Induction on Equality

Anti

Mathcamp 2011

The natural numbers

$$0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12 \ldots$$

The natural numbers

$$0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12 \ldots$$

Historical note

The positive natural numbers are as old as mathematics. The natural number 0 was used by Ptolemy around A.D. 140, by the Mayans in the 1st century B.C., and by Chinese mathematicians in the 4th century B.C. Nevertheless, you will occasionally run into people who have not adapted to this modern development and do not include 0 in the natural numbers.

The natural numbers

$$0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12 \ldots$$

Some properties of natural numbers:

- We can add them: $3 + 4 = 7$.

The natural numbers

$$0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12 \ldots$$

Some properties of natural numbers:

- We can add them: $3 + 4 = 7$.
- We can multiply them: $2 \cdot 4 = 8$.

The natural numbers

$$0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12 \dots$$

Some properties of natural numbers:

- We can add them: $3 + 4 = 7$.
- We can multiply them: $2 \cdot 4 = 8$.
- Addition and multiplication are well-behaved: $a + b = b + a$, $a + (b + c) = (a + b) + c$, and so on.

The natural numbers

$$0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12 \dots$$

Some properties of natural numbers:

- We can add them: $3 + 4 = 7$.
- We can multiply them: $2 \cdot 4 = 8$.
- Addition and multiplication are well-behaved: $a + b = b + a$, $a + (b + c) = (a + b) + c$, and so on.
- The pigeonhole principle

The natural numbers

$$0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12 \dots$$

Some properties of natural numbers:

- We can add them: $3 + 4 = 7$.
- We can multiply them: $2 \cdot 4 = 8$.
- Addition and multiplication are well-behaved: $a + b = b + a$, $a + (b + c) = (a + b) + c$, and so on.
- The pigeonhole principle
- Unique factorization into primes

The natural numbers

$$0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12 \ldots$$

Some properties of natural numbers:

- We can add them: $3 + 4 = 7$.
- We can multiply them: $2 \cdot 4 = 8$.
- Addition and multiplication are well-behaved: $a + b = b + a$, $a + (b + c) = (a + b) + c$, and so on.
- The pigeonhole principle
- Unique factorization into primes
- Quadratic Reciprocity

The natural numbers

$$0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12 \ldots$$

Some properties of natural numbers:

- We can add them: $3 + 4 = 7$.
- We can multiply them: $2 \cdot 4 = 8$.
- Addition and multiplication are well-behaved: $a + b = b + a$, $a + (b + c) = (a + b) + c$, and so on.
- The pigeonhole principle
- Unique factorization into primes
- Quadratic Reciprocity
- Fermat's Last Theorem

The most important property of the natural numbers

The most important property of the natural numbers

INDUCTION!

The most important property of the natural numbers

INDUCTION!

The principle of induction on natural numbers

To prove that every natural number has property P, it suffices to

1. Prove that 0 has property P, and
2. Prove that for all n, if n has property P, then so does $n + 1$.

This is the fundamental property of the natural numbers.

An example

Claim

$1 + 2 + 3 + \cdots + n = \frac{n(n+1)}{2}$ for all n.

An example

Claim

$1 + 2 + 3 + \cdots + n = \frac{n(n+1)}{2}$ for all n.

Proof

By induction, it suffices to

1. Prove that $0 = \frac{0(0+1)}{2}$, and
2. Prove that if $1 + 2 + \cdots + n = \frac{n(n+1)}{2}$, then $1 + 2 + \cdots + (n+1) = \frac{(n+1)(n+2)}{2}$.

An example

Claim

$1+2+3+\cdots+n=\frac{n(n+1)}{2}$ for all n.

Proof

By induction, it suffices to

1. Prove that $0=\frac{0(0+1)}{2}$, and
2. Prove that if $1+2+\cdots+n=\frac{n(n+1)}{2}$, then $1+2+\cdots+(n+1)=\frac{(n+1)(n+2)}{2}$.

The first is obvious. For the second, we have

$$\begin{aligned} 1+2+\cdots+(n+1) &= (1+2+\cdots+n)+(n+1) \\ &= \frac{n(n+1)}{2}+(n+1) \\ &= \frac{(n+1)(n+2)}{2} \end{aligned}$$

Equality

$$a = b$$

Equality

$$a = b$$

Historical note

The notion of equality is as old as mathematics, but the symbol "=" was first used by Robert Recorde in A.D. 1557. His justification was that "no two things can be more equal" than a pair of parallel lines.

Equality

$$a = b$$

Historical note

The notion of equality is as old as mathematics, but the symbol "$=$" was first used by Robert Recorde in A.D. 1557. His justification was that "no two things can be more equal" than a pair of parallel lines. ...thus showing that confusion about the nature of equality is also as old as mathematics.

