Division by two, omniscience, and homotopy type theory

Samuel Mimram Émile Oleon

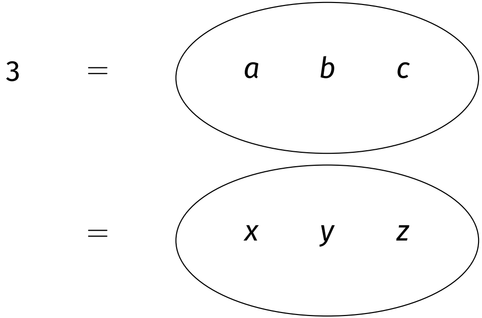
SCALP working group

February 16, 2023

Natural numbers as sets

The **natural numbers** $\mathbb{N}$ can be defined as the equivalence classes of finite sets under isomorphism (= cardinals).

For instance,



Operations on sets

When we have an operation on natural number we can therefore ask:

is the quotient of some operation on sets?

Operations on sets

When we have an operation on natural number we can therefore ask:

is the quotient of some operation on sets?

For instance,

- **addition** is the quotient of disjoint union:

$$3+2 \quad = \quad \{a \ b \ c\} \sqcup \{x \ y\} \quad = \quad \{a \ b \ c \ x \ y\} \quad = \quad 5$$

Operations on sets

When we have an operation on natural number we can therefore ask:

is the quotient of some operation on sets?

For instance,

- **addition** is the quotient of disjoint union:

$$3+2 = \{a \ \ b \ \ c\} \sqcup \{x \ \ y\} = \{a \ \ b \ \ c \ \ x \ \ y\} = 5$$

- **product** is the quotient of cartesian product:

$$3 \times 2 = \{a \ \ b \ \ c\} \times \{x \ \ y\} = \left\{\begin{matrix} (a,x) & (b,x) & (c,x) \\ (a,y) & (b,y) & (c,y) \end{matrix}\right\} = 6$$

Operations on sets

When we have an operation on natural number we can therefore ask:

is the quotient of some operation on sets?

This is satisfactory when it is the case because

- this is more “constructive”: we replace equality by isomorphism,
- we have an extension of the operations to infinite sets,
- we can study which axioms of set theory we need to perform this.

Subtraction by 1

Next interesting operation is subtraction by 1

Subtraction by 1

Next interesting operation is subtraction by 1 (or, rather, regularity of successor):

$$m + 1 = n + 1 \qquad \text{implies} \qquad m = n$$

Subtraction by 1

Next interesting operation is subtraction by 1 (or, rather, regularity of successor):

$$m + 1 = n + 1 \qquad \text{implies} \qquad m = n$$

At the level of sets, this means that we should have

$$A \sqcup \{\star\} \simeq B \sqcup \{\star\} \qquad \text{implies} \qquad A \simeq B$$

Subtraction by 1

Next interesting operation is subtraction by 1 (or, rather, regularity of successor):

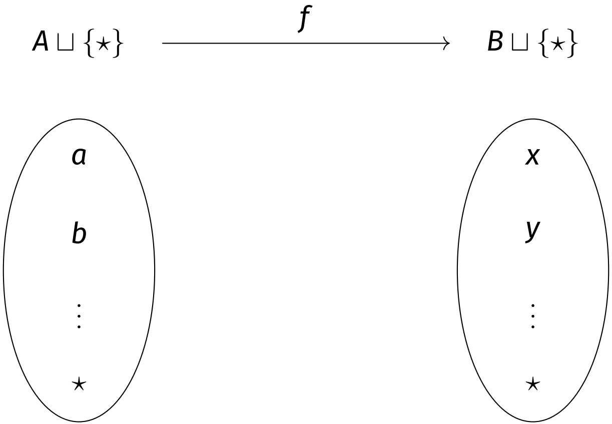
$$m + 1 = n + 1 \qquad \text{implies} \qquad m = n$$

At the level of sets, this means that we should have

$$A \sqcup \{\star\} \simeq B \sqcup \{\star\} \qquad \text{implies} \qquad A \simeq B$$

We see that this approach feels more constructive!

Subtraction by 1

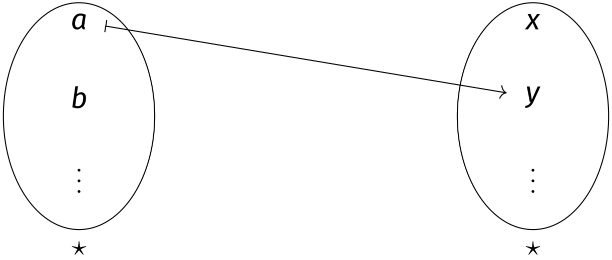


Subtraction by 1

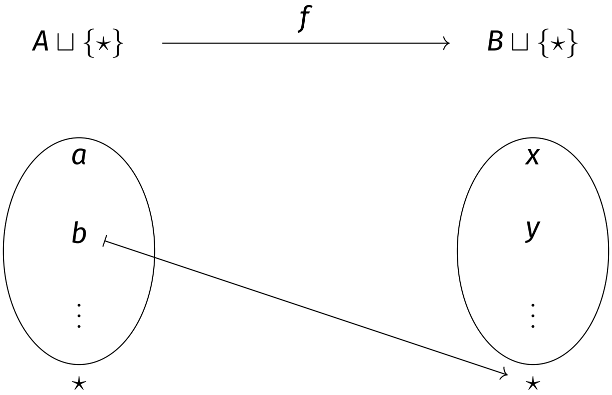
$$A \sqcup \{\star\} \xrightarrow{\quad f \quad} B \sqcup \{\star\}$$

Subtraction by 1

$$A \sqcup \{\star\} \xrightarrow{f} B \sqcup \{\star\}$$

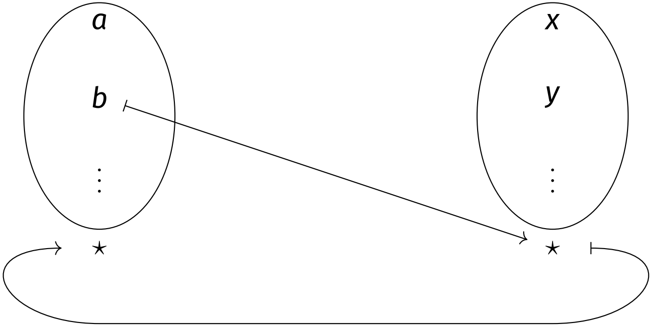


Subtraction by 1

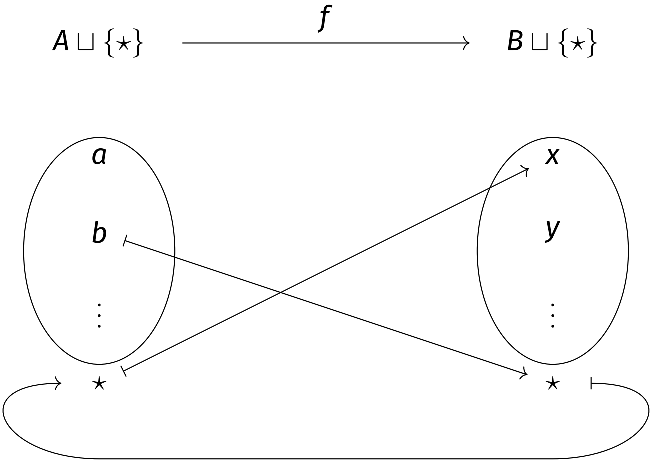


Subtraction by 1

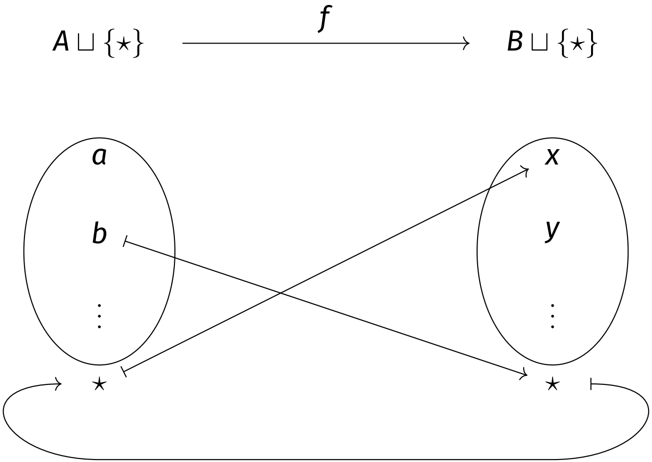
$$A \sqcup \{\star\} \xrightarrow{f} B \sqcup \{\star\}$$



Subtraction by 1



Subtraction by 1



(trace!)

