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,

$$3 = \{a, b, c\} = \{x, y, z\}$$
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:

\[
3 + 2 = \{a, b, c\} \sqcup \{x, y\} = \{a, b, c, x, y\} = 5
\]
Operations on sets

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

\[ \text{is the quotient of some operation on sets?} \]

For instance,

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

\[ 3 + 2 = \begin{array}{c} a \\ b \\ c \end{array} \sqcup \begin{array}{c} x \\ y \end{array} = \begin{array}{c} a \\ b \\ c \\ x \\ y \end{array} = 5 \]

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

\[ 3 \times 2 = \begin{array}{c} a \\ b \\ c \end{array} \times \begin{array}{c} x \\ y \end{array} = \begin{array}{c} (a, x) \\ (b, x) \\ (c, x) \\ (a, y) \\ (b, y) \\ (c, y) \end{array} = 6 \]
Operations on sets

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

\[ \text{is the quotient of some operation on sets?} \]

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

\[ m + 1 = n + 1 \quad \text{implies} \quad m = n \]
Subtraction by 1

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

\[ m + 1 = n + 1 \quad \text{implies} \quad m = n \]

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

\[ A \sqcup \{\star\} \simeq B \sqcup \{\star\} \quad \text{implies} \quad A \simeq B \]
Next interesting operation is subtraction by 1 (or, rather, regularity of successor):

\[ m + 1 = n + 1 \quad \text{implies} \quad m = n \]

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

\[ A \sqcup \{\star\} \simeq B \sqcup \{\star\} \quad \text{implies} \quad A \simeq B \]

We see that this approach feels more constructive!
Subtraction by 1

\[
A \sqcup \{\star\} \xrightarrow{f} B \sqcup \{\star\}
\]
Subtraction by 1

\[ A \sqcup \{\star\} \xrightarrow{f} B \sqcup \{\star\} \]

\begin{align*}
A & = \{a, b, \ldots, \star\} \\
B & = \{x, y, \ldots, \star\}
\end{align*}
Subtraction by 1

\[ A \sqcup \{ \star \} \xrightarrow{f} B \sqcup \{ \star \} \]
Subtraction by 1

\[ A \sqcup \{ \star \} \xrightarrow{f} B \sqcup \{ \star \} \]
Subtraction by 1

\[ A \sqcup \{\star\} \xrightarrow{f} B \sqcup \{\star\} \]

\[
\begin{array}{c}
  a \\
  b \\
  \vdots \\
  \star
\end{array}
\]

\[
\begin{array}{c}
  x \\
  y \\
  \vdots \\
  \star
\end{array}
\]
Subtraction by 1

\[ A \sqcup \{\star\} \xrightarrow{f} B \sqcup \{\star\} \]
Subtraction by 1

\[
A \sqcup \{\star\} \xrightarrow{f} B \sqcup \{\star\}
\]

(a, b) \rightarrow (x, \ldots)

(\star, \ldots) \rightarrow (\star, \ldots)

(trace!)
Division by 2

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

\[ m \times 2 = n \times 2 \quad \text{implies} \quad m = n \]
Division by 2

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

\[ m \times 2 = n \times 2 \quad \text{implies} \quad m = n \]

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

\[ A \times \{0, 1\} \simeq B \times \{0, 1\} \quad \text{implies} \quad A \simeq B \]
Division by 2

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

\[ m \times 2 = n \times 2 \quad \text{implies} \quad m = n \]

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

\[ A \times \{0, 1\} \simeq B \times \{0, 1\} \quad \text{implies} \quad A \simeq B \]

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

\textit{can this be performed \textit{constructively}?}

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

- the \textbf{excluded-middle}: \textit{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 \( A \sim A \sqcup A \).
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 \( \times 2 \) 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

\[ f : A \times 2 \to B \times 2 \]

\[ g : B \times 2 \to A \times 2 \]

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

\[ f : A \to B \]

\[ g : B \to A \]

without using the axiom of choice.
The Conway-Doyle-Serpiński construction of division by 2

Suppose given a bijection

\[
\begin{array}{ccc}
A \times 2 & \xrightarrow{f} & B \times 2 \\
A & \xleftarrow{g} & B \\
\end{array}
\]