Equality

$$a = b$$

Some properties of equality:

- Reflexivity: $a = a$.

Equality

$$a = b$$

Some properties of equality:

- Reflexivity: $a = a$.
- Symmetry: if $a = b$, then $b = a$.

Equality

$$a = b$$

Some properties of equality:

- Reflexivity: $a = a$.
- Symmetry: if $a = b$, then $b = a$.
- Transitivity: if $a = b$ and $b = c$, then $a = c$.

The most important property of equality

??

The most important property of equality

SUBSTITUTION!

The principle of substitution

To prove that every a, b with $a = b$ have property P, it suffices to

1. Prove that every pair a, a (for which $a = a$) has property P.

Note: P is a property of *pairs of equal things*; we could write it as $P(a, b)$ for clarity.

The most important property of equality

The principle of substitution

To prove that every a, b with $a = b$ have property P, it suffices to

1. Prove that every pair a, a (for which $a = a$) has property P.

Note: P is a property of *pairs of equal things*; we could write it as $P(a, b)$ for clarity.

Example

- Let's prove symmetry: if $a = b$, then $b = a$.

The most important property of equality

The principle of substitution

To prove that every a, b with $a = b$ have property P, it suffices to

1. Prove that every pair a, a (for which $a = a$) has property P.

Note: P is a property of *pairs of equal things*; we could write it as $P(a, b)$ for clarity.

Example

- Let's prove symmetry: if $a = b$, then $b = a$.
- Let $P(a, b)$ be "$b = a$"; then we want to prove that every pair a, b with $a = b$ has property $P(a, b)$.

The most important property of equality

The principle of substitution

To prove that every a, b with $a = b$ have property P, it suffices to

1. Prove that every pair a, a (for which $a = a$) has property P.

Note: P is a property of *pairs of equal things*; we could write it as $P(a, b)$ for clarity.

Example

- Let's prove symmetry: if $a = b$, then $b = a$.
- Let $P(a, b)$ be "$b = a$"; then we want to prove that every pair a, b with $a = b$ has property $P(a, b)$.
- By substitution, it suffices to prove that every pair a, a with $a = a$ has property $P(a, a)$, i.e. that $a = a$.

The most important property of equality

The principle of substitution

To prove that every a, b with $a = b$ have property P, it suffices to

1. Prove that every pair a, a (for which $a = a$) has property P.

Note: P is a property of *pairs of equal things*; we could write it as $P(a, b)$ for clarity.

Example

- Let's prove symmetry: if $a = b$, then $b = a$.
- Let $P(a, b)$ be "$b = a$"; then we want to prove that every pair a, b with $a = b$ has property $P(a, b)$.
- By substitution, it suffices to prove that every pair a, a with $a = a$ has property $P(a, a)$, i.e. that $a = a$.
- But this is obvious.

The most important property of equality

The principle of substitution

To prove that every a, b with $a = b$ have property P, it suffices to

1. Prove that every pair a, a (for which $a = a$) has property P.

Note: P is a property of *pairs of equal things*; we could write it as $P(a, b)$ for clarity.

Example

- Let's prove symmetry: if $a = b$, then $b = a$.
- Let $P(a, b)$ be "$b = a$"; then we want to prove that every pair a, b with $a = b$ has property $P(a, b)$.
- By substitution, it suffices to prove that every pair a, a with $a = a$ has property $P(a, a)$, i.e. that $a = a$.
- But this is obvious.

The most important property of equality

The principle of substitution

To prove that every a, b with $a = b$ have property P, it suffices to

1. Prove that every pair a, a (for which $a = a$) has property P.

Note: P is a property of *pairs of equal things*; we could write it as $P(a, b)$ for clarity.

Example

- Let's prove symmetry: if $a = b$, then $b = a$.
- Let $P(a, b)$ be "$b = a$"; then we want to prove that every pair a, b with $a = b$ has property $P(a, b)$.
- By substitution, it suffices to prove that every pair a, a with $a = a$ has property $P(a, a)$, i.e. that $a = a$.
- But this is obvious.

Exercise: Prove transitivity.

An analogy

Natural numbers	Equality
0 and “$+1$”	Reflexivity $a = a$
Induction	Substitution

An analogy

Natural numbers	Equality
0 and "+1"	Reflexivity $a = a$
Induction	Substitution

This may not look to you like a very strong analogy. Recognizing it was a stroke of genius by Per Martin-Löf.

Our goal is to find a framework that makes it precise.