Division by 2

Next interesting operation is division by 2 (or, rather, regularity of doubling):

$$m \times 2 = n \times 2 \qquad \text{implies} \qquad m = n$$

Division by 2

Next interesting operation is division by 2 (or, rather, regularity of doubling):

$$m \times 2 = n \times 2 \qquad \text{implies} \qquad m = n$$

At the level of sets, this means that we should have

$$A \times \{0, 1\} \simeq B \times \{0, 1\} \qquad \text{implies} \qquad A \simeq B$$

Division by 2

Next interesting operation is division by 2 (or, rather, regularity of doubling):

$$m \times 2 = n \times 2 \qquad \text{implies} \qquad m = n$$

At the level of sets, this means that we should have

$$A \times \{0, 1\} \simeq B \times \{0, 1\} \qquad \text{implies} \qquad A \simeq B$$

And this is indeed the case:

- if the two sets are finite, we are essentially working with natural numbers,
- otherwise we have $A \simeq A \sqcup A \simeq B \sqcup B \simeq B$.

Division by 2, constructively

This is the end of my talk

Division by 2, constructively

This could have been the end of my talk unless we wonder

*can this be performed **constructively**?*

Division by 2, constructively

This could have been the end of my talk unless we wonder

can this be performed ***constructively****?*

Namely, we have been using two dubious principles in the proof of division by 2:

Division by 2, constructively

This could have been the end of my talk unless we wonder

can this be performed ***constructively****?*

Namely, we have been using two dubious principles in the proof of division by 2:

- the **excluded-middle**: *any set is finite or not*,

Division by 2, constructively

This could have been the end of my talk unless we wonder

*can this be performed **constructively**?*

Namely, we have been using two dubious principles in the proof of division by 2:

- the **excluded-middle**: *any set is finite or not*,
- the **axiom of choice**: to construct the bijection $A \simeq A \sqcup A$.

History of division

- 1901: Bernstein gives a construction of **division by 2** in ZF
- 1922: Serpiński simplifies the construction
- 1926: Lindenbaum and Tarski construct **division by *n***
- 1943: Tarski forgets about the construction finds a new one
- 1994: Conway and Doyle manage to reinvent the 1926 solution
- 2015: Doyle, Qiu and Schartz further simplify the construction
- 2018: Swan shows that it cannot be performed entirely constructively
  by exhibiting a non-boolean topos in which $\times\mathbf{2}$ is not regular
- 2022: we extended this to HoTT
- 2023: we only need the limited principle of omniscience

Still an active research topic :)

In this work

We started from Conway and Doyle's 1994 paper *Division by three*:

- we focus on division by 2,
- we formalize the results in Agda,
- we generalize from sets to *spaces*.

The Conway-Doyle-Serpiński construction of division by 2

Suppose given a bijection

$$A \times \mathbf{2} \underset{g}{\overset{f}{\rightleftarrows}} B \times \mathbf{2}$$

with $\mathbf{2} = \{-, +\}$. We want to construct a bijection

$$A \rightleftarrows B$$

without using the axiom of choice.

The Conway-Doyle-Serpiński construction of division by 2

Suppose given a bijection

$$A \times 2 \underset{g}{\overset{f}{\rightleftarrows}} B \times 2$$

This data secretly corresponds to a directed **graph**:

- the elements of $A \times 2$ and $B \times 2$ are vertices,

The Conway-Doyle-Serpiński construction of division by 2

Suppose given a bijection

$$A \times \mathbf{2} \underset{g}{\overset{f}{\rightleftarrows}} B \times \mathbf{2}$$

This data secretly corresponds to a directed **graph**:

- the elements of $A \times \mathbf{2}$ and $B \times \mathbf{2}$ are vertices,
- the elements of A and B are edges: for $a \in A$,

$$(a, -) \xrightarrow{a} (a, +)$$

with $\mathbf{2} = \{-, +\}$

The Conway-Doyle-Serpiński construction of division by $\mathbf{2}$

Suppose given a bijection

$$A \times \mathbf{2} \underset{g}{\overset{f}{\rightleftarrows}} B \times \mathbf{2}$$

This data secretly corresponds to a directed **graph**:

- the elements of $A \times \mathbf{2}$ and $B \times \mathbf{2}$ are vertices,
- the elements of A and B are edges: for $a \in A$,

$$(a, -) \xrightarrow{a} (a, +)$$

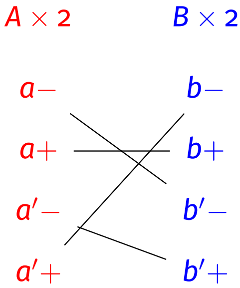
  with $\mathbf{2} = \{-, +\}$
- we identify any two vertices related by the bijection.

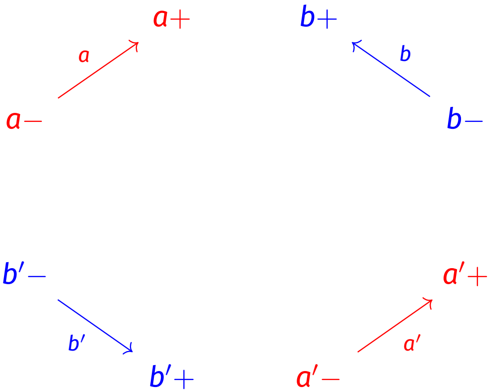
The bijection as a graph

For instance, suppose

$$A = \{a, a'\} \qquad B = \{b, b'\}$$

and consider the bijection



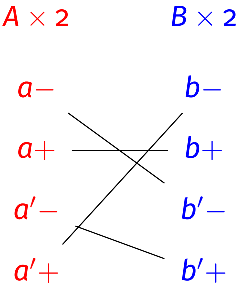


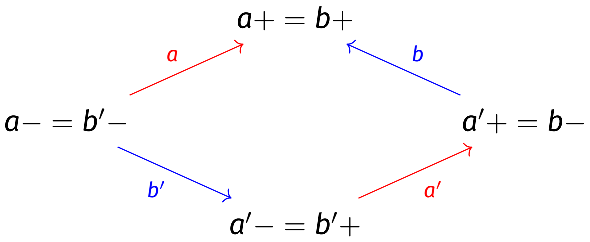
The bijection as a graph

For instance, suppose

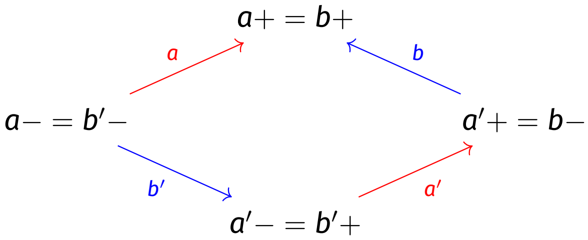
$$A = \{a, a'\} \qquad B = \{b, b'\}$$

and consider the bijection





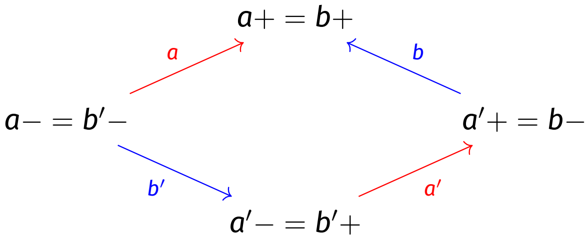
Properties of the graph



Such a graph is characterized by

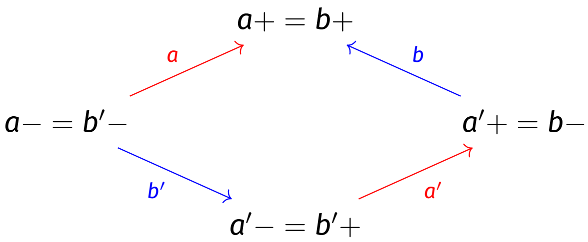
- every vertex is connected to exactly two edges
- in a path, edges alternate between elements of A and B

Chains



A **chain** is a connected component.