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

\[
\begin{array}{c}
A \times 2 \\
\Downarrow f \\
B \times 2
\end{array} \\
\begin{array}{c}
A \times 2 \\
\Downarrow g \\
B \times 2
\end{array}
\]

This data secretly corresponds to a directed graph:

- the elements of \( A \times 2 \) and \( B \times 2 \) are vertices,
- the elements of \( A \) and \( B \) are edges: for \( a \in A \),

\[
(a, -) \xrightarrow{a} (a, +)
\]

with \( 2 = \{-, +\} \).
The Conway-Doyle-Serpiński construction of division by 2

Suppose given a bijection

\[ \begin{align*}
A \times 2 & \xrightarrow{f} B \times 2 \\
B \times 2 & \xrightarrow{g} A \times 2
\end{align*} \]

This data secretly corresponds to a directed graph:

- the elements of \( A \times 2 \) and \( B \times 2 \) are vertices,
- the elements of \( A \) and \( B \) are edges: for \( a \in A \),

\[
(a, -) \xrightarrow{a} (a, +)
\]

with \( 2 = \{-, +\} \)

- we identify any two vertices related by the bijection.
The bijection as a graph

For instance, suppose

$$A = \{a, a'\}$$

and consider the bijection

$$B = \{b, b'\}$$
The bijection as a graph

For instance, suppose

\[ A = \{a, a'\} \quad \text{and} \quad B = \{b, b'\} \]

and consider the bijection

\[
\begin{align*}
A \times 2 & \quad \text{and} \quad B \times 2 \\
\begin{array}{r}
a- \\
a+ \\
a' - \\
a' + \\
b- \\
b+ \\
b' - \\
b' +
\end{array}
& \quad \begin{array}{r}
a+ = b+ \\
a- = b' - \\
a' + = b- \\
a' - = b' +
\end{array}
\end{align*}
\]
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 $B$
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.
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,
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

\[ a+ = b+ \]
\[ a- = b'- \]
\[ a' + = b- \]
\[ a' - = b'+ \]

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

\[ \cdots \rightarrow \cdot \rightarrow \cdot \rightarrow \cdot \leftarrow \cdot \rightarrow \cdot \leftarrow \cdot \leftarrow \cdots \]
Bracketing

Consider a chain

\[ \cdots \rightarrow ( \rightarrow \ . \ . \ . \ . \ ) \rightarrow ( \rightarrow \ . \ . \ . \ . \ ) \rightarrow ( \rightarrow \ . \ . \ . \ . \ ) \rightarrow ( \rightarrow \ . \ . \ . \ . \ ) \rightarrow ( \rightarrow \ . \ . \ . \ . \ ) \cdots \]

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

\[ \cdots \quad ( \rightarrow \quad . \quad ( \rightarrow \quad . \quad \leftarrow ) \quad . \quad ( \rightarrow \quad . \quad \leftarrow ) \quad . \quad \leftarrow ) \quad \cdots \]

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**: \[ \cdots \quad \rightarrow \quad . \quad \rightarrow \quad . \quad \rightarrow \quad . \quad \rightarrow \quad . \quad \rightarrow \quad . \quad \rightarrow \quad \cdots \]
    we can use any arrow as an orientation!
Bracketing

Consider a chain

\[ \ldots \rightarrow \ ( \rightarrow \cdot \leftarrow) \cdot \rightarrow \leftarrow) \cdot \leftarrow) \ldots \]

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**: \[ \ldots \rightarrow \cdot \rightarrow \cdot \rightarrow \cdot \rightarrow \cdot \rightarrow \cdot \rightarrow \ldots \]
    we can use any arrow as an orientation!
  - **switch**: \[ \ldots \leftarrow \cdot \leftarrow \cdot \leftarrow \cdot \rightarrow \cdot \rightarrow \cdot \rightarrow \ldots \]
    we have a canonical choice of an arrow for orientation!
Consider a chain

\[ \ldots \rightarrow ( \rightarrow \cdot \leftarrow ) \rightarrow ( \rightarrow \cdot \leftarrow ) \rightarrow ( \rightarrow \cdot \leftarrow ) \rightarrow ( \rightarrow \cdot \leftarrow ) \rightarrow ( \rightarrow \cdot \leftarrow ) \rightarrow ( \rightarrow \cdot \leftarrow ) \rightarrow ( \rightarrow \cdot \leftarrow ) \rightarrow \ldots \]