The other most important property of natural numbers

The principle of recursion on natural numbers

To construct something for every natural number, it suffices to

1. Construct that thing for 0, and
2. Given that it has been constructed for n, construct it for $n+1$.

The other most important property of natural numbers

The principle of recursion on natural numbers

To construct something for every natural number, it suffices to

1. Construct that thing for 0, and
2. Given that it has been constructed for n, construct it for $n+1$.

Example

Define a sequence (b_n), for natural numbers n, recursively as follows:

1. $b_0 = 0$
2. $b_{n+1} = b_n + n$

We obtain $0, 1, 3, 6, 10, 15, 21, \ldots$.

The Fibonacci sequence

$$F_0 = F_1 = 1$$
$$F_{n+2} = F_{n+1} + F_n$$

Can we define this using the principle of recursion as stated?

The Fibonacci sequence

$$F_0 = F_1 = 1$$
$$F_{n+2} = F_{n+1} + F_n$$

Can we define this using the principle of recursion as stated?

Yes! We define an ordered pair of numbers p_n, recursively for each n as follows:

$$p_0 = (1, 1)$$
$$\text{If } p_n = (x, y), \text{ then } p_{n+1} = (y, x + y)$$

We obtain the sequence

$$(1, 1), (1, 2), (2, 3), (3, 5), (5, 8), \ldots$$

so that $p_n = (F_n, F_{n+1})$.

Induction vs. Recursion

Induction vs. Recursion

Recursion implies Induction

To prove something about every natural number n is the same as to construct a proof for every n. Thus, using recursion, we can

1. Construct a proof for 0, and
2. Given that a proof for n has been constructed, construct a proof for $n + 1$.

But this is just induction.

Induction vs. Recursion

Recursion implies Induction

To prove something about every natural number n is the same as to construct a proof for every n. Thus, using recursion, we can

1. Construct a proof for 0, and
2. Given that a proof for n has been constructed, construct a proof for $n + 1$.

But this is just induction.

Induction implies Recursion

Basic idea: prove by induction on n that "we can construct such-and-such for all natural numbers up to n."

So they're really two faces of the same thing.

Why does induction work?

Suppose you met someone who had never encountered the natural numbers, and you wanted to tell him how to construct them. What would you say?

Why does induction work?

Suppose you met someone who had never encountered the natural numbers, and you wanted to tell him how to construct them. What would you say?

To construct a natural number, either

1. Take the symbol 0, or
2. Take the symbol $S(n)$, where n is a natural number you have already constructed.

The symbol S is pronounced "successor".
(We also have shorthand symbols such as $1 = S(0)$, $2 = S(S(0))$, and $3 = S(S(S(0)))$, but let's not confuse our friend with those.)

This is why induction and recursion work: by definition, the natural numbers are exactly the things that we can "reach" by starting from zero and applying the successor.

Lists

Definition

A list of real numbers is either

1. the symbol nil, or
2. the symbol $a :: \ell$, where a is a real number and ℓ is a list of real numbers previously constructed.

For example,

$$8 :: \pi :: \sqrt{2} :: \mathrm{nil} = 8 :: (\pi :: (\sqrt{2} :: \mathrm{nil}))$$

is a list with three elements.

Lists

Definition

A list of real numbers is either

1. the symbol nil, or
2. the symbol $a :: \ell$, where a is a real number and ℓ is a list of real numbers previously constructed.

For example,

$$8 :: \pi :: \sqrt{2} :: \mathrm{nil} = 8 :: (\pi :: (\sqrt{2} :: \mathrm{nil}))$$

is a list with three elements.

The recursion principle for lists

To construct something for every list of real numbers, it suffices to

1. construct it for the empty list nil, and
2. given that it has been constructed for ℓ, construct it for $a :: \ell$.

Functions defined by recursion on lists

Example

The length of a list is defined recursively by

1. $\mathrm{length}(\mathrm{nil}) = 0$
2. $\mathrm{length}(a :: \ell) = 1 + \mathrm{length}(\ell)$.

Functions defined by recursion on lists

Example

The length of a list is defined recursively by

1. $\mathrm{length}(\mathrm{nil}) = 0$
2. $\mathrm{length}(a :: \ell) = 1 + \mathrm{length}(\ell)$.

Thus

$$\begin{aligned}\mathrm{length}(8 :: \pi :: \sqrt{2} :: \mathrm{nil}) &= 1 + \mathrm{length}(\pi :: \sqrt{2} :: \mathrm{nil})\\ &= 1 + 1 + \mathrm{length}(\sqrt{2} :: \mathrm{nil})\\ &= 1 + 1 + 1 + \mathrm{length}(\mathrm{nil})\\ &= 1 + 1 + 1 + 0\\ &= 3.\end{aligned}$$

Functions defined by recursion on lists

Example

The concatenation $\ell_1 ++ \ell_2$ of two lists is defined recursively by

1. $\mathrm{nil} ++ \ell = \ell$
2. $(a :: \ell_1) ++ \ell_2 = a :: (\ell_1 ++ \ell_2)$.