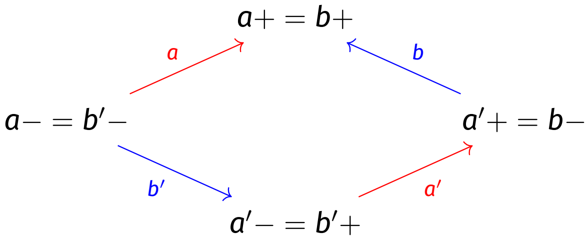
Chains



A **chain** is a connected component.

It is enough to make a bijection between the edges in A and in B in every chain.

Chains

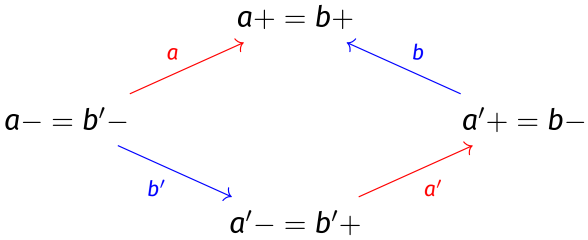


A **chain** is a connected component.

It is enough to make a bijection between the edges in A and in B in every chain.

Suppose that we pick a distinguished edge in every chain:

Chains



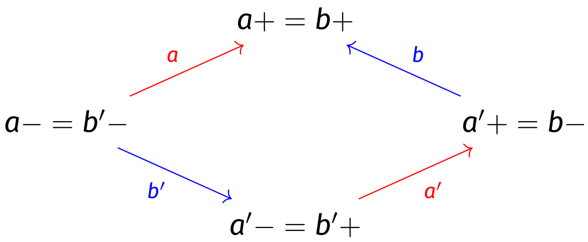
A **chain** is a connected component.

It is enough to make a bijection between the edges in A and in B in every chain.

Suppose that we pick a distinguished edge in every chain:

- every other edge is reachable from this one,

Chains



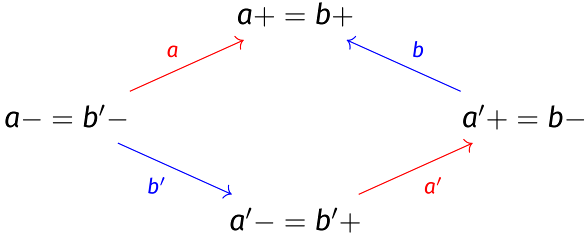
A **chain** is a connected component.

It is enough to make a bijection between the edges in A and in B in every chain.

Suppose that we pick a distinguished edge in every chain:

- every other edge is reachable from this one,
- we can thus send every red element to the "next" blue one.

Chains



A **chain** is a connected component.

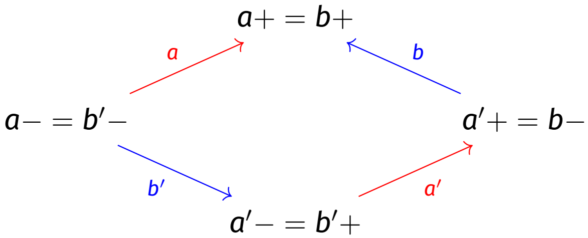
It is enough to make a bijection between the edges in A and in B in every chain.

Suppose that we pick a distinguished edge in every chain:

- every other edge is reachable from this one,
- we can thus send every red element to the “next” blue one.

We thus only need to pick an **orientation** in every chain

Chains



A **chain** is a connected component.

It is enough to make a bijection between the edges in A and in B in every chain.

Suppose that we pick a distinguished edge in every chain:

- every other edge is reachable from this one,
- we can thus send every red element to the "next" blue one.

We thus only need to pick an **orientation** in every chain …
which is not obvious without choice!

Bracketing

Consider a chain

$$\cdots \longrightarrow \cdot \longrightarrow \cdot \longleftarrow \cdot \longrightarrow \cdot \longleftarrow \cdot \longleftarrow \cdots$$

Bracketing

Consider a chain

$$\cdots \; \color{red}{\xrightarrow{\;(\;}} \color{black}{\;\cdot\;} \color{blue}{\xrightarrow{\;(\;}} \color{black}{\;\cdot\;} \color{red}{\xleftarrow{\;)\;}} \color{black}{\;\cdot\;} \color{blue}{\xrightarrow{\;(\;}} \color{black}{\;\cdot\;} \color{red}{\xleftarrow{\;)\;}} \color{black}{\;\cdot\;} \color{blue}{\xleftarrow{\;)\;}} \color{black}{\; \cdots}$$

We can interpret arrows as brackets, which does not require an orientation:

- if all the brackets are **matching**: we have a bijection,
- otherwise the non-matched brackets can have the following forms:

Bracketing

Consider a chain

$$\cdots \xrightarrow{\color{red}(} \cdot \xrightarrow{\color{blue}(} \cdot \xleftarrow{\color{red})} \cdot \xrightarrow{\color{blue}(} \cdot \xleftarrow{\color{red})} \cdot \xleftarrow{\color{blue})} \cdots$$

We can interpret arrows as brackets, which does not require an orientation:

- if all the brackets are **matching**: we have a bijection,
- otherwise the non-matched brackets can have the following forms:
    - **slope**: $\cdots \xrightarrow{} \cdot \xrightarrow{} \cdot \xrightarrow{} \cdot \xrightarrow{} \cdot \xrightarrow{} \cdot \xrightarrow{} \cdots$
      we can use any arrow as an orientation!

Bracketing

Consider a chain

$$\cdots \;{\color{red}\xrightarrow{(}}\; \cdot \;{\color{blue}\xrightarrow{(}}\; \cdot \;{\color{red}\xleftarrow{)}}\; \cdot \;{\color{blue}\xrightarrow{(}}\; \cdot \;{\color{red}\xleftarrow{)}}\; \cdot \;{\color{blue}\xleftarrow{)}}\; \cdots$$

We can interpret arrows as brackets, which does not require an orientation:

- if all the brackets are **matching**: we have a bijection,
- otherwise the non-matched brackets can have the following forms:
  - **slope**: $\cdots \;{\color{red}\longrightarrow}\; \cdot \;{\color{blue}\longrightarrow}\; \cdot \;{\color{red}\longrightarrow}\; \cdot \;{\color{blue}\longrightarrow}\; \cdot \;{\color{red}\longrightarrow}\; \cdot \;{\color{blue}\longrightarrow}\; \cdots$
    we can use any arrow as an orientation!
  - **switch**: $\cdots \;{\color{red}\longleftarrow}\; \cdot \;{\color{blue}\longleftarrow}\; \cdot \;{\color{red}\longleftarrow}\; \cdot \;{\color{blue}\longrightarrow}\; \cdot \;{\color{red}\longrightarrow}\; \cdot \;{\color{blue}\longrightarrow}\; \cdots$
    we have a canonical choice of an arrow for orientation!