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**: \[ \ldots \rightarrow \cdot \rightarrow \cdot \rightarrow \cdot \rightarrow \cdot \rightarrow \cdot \rightarrow \cdot \rightarrow \cdot \rightarrow \ldots \]
    we can use any arrow as an orientation!
  - **switch**: \[ \ldots \leftarrow \cdot \leftarrow \cdot \leftarrow \cdot \leftarrow \cdot \rightarrow \cdot \rightarrow \cdot \rightarrow \cdot \rightarrow \ldots \]
    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

- the law of excluded middle: for any proposition $A$, $A \lor \neg A$
- the axiom of choice: for $f : A \to \text{Type}$, $(\forall x : A. \| f x \|) \to \| (\forall 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 $A$,
    $A \lor \neg A$
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 $A$,
    \[
    A \lor \neg A
    \]
  - the **axiom of choice**: for $f : A \to \text{Type}$,
    \[
    ((x : A) \to \| f x \|) \to \| (x : A) \to f x \|
    \]
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 $A$,
    
    $$A \lor \lnot A$$
  
  - the **axiom of choice**: for $f : A \to \text{Type}$,
    
    $$((x : A) \to \| f x \|) \to \| ((x : A) \to f x) \|$$
  
- 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 $A$, $A \lor \neg A$
  - the **axiom of choice**: for $f : A \to \text{Type}$,
    $\left( (x : A) \to \|f x\| \right) \to \| (x : A) \to f x \|$  
- 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 $A$, $A \lor \neg A$
  - the **axiom of choice**: for $f : A \to \text{Type}$,
    $$(x : A) \to \| f x \| \to \| (x : A) \to f x \|$$
- 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 $A$,

$$A \lor \neg A$$
The limited principle of omniscience

The law of excluded middle is: for any proposition \( A \),

\[
A \lor \neg A
\]

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 \( P : \mathbb{Z} \rightarrow \text{Bool} \),

- either \( \forall (n : \mathbb{Z}) \neg (P \ n) \),
- or \( \exists (n : \mathbb{Z}) (P \ n) \).

NB: \( \text{Bool} \) is the type of decidable propositions
(think: we can decide the halting problem)
The limited principle of omniscience

The limited principle of omniscience

$$(P : \mathbb{Z} \rightarrow \text{Bool}) \rightarrow (\forall (n : \mathbb{Z}) \rightarrow \neg (P \ n)) \lor (\exists (n : \mathbb{Z}) \rightarrow P \ n))$$

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,*

$$A \times \mathbb{2} \sim B \times \mathbb{2} \quad \rightarrow \quad A \sim B.$$
From sets to spaces

We have formalized the original result:

**Theorem**

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

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

but also the generalization

**Theorem**

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

$$A \times \mathbb{2} \sim B \times \mathbb{2} \rightarrow A \sim B.$$

Note: we should use equivalences instead of isomorphisms for types.
From sets to spaces

We have formalized the original result:

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

$$A \times 2 \simeq B \times 2 \quad \rightarrow \quad A \simeq B.$$ 

but also the generalization

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

$$A \times 2 \simeq B \times 2 \quad \rightarrow \quad A \simeq B.$$ 

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

Given $a \in A$, we write $\text{shape}(a)$ 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 $\text{shape}(a)$ 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

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

which are thus “homotopy equivalent”.
Dividing homotopy types by 2

Theorem

Given types $A$ and $B$, we have

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

Proof.
Dividing homotopy types by 2

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

\[ A \times \{0,1\} \sim B \times \{0,1\} \quad \rightarrow \quad A \sim B \]

Proof.

\[ A \times \{0,1\} \sim B \times \{0,1\} \]
Dividing homotopy types by 2

Theorem
*Given types* \(A\) *and* \(B\), we have

\[ A \times 2 \simeq B \times 2 \rightarrow A \simeq B \]