Functions defined by recursion on lists

Example

The concatenation $\ell_1 +\!\!+ \ell_2$ of two lists is defined recursively by

1. $\mathrm{nil} +\!\!+ \ell = \ell$
2. $(a :: \ell_1) +\!\!+ \ell_2 = a :: (\ell_1 +\!\!+ \ell_2)$.

Thus

$$\begin{aligned}\Big(4 :: 2 :: \mathrm{nil}\Big) +\!\!+ \Big(\sqrt{3} :: \mathrm{nil}\Big) &= 4 :: \Big((2 :: \mathrm{nil}) +\!\!+ (\sqrt{3} :: \mathrm{nil})\Big)\\ &= 4 :: 2 :: \Big(\mathrm{nil} +\!\!+ (\sqrt{3} :: \mathrm{nil})\Big)\\ &= 4 :: 2 :: \sqrt{3} :: \mathrm{nil}\end{aligned}$$

A theorem proven by induction on lists

Theorem

$$\mathrm{length}(\ell_1 \mathbin{+\!\!+} \ell_2) = \mathrm{length}(\ell_1) + \mathrm{length}(\ell_2)$$

A theorem proven by induction on lists

Theorem

$\mathrm{length}(\ell_1 \mathbin{+\!\!+} \ell_2) = \mathrm{length}(\ell_1) + \mathrm{length}(\ell_2)$

Proof.

By induction on ℓ_1.

1. $\mathrm{length}(\mathrm{nil} \mathbin{+\!\!+} \ell_2) = \mathrm{length}(\ell_2) = 0 + \mathrm{length}(\ell_2)$.
2. Suppose $\mathrm{length}(\ell_1 \mathbin{+\!\!+} \ell_2) = \mathrm{length}(\ell_1) + \mathrm{length}(\ell_2)$. Then

$$\begin{aligned}
\mathrm{length}\Big((a :: \ell_1) \mathbin{+\!\!+} \ell_2\Big) &= \mathrm{length}\Big(a :: (\ell_1 \mathbin{+\!\!+} \ell_2)\Big) \\
&= 1 + \mathrm{length}\big(\ell_1 \mathbin{+\!\!+} \ell_2\big) \\
&= 1 + \mathrm{length}(\ell_1) + \mathrm{length}(\ell_2) \\
&= \mathrm{length}\big(a :: \ell_1\big) + \mathrm{length}(\ell_2).
\end{aligned}$$

□

A theorem proven by induction on lists

Theorem

$\mathrm{length}(\ell_1 +\!\!+\, \ell_2) = \mathrm{length}(\ell_1) + \mathrm{length}(\ell_2)$

Proof.

By induction on ℓ_1.

1. $\mathrm{length}(\mathrm{nil} +\!\!+\, \ell_2) = \mathrm{length}(\ell_2) = 0 + \mathrm{length}(\ell_2)$.
2. Suppose $\mathrm{length}(\ell_1 +\!\!+\, \ell_2) = \mathrm{length}(\ell_1) + \mathrm{length}(\ell_2)$. Then

$$\begin{aligned}\mathrm{length}\Big((a :: \ell_1) +\!\!+\, \ell_2\Big) &= \mathrm{length}\Big(a :: (\ell_1 +\!\!+\, \ell_2)\Big)\\ &= 1 + \mathrm{length}\big(\ell_1 +\!\!+\, \ell_2\big)\\ &= 1 + \mathrm{length}(\ell_1) + \mathrm{length}(\ell_2)\\ &= \mathrm{length}\big(a :: \ell_1\big) + \mathrm{length}(\ell_2).\end{aligned}$$

□

A theorem proven by induction on lists

Theorem

$\mathrm{length}(\ell_1 +\!\!+ \ell_2) = \mathrm{length}(\ell_1) + \mathrm{length}(\ell_2)$

Proof.

By induction on ℓ_1.