Bracketing

Consider a chain

$$\cdots \xrightarrow{\color{red}(} \cdot \xrightarrow{\color{blue}(} \cdot \xleftarrow{\color{red})} \cdot \xrightarrow{\color{blue}(} \cdot \xleftarrow{\color{red})} \cdot \xleftarrow{\color{blue})} \cdots$$

We can interpret arrows as brackets, which does not require an orientation:

- if all the brackets are **matching**: we have a bijection,
- otherwise the non-matched brackets can have the following forms:
  - **slope**: $\cdots \longrightarrow \cdot \longrightarrow \cdot \longrightarrow \cdot \longrightarrow \cdot \longrightarrow \cdot \longrightarrow \cdots$
    we can use any arrow as an orientation!
  - **switch**: $\cdots \longleftarrow \cdot \longleftarrow \cdot \longleftarrow \cdot \longrightarrow \cdot \longrightarrow \cdot \longrightarrow \cdots$
    we have a canonical choice of an arrow for orientation!

In each case we can pick an orientation without choice.

A formalization in homotopy type theory

We have formalized this result in classical **homotopy type theory** (Cubical Agda):

- we have more confidence in the result (sketchy papers, choice of orientation)

A formalization in homotopy type theory

We have formalized this result in classical **homotopy type theory** (Cubical Agda):

- we have more confidence in the result (sketchy papers, choice of orientation)
- we know that the following are independent

A formalization in homotopy type theory

We have formalized this result in classical **homotopy type theory** (Cubical Agda):

- we have more confidence in the result (sketchy papers, choice of orientation)
- we know that the following are independent
    - the law of **excluded middle**: for any *proposition* `A`,

      `A ∨ ¬ A`

A formalization in homotopy type theory

We have formalized this result in classical **homotopy type theory** (Cubical Agda):

- we have more confidence in the result (sketchy papers, choice of orientation)
- we know that the following are independent
  - the law of **excluded middle**: for any *proposition* `A`,
    `A ∨ ¬ A`
  - the **axiom of choice**: for `f : A → Type`,
    `((x : A) → ∥ f x ∥) → ∥ ((x : A) → f x) ∥`

A formalization in homotopy type theory

We have formalized this result in classical **homotopy type theory** (Cubical Agda):

- we have more confidence in the result (sketchy papers, choice of orientation)
- we know that the following are independent
  - the law of **excluded middle**: for any *proposition* `A`,
    `A ∨ ¬ A`
  - the **axiom of choice**: for `f : A → Type`,
    `((x : A) → ‖ f x ‖) → ‖ ((x : A) → f x) ‖`
- we have access to HITs, which are useful (propositional trunc., quotient types)

A formalization in homotopy type theory

We have formalized this result in classical **homotopy type theory** (Cubical Agda):

- we have more confidence in the result (sketchy papers, choice of orientation)
- we know that the following are independent
  - the law of **excluded middle**: for any *proposition* `A`,
    `A ∨ ¬ A`
  - the **axiom of choice**: for `f : A → Type`,
    `((x : A) → ∥ f x ∥) → ∥ ((x : A) → f x) ∥`
- we have access to HITs, which are useful (propositional trunc., quotient types)
- we generalize the result from sets to spaces

A formalization in homotopy type theory

We have formalized this result in classical **homotopy type theory** (Cubical Agda):

- we have more confidence in the result (sketchy papers, choice of orientation)
- we know that the following are independent
  - the law of **excluded middle**: for any *proposition* `A`,
    `A ∨ ¬ A`
  - the **axiom of choice**: for `f : A → Type`,
    `((x : A) → ∥ f x ∥) → ∥ ((x : A) → f x) ∥`
- we have access to HITs, which are useful (propositional trunc., quotient types)
- we generalize the result from sets to spaces

The limited principle of omniscience

The **law of excluded middle** is: for any *proposition* `A`,

$$\texttt{A} \lor \lnot\, \texttt{A}$$

The limited principle of omniscience

The **law of excluded middle** is: for any *proposition* `A`,

`A ∨ ¬ A`

Here and after, we do not need the full power of excluded middle, but only the **limited principle of omniscience** (LPO): $\mathbb{Z}$ is omniscient.

Given a sequence `P : ℤ → Bool`,

- *either* `∀ (n : ℤ) ¬ (P n)`,
- *or* `∃ (n : ℤ) (P n)`.

NB : `Bool` is the type of *decidable propositions*

(think: we can decide the halting problem)

The limited principle of omniscience

The **limited principle of omniscience**

```
(P : ℤ → Bool) → (∀ (n : ℤ) → ¬ (P n)) ∨ (∃ (n : ℤ) → P n)))
```

is used here to determine whether

- a bracket is matched
- all brackets are matched,
- we have a switching arrow.

And it does not seem that we can avoid it.

From sets to spaces

We have formalized the original result:

Theorem
For any two types A *and* B *which are sets,*

$$\mathtt{A} \times \mathbb{2} \simeq \mathtt{B} \times \mathbb{2} \quad \rightarrow \quad \mathtt{A} \simeq \mathtt{B}.$$

From sets to spaces

We have formalized the original result:

Theorem
For any two types A *and* B *which are sets,*

$$\mathtt{A} \times \mathbb{2} \simeq \mathtt{B} \times \mathbb{2} \qquad \to \qquad \mathtt{A} \simeq \mathtt{B}.$$

but also the generalization

Theorem
For any two types A *and* B*,*

$$\mathtt{A} \times \mathbb{2} \simeq \mathtt{B} \times \mathbb{2} \qquad \to \qquad \mathtt{A} \simeq \mathtt{B}.$$

From sets to spaces

We have formalized the original result:

Theorem
For any two types `A` *and* `B` *which are sets,*

$$\mathtt{A} \times \mathbb{2} \simeq \mathtt{B} \times \mathbb{2} \quad \to \quad \mathtt{A} \simeq \mathtt{B}.$$

but also the generalization

Theorem
For any two types `A` *and* `B`*,*

$$\mathtt{A} \times \mathbb{2} \simeq \mathtt{B} \times \mathbb{2} \quad \to \quad \mathtt{A} \simeq \mathtt{B}.$$

Note: we should use **equivalences** instead of isomorphisms for types.

Components

Given a type A, we write $\|A\|_0$ for its **set** of connected components.

Components

Given a type A, we write $\|A\|_0$ for its **set** of connected components.

Given $a \in A$, we write $\mathsf{shape}(a)$ for the actual component of A, which is a **space**.

Components

Given a type A, we write $\|A\|_0$ for its **set** of connected components.

Given $a \in A$, we write $\mathsf{shape}(a)$ for the actual component of A, which is a **space**.

The bijection

$$f : A \sqcup A \to B \sqcup B$$

induces, for $a \in A \sqcup A$, a bijection

$$f_a : \mathsf{shape}(a) \to \mathsf{shape}(f(a))$$

which are thus “homotopy equivalent”.

Dividing homotopy types by 2

Theorem
Given types A *and* B, *we have*

$$\mathtt{A} \times \mathbb{2} \simeq \mathtt{B} \times \mathbb{2} \qquad \to \qquad \mathtt{A} \simeq \mathtt{B}$$

Proof.