Proof.

\[ A \times 2 \simeq B \times 2 \]
\[ \| A \times 2 \|_o \simeq \| B \times 2 \|_o \]
Dividing homotopy types by 2

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

$$A \times 2 \simeq B \times 2 \rightarrow A \simeq B$$

Proof.

$$A \times 2 \simeq B \times 2$$

$$\parallel A \times 2 \parallel_o \simeq \parallel B \times 2 \parallel_o$$

$$\parallel A \parallel_o \times 2 \simeq \parallel B \parallel_o \times 2$$
Dividing homotopy types by 2

Theorem

Given types $A$ and $B$, we have

$$A \times 2 \simeq B \times 2 \rightarrow A \simeq B$$

Proof.

$$A \times 2 \simeq B \times 2$$

$$\| A \times 2 \|_o \simeq \| B \times 2 \|_o$$

$$\| A \|_o \times 2 \simeq \| B \|_o \times 2$$

$$\| A \|_o \simeq \| B \|_o$$
Dividing homotopy types by 2

Theorem
Given types \( A \) and \( B \), we have

\[
A \times 2 \simeq B \times 2 \quad \rightarrow \quad A \simeq B
\]

Proof.

\[
\begin{align*}
A \times 2 & \simeq B \times 2 \\
\| A \times 2 \|_0 & \simeq \| B \times 2 \|_0 \\
\| A \|_0 \times 2 & \simeq \| B \|_0 \times 2 \\
\| A \|_0 & \simeq \| B \|_0
\end{align*}
\]

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

\[
\text{shape } a \simeq \text{shape } b
\]
Dividing homotopy types by 2

Theorem

Given types $A$ and $B$, we have

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

Proof.

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

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

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

$$\| A \|_0 \simeq \| B \|_0$$

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

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

thus

$$A \simeq \Sigma [ \ a \in A \ ] \ (\text{shape } a) \simeq \Sigma [ \ b \in B \ ] \ (\text{shape } b) \simeq B$$
Dividing homotopy types by 2

**Theorem**
Given types \( A \) and \( B \), we have

\[
A \times 2 \simeq B \times 2 \quad \rightarrow \quad A \simeq B
\]

**Proof.**

\[
\parallel A \times 2 \parallel_0 \simeq \parallel B \times 2 \parallel_0 \\
\parallel A \parallel_0 \times 2 \simeq \parallel B \parallel_0 \times 2 \\
\parallel A \parallel_0 \simeq \parallel B \parallel_0
\]

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

\[
\text{shape } a \simeq \text{shape } b
\]

thus

\[
A \simeq \Sigma[\ a \in A \ ] \ (\text{shape } a) \simeq \Sigma[\ b \in B \ ] \ (\text{shape } b) \simeq B
\]
Consider the type \( \mathbb{2} \) with two elements \( \text{src} \) and \( \text{tgt} \)
Agda formalization

Consider the type \( \mathcal{2} \) with two elements \( \text{src} \) and \( \text{tgt} \) and suppose fixed a bijection

\[
A \times \mathcal{2} \simeq B \times \mathcal{2}
\]

with \( A \) and \( B \) sets.
Agda formalization

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

\[
A \times \mathbb{2} \simeq B \times \mathbb{2}
\]

with A and B sets. We define

- Arrows = \( A \uplus B \)
- Ends = Arrows \( \times \mathbb{2} = dArrows \)

The idea:

\[
(a, \text{src}) \xrightarrow{a} (a, \text{tgt})
\]
Agda formalization

Consider the type \( 2 \) with two elements \( \text{src} \) and \( \text{tgt} \) and suppose fixed a bijection

\[
A \times 2 \simeq B \times 2
\]

with \( A \) and \( B \) sets. We define

- \( \text{Arrows} = A \uplus B \)
- \( \text{Ends} = \text{Arrows} \times 2 = d\text{Arrows} \)

The idea:

\[
(a, \text{src}) \cdot \overset{a}{\rightarrow} \cdot (a, \text{tgt})
\]