1. $\mathrm{length}(\mathrm{nil} +\!\!+ \ell_2) = \mathrm{length}(\ell_2) = 0 + \mathrm{length}(\ell_2)$.
2. Suppose $\mathrm{length}(\ell_1 +\!\!+ \ell_2) = \mathrm{length}(\ell_1) + \mathrm{length}(\ell_2)$. Then

$$\begin{aligned}
\mathrm{length}\Big((a :: \ell_1) +\!\!+ \ell_2\Big) &= \mathrm{length}\Big(a :: (\ell_1 +\!\!+ \ell_2)\Big) \\
&= 1 + \mathrm{length}\big(\ell_1 +\!\!+ \ell_2\big) \\
&= 1 + \mathrm{length}(\ell_1) + \mathrm{length}(\ell_2) \\
&= \mathrm{length}\big(a :: \ell_1\big) + \mathrm{length}(\ell_2).
\end{aligned}$$

□

A theorem proven by induction on lists

Theorem

$\mathrm{length}(\ell_1 +\!\!+\, \ell_2) = \mathrm{length}(\ell_1) + \mathrm{length}(\ell_2)$

Proof.

By induction on ℓ_1.

1. $\mathrm{length}(\mathrm{nil} +\!\!+\, \ell_2) = \mathrm{length}(\ell_2) = 0 + \mathrm{length}(\ell_2)$.
2. Suppose $\mathrm{length}(\ell_1 +\!\!+\, \ell_2) = \mathrm{length}(\ell_1) + \mathrm{length}(\ell_2)$. Then

$$\begin{aligned}\mathrm{length}\Big((a :: \ell_1) +\!\!+\, \ell_2\Big) &= \mathrm{length}\Big(a :: (\ell_1 +\!\!+\, \ell_2)\Big)\\ &= 1 + \mathrm{length}\big(\ell_1 +\!\!+\, \ell_2\big)\\ &= 1 + \mathrm{length}(\ell_1) + \mathrm{length}(\ell_2)\\ &= \mathrm{length}\big(a :: \ell_1\big) + \mathrm{length}(\ell_2).\end{aligned}$$

□

A theorem proven by induction on lists

Theorem

$\text{length}(\ell_1 \mathbin{+\!\!+} \ell_2) = \text{length}(\ell_1) + \text{length}(\ell_2)$

Proof.

By induction on ℓ_1.

1. $\text{length}(\text{nil} \mathbin{+\!\!+} \ell_2) = \text{length}(\ell_2) = 0 + \text{length}(\ell_2)$.
2. Suppose $\text{length}(\ell_1 \mathbin{+\!\!+} \ell_2) = \text{length}(\ell_1) + \text{length}(\ell_2)$. Then

$$\begin{aligned}
\text{length}\Big((a :: \ell_1) \mathbin{+\!\!+} \ell_2\Big) &= \text{length}\Big(a :: (\ell_1 \mathbin{+\!\!+} \ell_2)\Big) \\
&= 1 + \text{length}\big(\ell_1 \mathbin{+\!\!+} \ell_2\big) \\
&= 1 + \text{length}(\ell_1) + \text{length}(\ell_2) \\
&= \text{length}\big(a :: \ell_1\big) + \text{length}(\ell_2).
\end{aligned}$$

□

Inductive definitions

An inductive definition of Xs consists of a finite number of constructors, each of which gives us a way to build new Xs out of things we have (possibly including already-constructed Xs).

Inductive definitions

An inductive definition of Xs consists of a finite number of constructors, each of which gives us a way to build new Xs out of things we have (possibly including already-constructed Xs).

Example

The natural numbers have two constructors.

1. 0 is a constructor which requires no input.
2. S is a constructor which requires one already-constructed natural number as input.

Inductive definitions

An inductive definition of Xs consists of a finite number of constructors, each of which gives us a way to build new Xs out of things we have (possibly including already-constructed Xs).

Example

The natural numbers have two constructors.

1. 0 is a constructor which requires no input.
2. S is a constructor which requires one already-constructed natural number as input.

Example

Lists have two constructors.

1. nil is a constructor which requires no input.
2. $::$ is a constructor which requires as input both a real number a and an already-constructed list ℓ.

Non-inductive inductive definitions

Example

The collection of glorps has two constructors.

1. left is a constructor which requires one real number as input.
2. right is a constructor which requires one orientable two-dimensional surface as input.

Non-inductive inductive definitions

Example

The collection of glorps has two constructors.

1. left is a constructor which requires one real number as input.
2. right is a constructor which requires one orientable two-dimensional surface as input.

The principle of recursion/induction for glorps

To construct/prove something for all glorps, it suffices to

1. Construct/prove it for glorps of the form $\mathrm{left}(x)$, for a real number x, and
2. Construct/prove it for glorps of the form $\mathrm{right}(S)$, for an orientable two-dimensional surface S.

Non-inductive inductive definitions

Example

The collection of glorps has two constructors.

