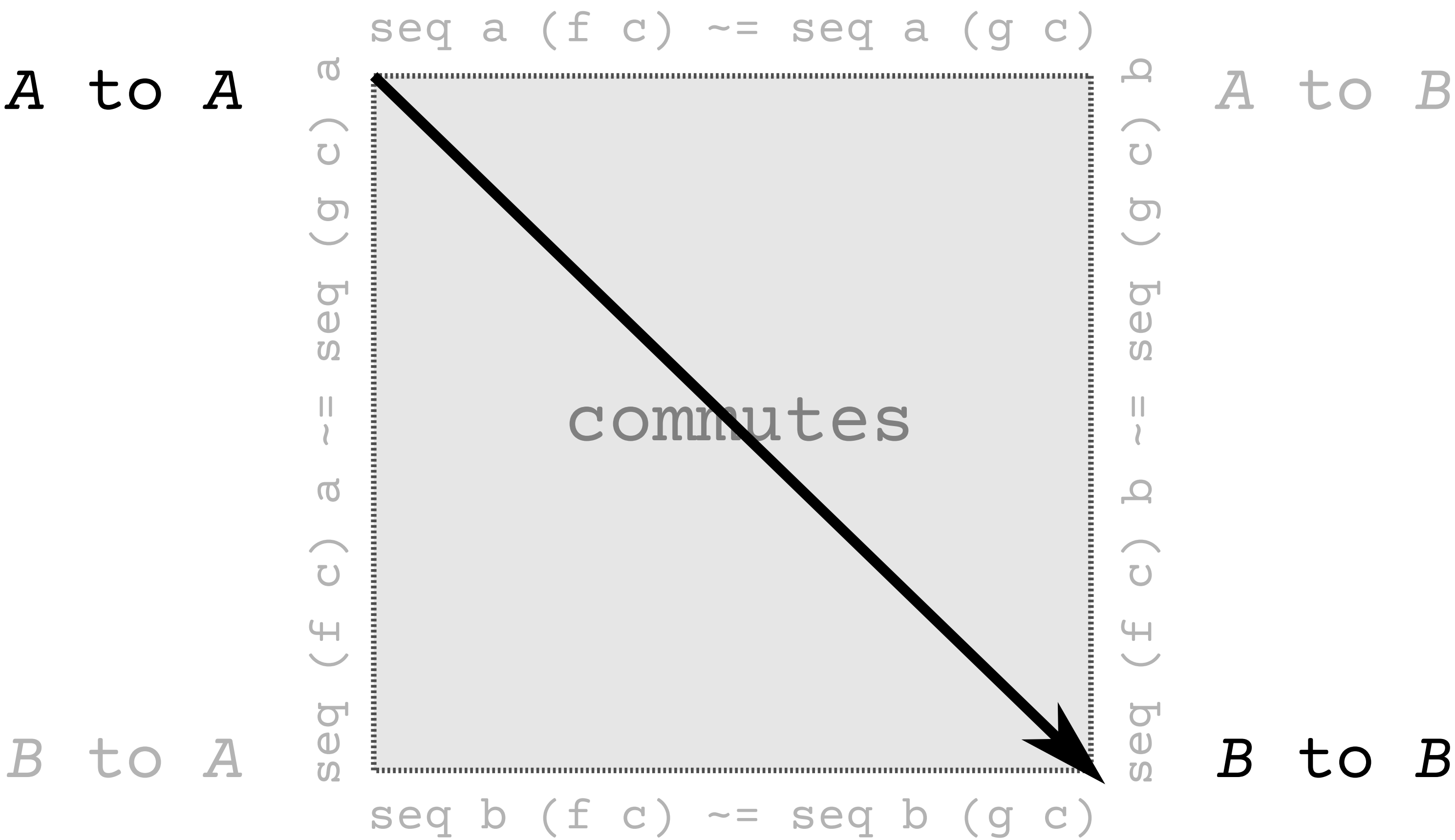


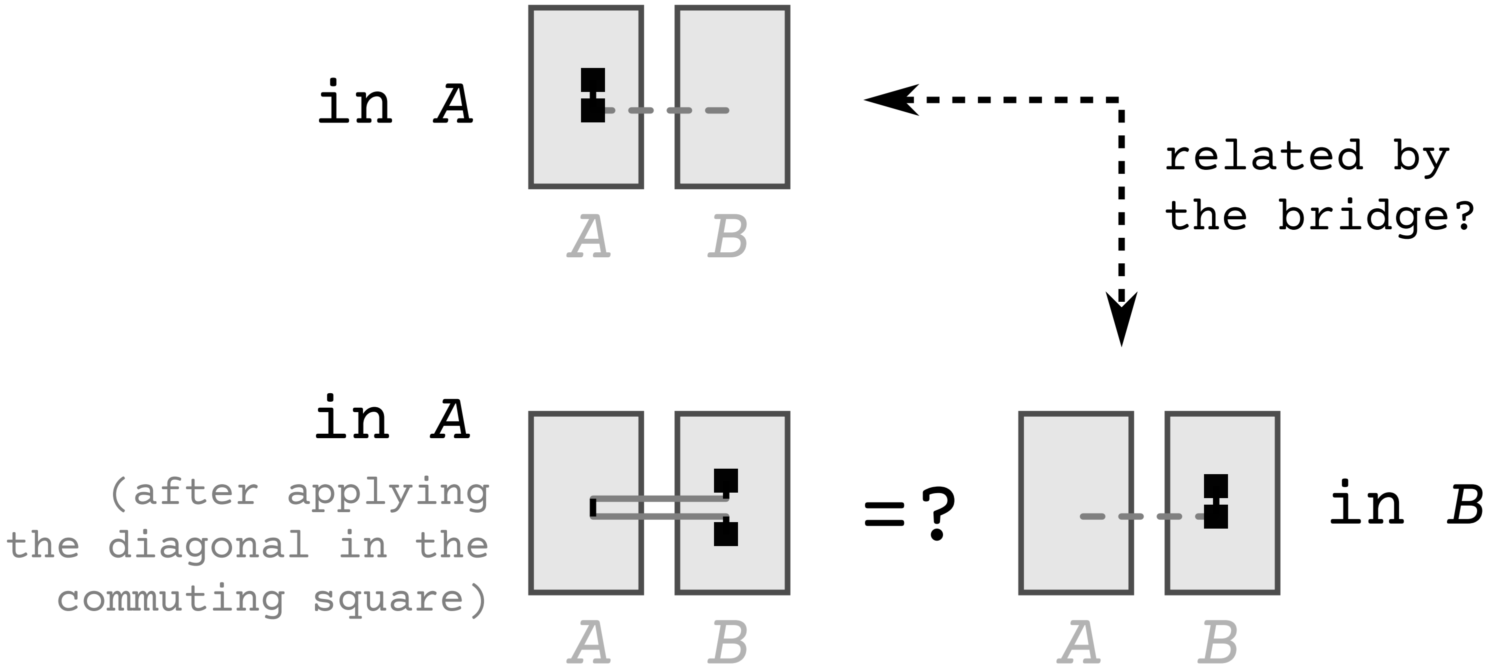
[trivial]

next: respecting bridges



in A
A
B
related by
the bridge?
A
B
in B


seq a (f c) ~= seq a (g c)
A to A
A to B
seq (f c) a ~= seq (g c) a
seq (f c) b ~= seq (g c) b
commutes
B to A
B to B
seq b (f c) ~= seq b (g c)



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!

grpd $\xrightarrow[\text{encode}]{}$ seqs $\xrightarrow[\text{decode}]{}$ grpd

again by path induction
(similar to "encode")

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

just compositions!

grpd $\xrightarrow[\text{encode}]{}$ seqs $\xrightarrow[\text{decode}]{}$ grpd

again by path induction
(similar to "encode")

seqs $\xrightarrow[\text{decode}]{}$ grpd $\xrightarrow[\text{encode}]{}$ seqs

induction on sequences

```
lemma: encode(decode[p1,p2,...])
    = p1 :: encode(decode[p2,...])
```

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