We also have functions

\[
\text{arr} : d\text{Arrows} \rightarrow \text{Arrows} \quad \text{fw} : \text{Arrows} \rightarrow d\text{Arrows}
\]

\[
(a, \text{src}) \mapsto a \quad a \mapsto (a, \text{src})
\]

\[
(a, \text{tgt}) \mapsto a
\]
Reachability

We can then define a function:

\[
\text{iterate} : \mathbb{Z} \rightarrow \text{dArrows} \rightarrow \text{dArrows}
\]
Reachability

We can then define a function:

\[ \text{iterate} : \mathbb{Z} \to \text{dArrows} \to \text{dArrows} \]

And thus

\[ \text{reachable} : \text{dArrows} \to \text{dArrows} \to \text{Type} \]

\[ \text{reachable } e \, e' = \Sigma [ n \in \mathbb{Z} ] (\text{iterate } n \, e \equiv e') \]
Reachability

We can then define a function:

\[
\text{iterate} : \mathbb{Z} \to \text{dArrows} \to \text{dArrows}
\]

And thus

\[
\text{reachable} : \text{dArrows} \to \text{dArrows} \to \text{Type}
\]

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

as well as

\[
\text{is-reachable} : \text{dArrows} \to \text{dArrows} \to \text{Type}
\]

\[
\text{is-reachable } e \; e' = \| \text{reachable } e \; e' \|\]
Revealing reachability

Recall,

\[
\text{reachable } e \ e' = \Sigma [ \ n \in \mathbb{Z} ] \ (\text{iterate } n \ e \equiv e')
\]
\[
\text{is-reachable } e \ e' = \| \text{reachable } e \ e' \|
\]

Clearly, \(\text{reachable } e \ e' \rightarrow \text{is-reachable } e \ e'\)
Revealing rechability

Recall,

\[ \text{reachable } e \  e' = \Sigma [ \ n \in \mathbb{Z} ] \ (\text{iterate } n \ e \equiv e') \]
\[ \text{is-reachable } e \  e' = \left\lVert \text{reachable } e \  e' \right\rVert \]

Clearly, \( \text{reachable } e \  e' \rightarrow \text{is-reachable } e \  e' \)

**Proposition**
\( \text{Conversely, is-reachable } e \  e' \rightarrow \text{reachable } e \  e' \)

**Proof.**
Revealing reachability

Recall,

\[ \text{reachable } e \leftrightarrow e' = \Sigma \left[ n \in \mathbb{Z} \right] (\text{iterate } n \ e \equiv e') \]
\[ \text{is-reachable } e \leftrightarrow e' = \parallel \text{reachable } e \leftrightarrow e' \parallel \]

Clearly, \( \text{reachable } e \leftrightarrow e' \rightarrow \text{is-reachable } e \leftrightarrow e' \)

**Proposition**

Conversely, \( \text{is-reachable } e \leftrightarrow e' \rightarrow \text{reachable } e \leftrightarrow e' \)

**Proof.**

Since \( A \) and \( B \) are sets, so is \( \text{dArrows} = (A \cup B) \times \mathcal{P} \).
Revealing reachability

Recall,

\[
\text{reachable } e \; e' = \Sigma \left[ n \in \mathbb{Z} \right] (\text{iterate } n \; e \equiv e')
\]
\[
\text{is-reachable } e \; e' = \| \text{reachable } e \; e' \|
\]

Clearly, \( \text{reachable } e \; e' \rightarrow \text{is-reachable } e \; e' \)

**Proposition**

Conversely, \( \text{is-reachable } e \; e' \rightarrow \text{reachable } e \; e' \)

**Proof.**

Since \( \text{A and B are sets} \), so is \( \text{dArrows} = (A \cup B) \times \mathcal{P} \).

Thus \( \text{reachable } e \; e' \) is a proposition,
Revealing reachability

Recall,

\[ \text{reachable } e \; e' = \sum_{n \in \mathbb{Z}} (\text{iterate } n \; e \equiv e') \]
\[ \text{is-reachable } e \; e' = \| \text{reachable } e \; e' \| \]

Clearly, \( \text{reachable } e \; e' \rightarrow \text{is-reachable } e \; e' \)

**Proposition**

Conversely, \( \text{is-reachable } e \; e' \rightarrow \text{reachable } e \; e' \)

**Proof.**

Since \( A \) and \( B \) are sets, so is \( \text{dArrows} = (A \cup B) \times \wp \).