1. left is a constructor which requires one real number as input.
2. right is a constructor which requires one orientable two-dimensional surface as input.

The principle of recursion/induction for glorps

To construct/prove something for all glorps, it suffices to

1. Construct/prove it for glorps of the form $\mathrm{left}(x)$, for a real number x, and
2. Construct/prove it for glorps of the form $\mathrm{right}(S)$, for an orientable two-dimensional surface S.

(Glorps form the disjoint union of real numbers and surfaces.)

Non-inductive inductive definitions

Example

The collection of snarks has one constructor.

1. $\star$ is a constructor which requires no input.

Non-inductive inductive definitions

Example

The collection of snarks has one constructor.

1. $\star$ is a constructor which requires no input.

The principle of recursion/induction for snarks

To construct/prove something for all snarks, it suffices to

1. Construct/prove it for $\star$.

Non-inductive inductive definitions

Example

The collection of boojums has no constructors.

Non-inductive inductive definitions

Example

The collection of boojums has no constructors.

The principle of recursion/induction for boojums

To construct/prove something for all boojums, it suffices to

Non-inductive inductive definitions

Example

The collection of boojums has no constructors.

The principle of recursion/induction for boojums

To construct/prove something for all boojums, it suffices to . . . do nothing.

Non-inductive inductive definitions

Example

The collection of boojums has no constructors.

The principle of recursion/induction for boojums

To construct/prove something for all boojums, it suffices to
. . . do nothing.

Boojums are the elements of the empty set $\emptyset$, the set with no elements. We can assert anything we like about all elements of the empty set, and it will be vacuously true.

A subtle example

Example

The collection of pseudonatural numbers has one constructor.

1. $\widehat{s}$ is a constructor which requires one already-constructed pseudonatural number as input.

A subtle example

Example

The collection of pseudonatural numbers has one constructor.

1. $\widehat{s}$ is a constructor which requires one already-constructed pseudonatural number as input.

There are no actual pseudonatural numbers! There is no way to "get started".

A subtle example

Example

The collection of pseudonatural numbers has one constructor.

1. $\widehat{s}$ is a constructor which requires one already-constructed pseudonatural number as input.

There are no actual pseudonatural numbers! There is no way to "get started".
We can make this precise:

Theorem

There is a function f from pseudonatural numbers to boojums.

Proof.

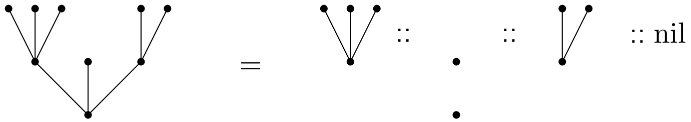
By recursion, it suffices to define $f(\widehat{s}(p))$, assuming that p is a pseudonatural number for which $f(p)$ has already been defined. But we can just define $f(\widehat{s}(p)) = f(p)$. □

Mutually inductive families

We can also inductively define several things at once. In this case, our constructors may take as input any of the already-constructed things being defined.

Example

Consider trees: a tree consists of a root node together with a list of trees (its branches).

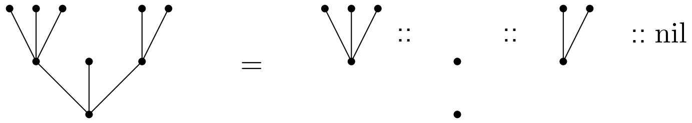


Mutually inductive families

We can also inductively define several things at once. In this case, our constructors may take as input any of the already-constructed things being defined.

Example

Consider trees: a tree consists of a root node together with a list of trees (its branches).



But the notion of "list of trees" should also be defined inductively, and requires the notion of tree! We can resolve this by defining both types at once.

Mutually inductive definitions

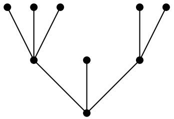
Definition

The collection of trees has one constructor.

1. node is a constructor which requires one already-constructed forest as input.

The collection of forests (lists of trees) has two constructors.

1. nil is a constructor which requires no input.
2. $::$ is a constructor which requires as input both an already-constructed tree and an already-constructed forest.



$$= \quad \mathrm{node}\left(\begin{array}{l} \mathrm{node}\Big(\mathrm{node}(\mathrm{nil}) :: \mathrm{node}(\mathrm{nil}) :: \mathrm{node}(\mathrm{nil}) :: \mathrm{nil}\Big) \\ :: \mathrm{node}\big(\mathrm{nil}\big) \\ :: \mathrm{node}\Big(\mathrm{node}(\mathrm{nil}) :: \mathrm{node}(\mathrm{nil}) :: \mathrm{nil}\Big) \\ :: \mathrm{nil} \end{array}\right)$$

Inductive families

We can even define infinitely many things inductively at once!