Dividing homotopy types by 2

Theorem
Given types $\mathtt{A}$ *and* $\mathtt{B}$*, we have*

$$\mathtt{A} \times \mathbb{2} \simeq \mathtt{B} \times \mathbb{2} \quad \to \quad \mathtt{A} \simeq \mathtt{B}$$

Proof.

$$\mathtt{A} \times \mathbb{2} \simeq \mathtt{B} \times \mathbb{2}$$

Dividing homotopy types by 2

Theorem
Given types A *and* B, *we have*

$$\mathtt{A} \times \mathbb{2} \simeq \mathtt{B} \times \mathbb{2} \quad \to \quad \mathtt{A} \simeq \mathtt{B}$$

Proof.

$$\mathtt{A} \times \mathbb{2} \simeq \mathtt{B} \times \mathbb{2}$$
$$\| \mathtt{A} \times \mathbb{2} \|_0 \simeq \| \mathtt{B} \times \mathbb{2} \|_0$$

Dividing homotopy types by 2

Theorem
Given types A *and* B, *we have*

$$\mathtt{A} \times \mathbb{2} \simeq \mathtt{B} \times \mathbb{2} \quad \to \quad \mathtt{A} \simeq \mathtt{B}$$

Proof.

$$\mathtt{A} \times \mathbb{2} \simeq \mathtt{B} \times \mathbb{2}$$
$$\| \mathtt{A} \times \mathbb{2} \|_0 \simeq \| \mathtt{B} \times \mathbb{2} \|_0$$
$$\| \mathtt{A} \|_0 \times \mathbb{2} \simeq \| \mathtt{B} \|_0 \times \mathbb{2}$$

Dividing homotopy types by $\mathbb{2}$

Theorem
Given types A *and* B, *we have*

$$\mathtt{A} \times \mathbb{2} \simeq \mathtt{B} \times \mathbb{2} \quad \to \quad \mathtt{A} \simeq \mathtt{B}$$

Proof.

$$\mathtt{A} \times \mathbb{2} \simeq \mathtt{B} \times \mathbb{2}$$
$$\| \mathtt{A} \times \mathbb{2} \|_0 \simeq \| \mathtt{B} \times \mathbb{2} \|_0$$
$$\| \mathtt{A} \|_0 \times \mathbb{2} \simeq \| \mathtt{B} \|_0 \times \mathbb{2}$$
$$\| \mathtt{A} \|_0 \simeq \| \mathtt{B} \|_0$$

Dividing homotopy types by 2

Theorem
Given types `A` *and* `B`, *we have*

$$\mathtt{A} \times \mathbb{2} \simeq \mathtt{B} \times \mathbb{2} \quad \rightarrow \quad \mathtt{A} \simeq \mathtt{B}$$

Proof.

$$\mathtt{A} \times \mathbb{2} \simeq \mathtt{B} \times \mathbb{2}$$
$$\| \mathtt{A} \times \mathbb{2} \|_0 \simeq \| \mathtt{B} \times \mathbb{2} \|_0$$
$$\| \mathtt{A} \|_0 \times \mathbb{2} \simeq \| \mathtt{B} \|_0 \times \mathbb{2}$$
$$\| \mathtt{A} \|_0 \simeq \| \mathtt{B} \|_0$$

Since this bijection sends a directed arrow `a` to a reachable one `b`,

$$\mathtt{shape\ a} \simeq \mathtt{shape\ b}$$

Dividing homotopy types by 2

Theorem
Given types A *and* B, *we have*

$$\texttt{A} \times \mathbb{2} \simeq \texttt{B} \times \mathbb{2} \qquad \to \qquad \texttt{A} \simeq \texttt{B}$$

Proof.

$$\texttt{A} \times \mathbb{2} \simeq \texttt{B} \times \mathbb{2}$$
$$\| \texttt{A} \times \mathbb{2} \|_0 \simeq \| \texttt{B} \times \mathbb{2} \|_0$$
$$\| \texttt{A} \|_0 \times \mathbb{2} \simeq \| \texttt{B} \|_0 \times \mathbb{2}$$
$$\| \texttt{A} \|_0 \simeq \| \texttt{B} \|_0$$

Since this bijection sends a directed arrow a to a reachable one b,

$$\texttt{shape a} \simeq \texttt{shape b}$$

thus $\texttt{A} \simeq \Sigma[\ \texttt{a} \in \texttt{A}\ ]\ (\texttt{shape a}) \simeq \Sigma[\ \texttt{b} \in \texttt{B}\ ]\ (\texttt{shape b}) \simeq \texttt{B}$

Dividing homotopy types by 2

Theorem
Given types `A` *and* `B`, *we have*

$$\mathtt{A} \times \mathbb{2} \simeq \mathtt{B} \times \mathbb{2} \quad \to \quad \mathtt{A} \simeq \mathtt{B}$$

Proof.

$$\mathtt{A} \times \mathbb{2} \simeq \mathtt{B} \times \mathbb{2}$$
$$\| \mathtt{A} \times \mathbb{2} \|_0 \simeq \| \mathtt{B} \times \mathbb{2} \|_0$$
$$\| \mathtt{A} \|_0 \times \mathbb{2} \simeq \| \mathtt{B} \|_0 \times \mathbb{2}$$
$$\| \mathtt{A} \|_0 \simeq \| \mathtt{B} \|_0$$

Since this bijection sends a directed arrow `a` to a reachable one `b`,

$$\mathtt{shape\ a} \simeq \mathtt{shape\ b}$$

thus $\mathtt{A} \simeq \Sigma[\ \mathtt{a} \in \mathtt{A}\ ]\ (\mathtt{shape\ a}) \simeq \Sigma[\ \mathtt{b} \in \mathtt{B}\ ]\ (\mathtt{shape\ b}) \simeq \mathtt{B}$

skip

Agda formalization

Consider the type $\mathbb{2}$ with two elements `src` and `tgt`

Agda formalization

Consider the type $\mathbb{2}$ with two elements `src` and `tgt` and suppose fixed a bijection

$$\mathtt{A} \times \mathbb{2} \simeq \mathtt{B} \times \mathbb{2}$$

with `A` and `B` sets.

Agda formalization

Consider the type $\mathbb{2}$ with two elements `src` and `tgt` and suppose fixed a bijection

$$\mathtt{A} \times \mathbb{2} \simeq \mathtt{B} \times \mathbb{2}$$

with `A` and `B` sets. We define

- `Arrows = A ⊎ B`
- `Ends = Arrows ×` $\mathbb{2}$ `= dArrows`

The idea:

$$\texttt{(a , src)} \cdot \xrightarrow{\texttt{a}} \cdot \texttt{(a , tgt)}$$

Agda formalization

Consider the type $\mathbb{2}$ with two elements `src` and `tgt` and suppose fixed a bijection

$$\texttt{A} \times \mathbb{2} \simeq \texttt{B} \times \mathbb{2}$$

with `A` and `B` sets. We define

- `Arrows = A` $\uplus$ `B`
- `Ends = Arrows ×` $\mathbb{2}$ `= dArrows`

The idea:

$$(\texttt{a}\ ,\ \texttt{src}) \cdot \xrightarrow{\texttt{a}} \cdot (\texttt{a}\ ,\ \texttt{tgt})$$

We also have functions

$$\begin{aligned} \texttt{arr} : \texttt{dArrows} &\to \texttt{Arrows} \\ (\texttt{a,src}) &\mapsto \texttt{a} \\ (\texttt{a,tgt}) &\mapsto \texttt{a} \end{aligned} \qquad \begin{aligned} \texttt{fw} : \texttt{Arrows} &\to \texttt{dArrows} \\ \texttt{a} &\mapsto (\texttt{a,src}) \end{aligned}$$

Reachability

We can then define a function:

```
iterate : ℤ → dArrows → dArrows
```

Reachability

We can then define a function:

```
iterate : ℤ → dArrows → dArrows
```

And thus

```
reachable : dArrows → dArrows → Type
reachable e e' = Σ[ n ∈ ℤ ] (iterate n e ≡ e')
```

Reachability

$$\cdots \longrightarrow \cdot \longrightarrow \cdot \longleftarrow \cdot \longrightarrow \cdot \longleftarrow \cdot \longleftarrow \cdots$$

We can then define a function:

```
iterate : ℤ → dArrows → dArrows
```

And thus

```
reachable : dArrows → dArrows → Type
reachable e e' = Σ[ n ∈ ℤ ] (iterate n e ≡ e')
```

as well as

```
is-reachable : dArrows → dArrows → Type
is-reachable e e' = ∥ reachable e e' ∥
```

Revealing rechability

Recall,

```
   reachable e e' = Σ[ n ∈ ℤ ] (iterate n e ≡ e')
is-reachable e e' = ∥ reachable e e' ∥
```

Clearly, `reachable e e' → is-reachable e e'`

Revealing rechability

Recall,

$$\texttt{reachable e e' = }\Sigma\texttt{[n }\in \mathbb{Z}\texttt{] (iterate n e }\equiv\texttt{ e')}$$

$$\texttt{is-reachable e e' = }\|\texttt{ reachable e e' }\|$$

Clearly, `reachable e e' → is-reachable e e'`

Proposition
Conversely, `is-reachable e e' → reachable e e'`

Proof.