Thus \( \text{reachable } e \; e' \) is a proposition,

which is moreover decidable because we are classical.
Revealing reachability

Recall,

\[
\text{reachable } e \ e' = \Sigma \left[ \ n \in \mathbb{Z} \right] (\text{iterate } n \ e \equiv e')
\]

\[
\text{is-reachable } e \ e' = \| \text{reachable } e \ e' \|
\]

Clearly, \(\text{reachable } e \ e' \rightarrow \text{is-reachable } e \ e'\)

**Proposition**

Conversely, \(\text{is-reachable } e \ e' \rightarrow \text{reachable } e \ e'\)

**Proof.**

Since \(A\) and \(B\) are sets, so is \(\text{dArrows} = (A \cup B) \times \mathcal{P}\).

Thus \(\text{reachable } e \ e'\) is a proposition,

which is moreover decidable because we are classical.

Supposing \(\text{reachable } e \ e'\), since we have a way to enumerate \(\mathbb{Z}\),

we can therefore find an \(n : \mathbb{Z}\) such that \(\text{iterate } n \ e \equiv e'\).
We are tempted to define directed chains as

\[ \Sigma \left[ e \in \text{dArrows} \right] (\Sigma \left[ e' \in \text{dArrows} \right] (\text{is-reachable} e e')) \]
Chains

We are tempted to define directed chains as

\[ \sum_{e \in dArrows} \left( \sum_{e' \in dArrows} \text{is-reachable } e \leftrightarrow e' \right) \]

However, this are rather pointed chains.
We are tempted to define directed chains as

$$\Sigma[ e \in d\text{Arrows} ] (\Sigma[ e' \in d\text{Arrows} ] (\text{is-reachable} \ e \ e'))$$

However, this are rather *pointed* chains.

A satisfactory definition of directed chains

$$d\text{Chains} = d\text{Arrows} / \text{is-reachable}$$
We are tempted to define directed chains as

$$\Sigma[ e \in dArrows ] (\Sigma[ e' \in dArrows ] (\text{is-reachable} e e'))$$

However, this are rather pointed chains.

A satisfactory definition of directed chains

$$dChains = dArrows / \text{is-reachable}$$

and similarly, we define chains as

$$\text{Chains} = \text{Arrows} / \text{is-reachable-arr}$$
Building the bijection chainwise

Given a chain \( c \), we write \texttt{chainA } c \texttt{(resp. chainB } c \texttt{)} for the type of its elements in \( A \) (resp. \( B \)).
Building the bijection chainwise

Given a chain \( c \), we write \( \text{chain}_A \ c \) (resp. \( \text{chain}_B \ c \)) for the type of its elements in \( A \) (resp. \( B \)).

**Lemma**

If, for every chain \( c \), we have \( \text{chain}_A \ c \cong \text{chain}_B \ c \), then \( A \cong B \).

**Proof.**

Given a relation \( R \) on a type \( A \), the type is the union of its equivalence classes:

\[
A \cong \Sigma \left[ c \in A / R \right] \ \text{fiber} \ \ _c
\]

The result can be deduced from this and standard equivalences.
Types of chain

Recall that a chain $c$ can be

- well-bracketed:

  \[ \cdots \rightarrow \cdot \left( \rightarrow \cdot \leftarrow \right) \cdot \left( \rightarrow \cdot \leftarrow \right) \cdot \leftarrow \cdots \]

- a switching chain:

  \[ \cdots \leftarrow \cdot \leftrightarrow \cdot \leftrightarrow \cdot \rightarrow \cdot \rightarrow \cdot \rightarrow \cdots \]

- a slope:

  \[ \cdots \rightarrow \cdot \rightarrow \cdot \rightarrow \cdot \rightarrow \cdot \rightarrow \cdot \rightarrow \cdots \]