Definition

The collection of lists of real numbers of length n, for every natural number n, is defined as follows.

1. nil is a constructor which requires no input, and produces a list of length 0.
2. $::$ is a constructor which requires a real number a and an already-constructed list ℓ of length n, and produces a list $(a :: \ell)$ of length $n + 1$.

Note: Rather than length being a function on the set of all lists, now we define separately "the set of lists of length n" for each n.

Inductive families

Definition

The collection of elists of real numbers of length n, for every natural number n, is defined as follows.

1. nil is a constructor which requires no input, and produces a list of length 0.
2. econs is a constructor which requires two real numbers a, b and an already-constructed elist ℓ of length n, and produces an elist "$a :: b :: \ell$" of length $n + 2$.

Inductive families

Definition

The collection of elists of real numbers of length n, for every natural number n, is defined as follows.

1. nil is a constructor which requires no input, and produces a list of length 0.
2. econs is a constructor which requires two real numbers a, b and an already-constructed elist ℓ of length n, and produces an elist "$a :: b :: \ell$" of length $n + 2$.

We have defined the notion of a "list of even length" without requiring a general definition of a "list"!

Equality, revisited

Let A be any collection of things.

Definition

The collection of proofs that $a = b$, for every $a, b \in A$, is defined as follows.

1. refl is a constructor which requires one $a \in A$ as input, and produces a proof that $a = a$.

Equality, revisited

Let A be any collection of things.

Definition

The collection of proofs that $a = b$, for every $a, b \in A$, is defined as follows.

1. refl is a constructor which requires one $a \in A$ as input, and produces a proof that $a = a$.

- Note 1: This is a non-inductive inductive family.

Equality, revisited

Let A be any collection of things.

Definition

The collection of proofs that $a = b$, for every $a, b \in A$, is defined as follows.

1. refl is a constructor which requires one $a \in A$ as input, and produces a proof that $a = a$.

- Note 1: This is a non-inductive inductive family.
- Note 2: The idea of "inductively defining a collection of proofs" is reminiscent of how we proved induction from recursion.

Equality, revisited

Let A be any collection of things.

Definition

The collection of proofs that $a = b$, for every $a, b \in A$, is defined as follows.

1. refl is a constructor which requires one $a \in A$ as input, and produces a proof that $a = a$.

- Note 1: This is a non-inductive inductive family.
- Note 2: The idea of “inductively defining a collection of proofs” is reminiscent of how we proved induction from recursion.
- The corresponding induction principle is. . .

Equality, revisited

Let A be any collection of things.

Definition

The collection of proofs that $a = b$, for every $a, b \in A$, is defined as follows.

1. refl is a constructor which requires one $a \in A$ as input, and produces a proof that $a = a$.

The principle of induction on equality proofs

To prove that for every a, b and every proof that $a = b$, property P holds, it suffices to

1. Prove that for every pair a, a and the particular proof $\mathrm{refl}(a)$ that $a = a$, property P holds.

Equality, revisited

Let A be any collection of things.

Definition

The collection of proofs that $a = b$, for every $a, b \in A$, is defined as follows.

1. refl is a constructor which requires one $a \in A$ as input, and produces a proof that $a = a$.

The principle of induction on equality proofs

To prove that for every a, b and every proof that $a = b$, property P holds, it suffices to

1. Prove that for every pair a, a and the particular proof $\mathrm{refl}(a)$ that $a = a$, property P holds.

This is (almost) exactly the principle of substitution!

A useful theorem

Theorem

$0 \neq 1$.

A useful theorem

Theorem

$0 \neq 1$.

Lemma

Suppose that f is a function from A to sets (so that each value $f(a)$ is a set). Then for any $a, b \in A$, and any proof that $a = b$, if $f(b)$ contains an element then so does $f(a)$.

A useful theorem

Theorem

$0 \neq 1$.

Lemma

Suppose that f is a function from A to sets (so that each value $f(a)$ is a set). Then for any $a, b \in A$, and any proof that $a = b$, if $f(b)$ contains an element then so does $f(a)$.

Proof of Lemma

Given a, b, and a proof that $a = b$, let property P assert that "if $f(b)$ contains an element then so does $f(a)$". By induction on equality proofs, it suffices to prove that given a and the proof $\mathrm{refl}(a)$ that $a = a$, property P holds, i.e. that if $f(a)$ contains an element then so does $f(a)$. But this is obvious.

A useful theorem

Theorem

$0 \neq 1$.

Proof.