Revealing rechability

Recall,

```
reachable e e' = Σ[ n ∈ ℤ ] (iterate n e ≡ e')
is-reachable e e' = ∥ reachable e e' ∥
```

Clearly, `reachable e e' → is-reachable e e'`

Proposition

Conversely, `is-reachable e e' → reachable e e'`

Proof.

Since `A` and `B` are <u>sets</u>, so is `dArrows = (A ⊎ B) × 𝟚`.

Revealing rechability

Recall,

$$\texttt{reachable e e' = } \Sigma\texttt{[n} \in \mathbb{Z}\texttt{] (iterate n e} \equiv \texttt{e')}$$

$$\texttt{is-reachable e e' = } \| \texttt{ reachable e e' } \|$$

Clearly, `reachable e e' → is-reachable e e'`

Proposition
Conversely, `is-reachable e e' → reachable e e'`

Proof.

Since `A` and `B` are sets, so is `dArrows = (A ⊎ B) × 𝟚`.

Thus `reachable e e'` is a proposition,

Revealing rechability

Recall,

$$\texttt{reachable e e' = } \Sigma\texttt{[n} \in \mathbb{Z} \texttt{] (iterate n e} \equiv \texttt{e')}$$

$$\texttt{is-reachable e e' = } \| \texttt{ reachable e e' } \|$$

Clearly, `reachable e e' → is-reachable e e'`

Proposition
Conversely, `is-reachable e e' → reachable e e'`

Proof.

Since `A` and `B` are <u>sets</u>, so is `dArrows = (A ⊎ B) ×` $\mathbb{2}$.

Thus `reachable e e'` is a proposition,

which is moreover decidable because we are <u>classical</u>.

Revealing rechability

Recall,

$$\texttt{reachable e e' = } \Sigma\texttt{[n} \in \mathbb{Z}\texttt{] (iterate n e} \equiv \texttt{e')}$$

$$\texttt{is-reachable e e' = } \| \texttt{ reachable e e' } \|$$

Clearly, `reachable e e' → is-reachable e e'`

Proposition
Conversely, `is-reachable e e' → reachable e e'`

Proof.
Since `A` and `B` are sets, so is `dArrows = (A ⊎ B) ×` $\mathbb{2}$.
Thus `reachable e e'` is a proposition,
which is moreover decidable because we are classical.
Supposing `reachable e e'`, since we have a way to enumerate $\mathbb{Z}$,
we can therefore find an `n :` $\mathbb{Z}$ such that `iterate n e ≡ e'`. □

Chains

We are tempted to define directed chains as

```
Σ[ e ∈ dArrows ] (Σ[ e' ∈ dArrows ] (is-reachable e e'))
```

Chains

We are tempted to define directed chains as

```
Σ[ e ∈ dArrows ] (Σ[ e’ ∈ dArrows ] (is-reachable e e’))
```

However, this are rather *pointed* chains.

Chains

We are tempted to define directed chains as

```
Σ[ e ∈ dArrows ] (Σ[ e’ ∈ dArrows ] (is-reachable e e’))
```

However, this are rather *pointed* chains.

A satisfactory definition of directed chains

```
dChains = dArrows / is-reachable
```

Chains

We are tempted to define directed chains as

```
Σ[ e ∈ dArrows ] (Σ[ e' ∈ dArrows ] (is-reachable e e'))
```

However, this are rather *pointed* chains.

A satisfactory definition of directed chains

```
dChains = dArrows / is-reachable
```

and similarly, we define chains as

```
Chains = Arrows / is-reachable-arr
```

Building the bijection chainwise

Given a chain `c`, we write `chainA c` (resp. `chainB c`) for the type of its elements in `A` (resp. `B`).

Building the bijection chainwise

Given a chain `c`, we write `chainA c` (resp. `chainB c`) for the type of its elements in `A` (resp. `B`).

Lemma
If, for every chain `c`*, we have* `chainA c ≃ chainB c`*, then* `A ≃ B`.

Proof.
Given a relation `R` on a type `A`, the type is the union of its equivalence classes:

```
A ≃ Σ[ c ∈ A / R ] (fiber [_] c)
```

The result can be deduced from this and standard equivalences. □

Types of chain

Recall that a chain c can be

- well-bracketed:

$$\cdots \xrightarrow{(} \cdot \xrightarrow{(} \cdot \xleftarrow{)} \cdot \xrightarrow{(} \cdot \xleftarrow{)} \cdot \xleftarrow{)} \cdots$$

- a switching chain:

$$\cdots \longleftarrow \cdot \longleftarrow \cdot \longleftarrow \cdot \longrightarrow \cdot \longrightarrow \cdot \longrightarrow \cdots$$

- a slope:

$$\cdots \longrightarrow \cdot \longrightarrow \cdot \longrightarrow \cdot \longrightarrow \cdot \longrightarrow \cdot \longrightarrow \cdots$$

By excluded-middle, we know that we are in one of those three cases (provided we show that they are propositions).

Types of chain

Recall that a chain `c` can be

- well-bracketed:

$$\cdots \;\color{red}{\xrightarrow{(}}\; \cdot \;\color{blue}{\xrightarrow{(}}\; \cdot \;\color{red}{\xleftarrow{)}}\; \cdot \;\color{blue}{\xrightarrow{(}}\; \cdot \;\color{red}{\xleftarrow{)}}\; \cdot \;\color{blue}{\xleftarrow{)}}\; \cdots$$

- a switching chain:

$$\cdots \;\color{red}{\longleftarrow}\; \cdot \;\color{blue}{\longleftarrow}\; \cdot \;\color{red}{\longleftarrow}\; \cdot \;\color{blue}{\longrightarrow}\; \cdot \;\color{red}{\longrightarrow}\; \cdot \;\color{blue}{\longrightarrow}\; \cdots$$

- a slope:

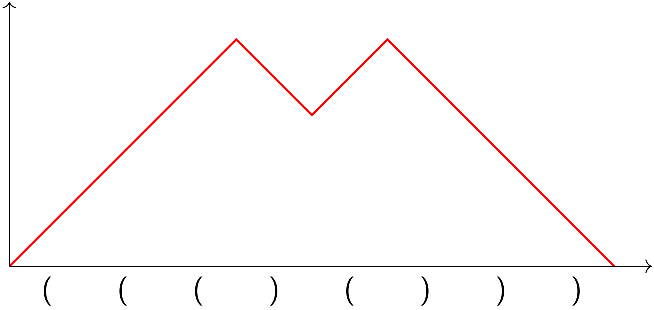
$$\cdots \;\color{red}{\longrightarrow}\; \cdot \;\color{blue}{\longrightarrow}\; \cdot \;\color{red}{\longrightarrow}\; \cdot \;\color{blue}{\longrightarrow}\; \cdot \;\color{red}{\longrightarrow}\; \cdot \;\color{blue}{\longrightarrow}\; \cdots$$

By excluded-middle, we know that we are in one of those three cases (provided we show that they are propositions).

It only remains to show `chainA c` $\simeq$ `chainB c` in each case (we will only present well-bracketing).