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:
  \[
  \cdots \xrightarrow{\cdot} \xrightarrow{(\cdot)} \xleftarrow{\cdot} \xrightarrow{(\cdot)} \xleftarrow{\cdot} \xrightarrow{(\cdot)} \xleftarrow{\cdot} \xrightarrow{(\cdot)} \xleftarrow{\cdot} \cdots
  \]

- a switching chain:
  \[
  \cdots \leftarrow \xrightarrow{\cdot} \leftarrow \xrightarrow{\cdot} \leftarrow \xrightarrow{\cdot} \cdots \xrightarrow{\cdot} \leftarrow \xrightarrow{\cdot} \cdots
  \]

- a slope:
  \[
  \cdots \xrightarrow{\cdot} \xrightarrow{\cdot} \xrightarrow{\cdot} \xrightarrow{\cdot} \xrightarrow{\cdot} \xrightarrow{\cdot} \xrightarrow{\cdot} \cdots
  \]

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 $\text{chain}_A \ c \simeq \text{chain}_B \ c$ 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:

\[
\cdot \; \frac{(1 \rightarrow)}{1} \cdot \; \frac{(1 \rightarrow)}{1} \cdot \; \frac{(\leftarrow \; -1)}{} \cdot \; \frac{(1 \rightarrow)}{1} \cdot 
\]
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

\[
\Sigma[ n \in \mathbb{N} ] ( \\
\text{height (suc } n) (\text{fw } a) \equiv 0 \land \\
((k : \mathbb{N}) \to k < \text{suc } n \to \neg (\text{height } k (\text{fw } x) \equiv 0)))
\]
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 $\circ$. 
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

\[
\text{Chains} = \frac{\text{Arrows}}{\text{is-reachable-arr}}
\]

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 \( \text{HProp} \), 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 \( \circ \).
Well-bracketing

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

**Remark**

Since

\[
\text{Chains} = \text{Arrows} / \text{is-reachable-arr}
\]

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 \( \text{HProp} \), 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 \( \text{chainA} \ c \simeq \text{chainB} \ c \).*
The two other cases

• switching chains
• slopes

are handled similarly.
Division by 2

Theorem

For any two types $A$ and $B$ which are sets,

$$A \times 2 \cong B \times 2 \quad \rightarrow \quad A \cong B.$$
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 $A \times \mathbb{2} \simeq B \times \mathbb{2}$. 
The set truncation

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

$\|\bullet\bullet\bullet\bullet\bullet\bullet\bullet\bullet\cdot\|_0 = \bullet \bullet$
The set truncation

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

\[\|\bullet\bullet\bullet\bullet\bullet\|_0 = \bullet\bullet\bullet\bullet\bullet\]

We have a quotient map

\[|\_\_|_0 : A \rightarrow \|A\|_0\]
The set truncation

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

$\|\bullet\bullet\bullet\bullet\bullet\bullet\bullet\bullet\|_0 = \bullet \bullet$

We have a quotient map

$\ |-|_0 : A \to \|A\|_0$

The picture we should have in mind is

Given $a : A$,
- $\ |a|_0$ is its connected component,
- fiber $\ |-|_0 \mid a \mid_0$ are the elements of this connected component.
Proposition
Suppose given an equivalence $A \simeq B$ (with $f : A \to B$).

- There is an induced equivalence $\| A \|_0 \simeq \| B \|_0$.
- Given $x : \| A \|_0$, we have an equivalence
  \[
  \text{fiber } |-|_0 x \simeq \text{fiber } |-|_0 (\| \|_0 \text{-map } f x)
  \]
Equivalences and set truncation

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

$$P x \simeq Q (f x)$$

Then

$$\Sigma A_0 P \simeq \Sigma B_0 Q$$
Reachability and equivalence

Proposition
Given directed arrows $a$ and $b$ in $\parallel d\text{Arrows} \parallel_o$ reachable from the other, we have

$$\text{fiber } \|_0 a \simeq \text{fiber } \|_0 b$$

Proof.
We can define functions

$$\text{next} : d\text{Arrows} \rightarrow d\text{Arrows} \quad \quad \text{prev} : d\text{Arrows} \rightarrow d\text{Arrows}$$