We will show that there are no proofs that $0 = 1$. To do this, we will suppose that there is such a proof, and construct a boojum (an element of $\emptyset$), which is a contradiction.

A useful theorem

Theorem

$0 \neq 1$.

Proof.

We will show that there are no proofs that $0 = 1$. To do this, we will suppose that there is such a proof, and construct a boojum (an element of $\emptyset$), which is a contradiction.
Define a function f on the natural numbers recursively as follows:

- $f(0) =$ the set of boojums.
- $f(S(n)) =$ the set of snarks (with constructor $\star$), for any n.

A useful theorem

Theorem

$0 \neq 1$.

Proof.

We will show that there are no proofs that $0 = 1$. To do this, we will suppose that there is such a proof, and construct a boojum (an element of $\emptyset$), which is a contradiction.

Define a function f on the natural numbers recursively as follows:

- $f(0) =$ the set of boojums.
- $f(S(n)) =$ the set of snarks (with constructor $\star$), for any n.

Then $f(1)$ contains an element, namely $\star$. But if $0 = 1$, then by the lemma, $f(0)$ also contains an element, which must be a boojum. $\square$

A parting thought

Question

Is $\mathrm{refl}(a)$ the only proof that $a = a$?

A parting thought

Question

Is $\mathrm{refl}(a)$ the only proof that $a = a$?

Answer

It depends on what type of thing a is.

- $\mathrm{refl}(3)$ is the only proof that $3 = 3$.
- Similarly, there is only one proof that $\pi = \pi$, that $\sqrt{2} = \sqrt{2}$, or even that $8 :: \pi :: \sqrt{2} :: \mathrm{nil} = 8 :: \pi :: \sqrt{2} :: \mathrm{nil}$.

A parting thought

Question

Is $\mathrm{refl}(a)$ the only proof that $a = a$?

Answer

It depends on what type of thing a is.

- $\mathrm{refl}(3)$ is the only proof that $3 = 3$.
- Similarly, there is only one proof that $\pi = \pi$, that $\sqrt{2} = \sqrt{2}$, or even that $8 :: \pi :: \sqrt{2} :: \mathrm{nil} = 8 :: \pi :: \sqrt{2} :: \mathrm{nil}$.

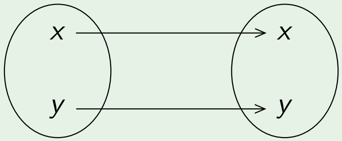
But if a is a set, a group, a topological space, a surface, an elliptic curve, a category, or another sort of mathematical *structure*, then it can be the same as itself in more than one way.

Isomorphisms

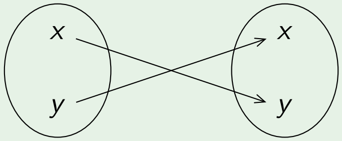
Two structures are the same if they are isomorphic, which (usually) means there is a structure-preserving bijection between them.

Example

Let $\{x, y\}$ be a set with two elements. Here is one way that $\{x, y\}$ is the same as itself:



And here is another:



Homotopy Type Theory

- Suppose p is a proof that $a = a$.
  Can there be more than one proof that $p = p$?

Homotopy Type Theory

- Suppose p is a proof that $a = a$.
  Can there be more than one proof that $p = p$?
- Now suppose q is some proof that $p = p$.
  Can there be more than one proof that $q = q$?

Homotopy Type Theory

- Suppose p is a proof that $a = a$.
  Can there be more than one proof that $p = p$?
- Now suppose q is some proof that $p = p$.
  Can there be more than one proof that $q = q$?
- Now suppose r is some proof that $q = q$.
  Can there be more than one proof that $r = r$?

Homotopy Type Theory

- Suppose p is a proof that $a = a$.
  Can there be more than one proof that $p = p$?
- Now suppose q is some proof that $p = p$.
  Can there be more than one proof that $q = q$?
- Now suppose r is some proof that $q = q$.
  Can there be more than one proof that $r = r$?
- $\ldots$

Homotopy Type Theory

- Suppose p is a proof that $a = a$.
  Can there be more than one proof that $p = p$?
- Now suppose q is some proof that $p = p$.
  Can there be more than one proof that $q = q$?
- Now suppose r is some proof that $q = q$.
  Can there be more than one proof that $r = r$?
- $\dots$

Homotopy Type Theory

- Suppose p is a proof that $a = a$.
  Can there be more than one proof that $p = p$?
- Now suppose q is some proof that $p = p$.
  Can there be more than one proof that $q = q$?
- Now suppose r is some proof that $q = q$.
  Can there be more than one proof that $r = r$?
- $\dots$

In general, the answer to all of these questions is yes!

But that's quite enough for today.