Well-bracketing

A word over $\{(,)\}$ may be interpreted as a *Dyck path*:



Well-bracketing

The **height** of the following path is 4:

$$\cdot \xrightarrow[1]{(} \cdot \xrightarrow[1]{(} \cdot \xleftarrow[-1]{)} \cdot \xrightarrow[1]{(} \cdot$$

Well-bracketing

The **height** of the following path is 4:

$$\cdot \xrightarrow[1]{(} \cdot \xrightarrow[1]{(} \cdot \xleftarrow[-1]{)} \cdot \xrightarrow[1]{(} \cdot$$

An arrow `a` is **matched** when it satisfies

```
Σ[ n ∈ ℕ ] (
  height (suc n) (fw a) ≡ 0 ∧
  ((k : ℕ) → k < suc n → ¬ (height k (fw x) ≡ 0)))
```

Well-bracketing

The chain of an arrow $\circ$ is **well-bracketed** when every arrow reachable from $\circ$ is matched.

Proposition
Being well-bracketed for a reachable arrow is a proposition, which is independent of the choice of $\circ$.

Well-bracketing

The chain of an arrow $\circ$ is **well-bracketed** when every arrow reachable from $\circ$ is matched.

Proposition
Being well-bracketed for a reachable arrow is a proposition, which is independent of the choice of $\circ$.

A chain is **well-bracketed** when each of its arrow is well-bracketed in the above sense.

Well-bracketing

A chain is **well-bracketed** when each of its arrow is well-bracketed.

Remark
Since

```
Chains = Arrows / is-reachable-arr
```

in order for this definition to make sense:

- we need to eliminate to a set (by definition of chains as *quotients*): here, we eliminate to `HProp`, which is a set, of which being well-bracketed is an element!
- we need to show that this is independent of the choice of the representative for the origin `o`.

Well-bracketing

A chain is **well-bracketed** when each of its arrow is well-bracketed.

Remark
Since

$$\texttt{Chains = Arrows / is-reachable-arr}$$

in order for this definition to make sense:

- we need to eliminate to a set (by definition of chains as *quotients*): here, we eliminate to `HProp`, which is a set, of which being well-bracketed is an element!
- we need to show that this is independent of the choice of the representative for the origin `o`.

Proposition
Given a well-bracketed chain `c`*, we have an equivalence* `chainA c` $\simeq$ `chainB c`.

The two other cases

- switching chains
- slopes

are handled similarly.

Division by 2

Theorem
For any two types `A` *and* `B` *which are sets,*

$$\mathtt{A} \times \mathbb{2} \simeq \mathtt{B} \times \mathbb{2} \qquad \to \qquad \mathtt{A} \simeq \mathtt{B}.$$

Our aim is now to generalize the theorem to the situation where `A` and `B` are arbitrary types (as opposed to sets).

We suppose fixed an equivalence $\texttt{A} \times \mathbb{2} \simeq \texttt{B} \times \mathbb{2}$.

The set truncation

Given a type `A`, we write $\|\ \mathtt{A}\ \|_0$ for its **set truncation**:

The set truncation

Given a type `A`, we write $\|$ `A` $\|_0$ for its **set truncation**:

We have a quotient map

$$|\_|_0 \texttt{ : A → } \| \texttt{ A } \|_0$$

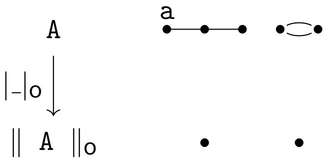
The set truncation

Given a type `A`, we write ‖ `A` ‖$_0$ for its **set truncation**:

We have a quotient map

|_|$_0$ `: A → ‖ A ‖`$_0$

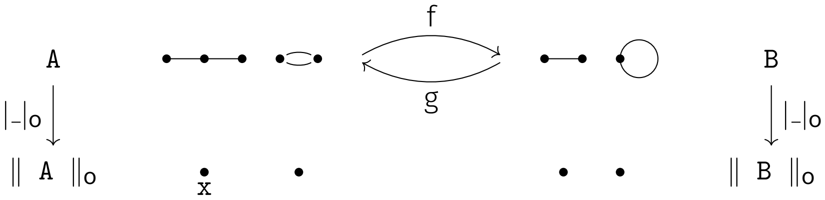
The picture we should have in mind is



Given `a : A`,

- | `a` |$_0$ is its connected component,
- `fiber` |_|$_0$ | `a` |$_0$ are the elements of this connected component.

Equivalences and set truncation

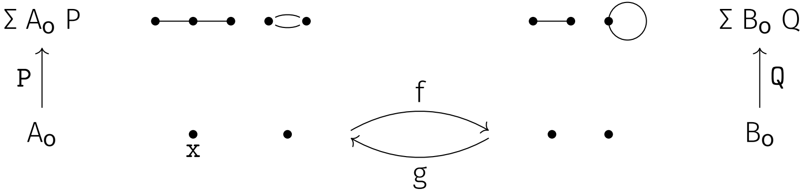


Proposition
Suppose given an equivalence `A` $\simeq$ `B` *(with* `f : A → B`*).*

- *There is an induced equivalence* $\|$ `A` $\|_0 \simeq \|$ `B` $\|_0$.
- *Given* `x :` $\|$ `A` $\|_0$*, we have an equivalence*

$$\texttt{fiber } |\_|_0 \texttt{ x} \simeq \texttt{fiber } |\_|_0 \texttt{ (}\|\|_0\texttt{-map f x)}$$

Equivalences and set truncation



Proposition
Given an equivalence $\mathtt{A_0} \simeq \mathtt{B_0}$ *(with* $\mathtt{f} : \mathtt{A_0} \to \mathtt{B_0}$*), and type families* $\mathtt{P} : \mathtt{A_0} \to \mathtt{Type}$ *and* $\mathtt{Q} : \mathtt{B_0} \to \mathtt{Type}$*, such that for* $\mathtt{x} : \mathtt{A}$*, we have*

$$\mathtt{P\ x} \simeq \mathtt{Q\ (f\ x)}$$

Then

$$\Sigma\ \mathtt{A_0}\ \mathtt{P} \simeq \Sigma\ \mathtt{B_0}\ \mathtt{Q}$$

Reachability and equivalence

Proposition
Given directed arrows $\mathtt{a}$ *and* $\mathtt{b}$ *in* $\| \ \mathtt{dArrows} \ \|_{\mathtt{O}}$ *reachable from the other, we have*

$$\mathtt{fiber}\ |\_|_{\mathtt{O}}\ \mathtt{a} \simeq \mathtt{fiber}\ |\_|_{\mathtt{O}}\ \mathtt{b}$$

Proof.
We can define functions

$$\mathtt{next} : \mathtt{dArrows} \to \mathtt{dArrows} \qquad\qquad \mathtt{prev} : \mathtt{dArrows} \to \mathtt{dArrows}$$

sending a directed arrow to the next one (in the direction), which form an equivalence, thus

$$\mathtt{fiber}\ |\_|_{\mathtt{O}}\ \mathtt{a} \simeq \mathtt{fiber}\ |\_|_{\mathtt{O}}\ (\|\ \mathtt{next}\ \|_{\mathtt{O}}\ \mathtt{a})$$

by previous proposition and we conclude by induction. □

Dividing homotopy types by 2

Theorem
Given types $\mathtt{A}$ *and* $\mathtt{B}$, *we have*

$$\mathtt{A} \times \mathbb{2} \simeq \mathtt{B} \times \mathbb{2} \quad \rightarrow \quad \mathtt{A} \simeq \mathtt{B}$$

Proof.