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

$$\text{fiber } \|_0 a \simeq \text{fiber } \|_0 (\parallel \text{next} \parallel_o a)$$

by previous proposition and we conclude by induction.
Dividing homotopy types by 2

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

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

Proof.
Dividing homotopy types by 2

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

$$A \times 2 \simeq B \times 2 \quad \rightarrow \quad A \simeq B$$

Proof.

$$A \times 2 \simeq B \times 2$$
Dividing homotopy types by 2

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

$$A \times 2 \simeq B \times 2 \quad \rightarrow \quad A \simeq B$$

Proof.

$$A \times 2 \simeq B \times 2$$

$$\| A \times 2 \|_o \simeq \| B \times 2 \|_o$$
Dividing homotopy types by 2

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

\[
A \times \{0,1\} \simeq B \times \{0,1\} \Rightarrow A \simeq B
\]

**Proof.**

\[
\begin{align*}
A \times \{0,1\} &\simeq B \times \{0,1\} \\
\| A \times \{0,1\} \|_0 &\simeq \| B \times \{0,1\} \|_0 \\
\| A \|_0 \times \{0,1\} &\simeq \| B \|_0 \times \{0,1\}
\end{align*}
\]
Dividing homotopy types by 2

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

$$A \times 2 \cong B \times 2 \quad \rightarrow \quad A \cong B$$

Proof.

$$A \times 2 \cong B \times 2$$

$$\parallel A \times 2 \parallel_0 \cong \parallel B \times 2 \parallel_0$$

$$\parallel A \parallel_0 \times 2 \cong \parallel B \parallel_0 \times 2$$

$$\parallel A \parallel_0 \cong \parallel B \parallel_0$$
Dividing homotopy types by 2

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

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

Proof.

$$\| A \times \mathbb{2} \|_o \simeq \| B \times \mathbb{2} \|_o$$

$$\| A \|_o \times \mathbb{2} \simeq \| B \|_o \times \mathbb{2}$$

$$\| A \|_o \simeq \| B \|_o$$

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

$$\text{fiber } \|_o a \simeq \text{fiber } \|_o b$$
Dividing homotopy types by 2

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

$$A \times 2 \simeq B \times 2 \rightarrow A \simeq B$$

Proof.

$$A \times 2 \simeq B \times 2$$

$$\parallel A \times 2 \parallel_0 \simeq \parallel B \times 2 \parallel_0$$

$$\parallel A \parallel_0 \times 2 \simeq \parallel B \parallel_0 \times 2$$

$$\parallel A \parallel_0 \simeq \parallel B \parallel_0$$

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

$$\text{fiber } \downarrow_0 a \simeq \text{fiber } \downarrow_0 b$$

thus

$$A \simeq \Sigma[ a \in A ] (\text{fiber } \downarrow_0 a) \simeq \Sigma[ b \in B ] (\text{fiber } \downarrow_0 b) \simeq B$$
The Cantor-Bernstein-Schröder theorem

Theorem (Cantor-Bernstein-Schröder)
Given injections $f: A \to B$ and $g: B \to A$ there is a bijection $h: A \simeq B$.
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 $f : A \rightarrow B$ and $g : B \rightarrow A$ 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 $B = \{\star \mid P\} \cup \mathbb{N}$. 

\[ 0 \leftarrow \cdots \leftarrow \star \leftarrow 1 \leftarrow 0 \leftarrow 2 \leftarrow 1 \leftarrow \cdots \]
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 $P : \mathbb{Z} \to \text{Bool}$. 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 $\cdot \xleftarrow{n-1} \cdot \xrightarrow{n} \cdot$
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 $P : \mathbb{Z} \to \text{Bool}$. We take the bijection $f : A \to B$ such that

1. if $\neg P(n)$ then \( \cdot \xrightarrow{n} \cdot \xleftarrow{n} \cdot \)
2. if $P(n)$ then \( \cdot \xleftarrow{n} \cdot \xleftarrow{n} \cdot \)
3. we link \( \cdot \xleftarrow{n-1} \cdot \xrightarrow{n} \cdot \)

Thus

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

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