Dividing homotopy types by 2

Theorem
Given types $\mathtt{A}$ *and* $\mathtt{B}$*, we have*

$$\mathtt{A} \times \mathbb{2} \simeq \mathtt{B} \times \mathbb{2} \quad \rightarrow \quad \mathtt{A} \simeq \mathtt{B}$$

Proof.

$$\mathtt{A} \times \mathbb{2} \simeq \mathtt{B} \times \mathbb{2}$$

Dividing homotopy types by 2

Theorem
Given types $\mathtt{A}$ *and* $\mathtt{B}$, *we have*

$$\mathtt{A} \times \mathbb{2} \simeq \mathtt{B} \times \mathbb{2} \quad \to \quad \mathtt{A} \simeq \mathtt{B}$$

Proof.

$$\mathtt{A} \times \mathbb{2} \simeq \mathtt{B} \times \mathbb{2}$$
$$\| \mathtt{A} \times \mathbb{2} \|_0 \simeq \| \mathtt{B} \times \mathbb{2} \|_0$$

Dividing homotopy types by 2

Theorem
Given types A *and* B, *we have*

$$\mathtt{A} \times \mathbb{2} \simeq \mathtt{B} \times \mathbb{2} \quad \to \quad \mathtt{A} \simeq \mathtt{B}$$

Proof.

$$\mathtt{A} \times \mathbb{2} \simeq \mathtt{B} \times \mathbb{2}$$
$$\| \mathtt{A} \times \mathbb{2} \|_0 \simeq \| \mathtt{B} \times \mathbb{2} \|_0$$
$$\| \mathtt{A} \|_0 \times \mathbb{2} \simeq \| \mathtt{B} \|_0 \times \mathbb{2}$$

Dividing homotopy types by 2

Theorem
Given types $\mathtt{A}$ *and* $\mathtt{B}$*, we have*

$$\mathtt{A} \times \mathbb{2} \simeq \mathtt{B} \times \mathbb{2} \qquad \to \qquad \mathtt{A} \simeq \mathtt{B}$$

Proof.

$$\mathtt{A} \times \mathbb{2} \simeq \mathtt{B} \times \mathbb{2}$$
$$\| \mathtt{A} \times \mathbb{2} \|_0 \simeq \| \mathtt{B} \times \mathbb{2} \|_0$$
$$\| \mathtt{A} \|_0 \times \mathbb{2} \simeq \| \mathtt{B} \|_0 \times \mathbb{2}$$
$$\| \mathtt{A} \|_0 \simeq \| \mathtt{B} \|_0$$

Dividing homotopy types by 2

Theorem
Given types A *and* B*, we have*

$$\mathtt{A} \times \mathbb{2} \simeq \mathtt{B} \times \mathbb{2} \quad\to\quad \mathtt{A} \simeq \mathtt{B}$$

Proof.

$$\begin{gathered}
\mathtt{A} \times \mathbb{2} \simeq \mathtt{B} \times \mathbb{2} \\
\| \mathtt{A} \times \mathbb{2} \|_0 \simeq \| \mathtt{B} \times \mathbb{2} \|_0 \\
\| \mathtt{A} \|_0 \times \mathbb{2} \simeq \| \mathtt{B} \|_0 \times \mathbb{2} \\
\| \mathtt{A} \|_0 \simeq \| \mathtt{B} \|_0
\end{gathered}$$

Since this bijection sends a directed arrow a to a reachable one b,

$$\mathtt{fiber}\ |\_|_0\ \mathtt{a} \simeq \mathtt{fiber}\ |\_|_0\ \mathtt{b}$$

Dividing homotopy types by 2

Theorem
Given types A *and* B, *we have*

$$\mathtt{A} \times \mathbb{2} \simeq \mathtt{B} \times \mathbb{2} \quad \to \quad \mathtt{A} \simeq \mathtt{B}$$

Proof.

$$\mathtt{A} \times \mathbb{2} \simeq \mathtt{B} \times \mathbb{2}$$
$$\| \mathtt{A} \times \mathbb{2} \|_0 \simeq \| \mathtt{B} \times \mathbb{2} \|_0$$
$$\| \mathtt{A} \|_0 \times \mathbb{2} \simeq \| \mathtt{B} \|_0 \times \mathbb{2}$$
$$\| \mathtt{A} \|_0 \simeq \| \mathtt{B} \|_0$$

Since this bijection sends a directed arrow a to a reachable one b,

$$\mathtt{fiber}\ |\_|_0\ \mathtt{a} \simeq \mathtt{fiber}\ |\_|_0\ \mathtt{b}$$

thus $\mathtt{A} \simeq \Sigma[\ \mathtt{a} \in \mathtt{A}\ ]\ (\mathtt{fiber}\ |\_|_0\ \mathtt{a}) \simeq \Sigma[\ \mathtt{b} \in \mathtt{B}\ ]\ (\mathtt{fiber}\ |\_|_0\ \mathtt{b}) \simeq \mathtt{B}$

The Cantor-Bernstein-Schröder theorem

Theorem (Cantor-Bernstein-Schröder)
Given injections $f : A \to B$ *and* $g : B \to A$ *there is a bijection* $h : A \simeq B$

The Cantor-Bernstein-Schröder theorem

Theorem (Cantor-Bernstein-Schröder)
Given injections $f : A \to B$ and $g : B \to A$ there is a bijection $h : A \simeq B$

It can be shown in classical logic.

The Cantor-Bernstein-Schröder theorem

Theorem (Cantor-Bernstein-Schröder)
Given injections $f : A \to B$ *and* $g : B \to A$ *there is a bijection* $h : A \simeq B$ *such that* $h(x) = y$ *implies* $f(x) = y$ *or* $x = g(y)$.

It can be shown in classical logic.

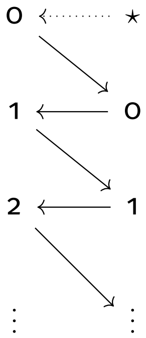
The Cantor-Bernstein-Schröder theorem

Theorem (Cantor-Bernstein-Schröder)
Given injections $f : A \to B$ and $g : B \to A$ there is a bijection $h : A \simeq B$ such that $h(x) = y$ implies $f(x) = y$ or $x = g(y)$.

It can be shown in classical logic.

Theorem (Pradic-Brown'22)
CBS implies excluded middle.

Proof.
Given P, take $A = \mathbb{N}$ and $B = \{\star \mid P\} \uplus \mathbb{N}$.



The Cantor-Bernstein-Schröder theorem

Theorem (Cantor-Bernstein-Schröder)
Given injections $f : A \to B$ and $g : B \to A$ there is a bijection $h : A \simeq B$

It can be shown in classical logic.

Theorem (Pradic-Brown'22)
CBS implies excluded middle.

Proof.
Replace $\mathbb{N}$ with an infinite type for which LPO holds

(yes, this exists! [Escardò'13]) □

The converse implication

Conjecture
"For every A and B, $2A \simeq 2B$ implies $A \simeq B$" implies LPO.

Proof.
Take $A = B = \mathbb{Z}$ and $P : \mathbb{Z} \to \texttt{Bool}$. We take the bijection $f : A \to B$ such that

- if $\neg P(n)$ then $\cdot \xrightarrow[(]{n} \cdot \xleftarrow[)]{n} \cdot$
- if $P(n)$ then $\cdot \xleftarrow[)]{n} \cdot \xleftarrow[)]{n} \cdot$
- we link $\cdot \xleftrightarrow{n-1} \cdot \xleftrightarrow{n} \cdot$

The converse implication

Conjecture

"For every A and B, $2A \simeq 2B$ implies $A \simeq B$" implies LPO.

Proof.

Take $A = B = \mathbb{Z}$ and $P : \mathbb{Z} \to$ `Bool`. We take the bijection $f : A \to B$ such that

- if $\neg P(n)$ then $\cdot \xrightarrow[(]{n} \cdot \xleftarrow[)]{n} \cdot$
- if $P(n)$ then $\cdot \xleftarrow[)]{n} \cdot \xleftarrow[)]{n} \cdot$
- we link $\cdot \xleftrightarrow{n-1} \cdot \xleftrightarrow{n} \cdot$

Thus

- if $\forall n.\neg P(n)$ then we are well-bracketed and match n with n
- if $\exists n.P(n)$ then there is an excess in ")" and we match n with $n - 1$

We have $\exists n.(P)$ if $h(0) = -1$! $\square$

Quick announcements

- the **SYCO conference** will take place at École polytechnique on *20-21 April 2023* (deadline: 6 March 2023)
- there is an open assistant professor position in *foundations of computer science* open at École polytechnique (deadline: 15 March 2023)
- please also consider submitting posters for GT **LHC**!

Questions?