MATH 301

INTRODUCTION TO PROOFS

Sina Hazratpour
Johns Hopkins University
Fall 2021

- Recursion

Relevant sections of the textbook

- Chapter 4

Overview

1. Recursion
2. Applications of recursion theorem
3. Recursion in Lean

Recall that in the last lecture, we defined the set of natural numbers $\mathbb{N}$ to be a set *generated* by the by the number 0 and the successor function $\mathrm{succ}\colon \mathbb{N} \to \mathbb{N}$.

Recall that in the last lecture, we defined the set of natural numbers $\mathbb{N}$ to be a set *generated* by the by the number 0 and the successor function $\mathsf{succ}\colon \mathbb{N} \to \mathbb{N}$.

This means that the only way to construct a natural number is to start with 0 and apply the successor function finitely many times.

Recall that in the last lecture, we defined the set of natural numbers $\mathbb{N}$ to be a set *generated* by the by the number 0 and the successor function $\text{succ} \colon \mathbb{N} \to \mathbb{N}$.

This means that the only way to construct a natural number is to start with 0 and apply the successor function finitely many times. Therefore, natural numbers are

$$0, \text{succ}(0), \text{succ}(\text{succ}(0)), \ldots$$

Recall that in the last lecture, we defined the set of natural numbers $\mathbb{N}$ to be a set *generated* by the by the number 0 and the successor function $\mathrm{succ}\colon \mathbb{N} \to \mathbb{N}$.

This means that the only way to construct a natural number is to start with 0 and apply the successor function finitely many times. Therefore, natural numbers are

$$0, \mathrm{succ}(0), \mathrm{succ}(\mathrm{succ}(0)), \ldots$$

We also postulated the principle of induction on natural numbers.

Predicates and subsets

Recall that we can think of a predicate P on natural numbers as a function $P\colon \mathbb{N} \to \mathbf{2}$ where the set $\mathbf{2}$ consists of truth values $\perp$ and $\top$.

Predicates and subsets

Recall that we can think of a predicate P on natural numbers as a function $P\colon \mathbb{N} \to \mathbf{2}$ where the set $\mathbf{2}$ consists of truth values $\perp$ and $\top$.

Note that the set $\mathbb{N} \to \mathbf{2}$ is in bijection with $\mathcal{P}(\mathbb{N})$.

Predicates and subsets

Recall that we can think of a predicate P on natural numbers as a function $P\colon \mathbb{N} \to \mathbf{2}$ where the set $\mathbf{2}$ consists of truth values $\perp$ and $\top$.
Note that the set $\mathbb{N} \to \mathbf{2}$ is in bijection with $\mathcal{P}(\mathbb{N})$.
In one way, we construct a function

$$\eta(-)\colon (\mathbb{N} \to \mathbf{2}) \to \mathcal{P}(\mathbb{N})$$

whose value at a predicate P is the set consisting of all $n \in \mathbb{N}$ such that $P(n)$ is true, i.e.

$$\eta(P) =_{\text{def}} \{n \in \mathbb{N} \mid P(n)\}$$

Predicates and subsets

Recall that we can think of a predicate P on natural numbers as a function $P \colon \mathbb{N} \to \mathbf{2}$ where the set $\mathbf{2}$ consists of truth values $\bot$ and $\top$.
Note that the set $\mathbb{N} \to \mathbf{2}$ is in bijection with $\mathcal{P}(\mathbb{N})$.
In one way, we construct a function

$$\eta(-) \colon (\mathbb{N} \to \mathbf{2}) \to \mathcal{P}(\mathbb{N})$$

whose value at a predicate P is the set consisting of all $n \in \mathbb{N}$ such that $P(n)$ is true, i.e.

$$\eta(P) =_{\text{def}} \{n \in \mathbb{N} \mid P(n)\}$$

In the other direction, we take a subset S of $\mathbb{N}$ to the characteristic function $\chi_S \colon \mathbb{N} \to \mathbf{2}$.

The principle of induction

The principle of induction says that for any property $P\colon \mathbb{N} \to \mathbf{2}$ of natural numbers, if

1. $P(0)$ holds, and
2. whenever $P(n)$ holds then $P(n+1)$ holds,

we have that P holds of every natural number.

The principle of induction reformulated

Let $S \subseteq \mathbb{N}$ be any set of natural numbers that contains 0 and is closed under the successor operation. Then $S = \mathbb{N}$.

Proofs vs computation

We saw that the principle of induction is a very powerful tool in proving universally quantified statements about natural numbers.

Proofs vs computation

We saw that the principle of induction is a very powerful tool in proving universally quantified statements about natural numbers.

Example

- *For any finite set S, if S has n elements, then there are 2^n subsets of S.*
- *For every $n \in \mathbb{N}$, we have $0^2 + 1^2 + 2^2 + \ldots n^2 = \frac{1}{6} n (1 + n) (1 + 2n)$.*

Proofs vs computation

We saw that the principle of induction is a very powerful tool in proving universally quantified statements about natural numbers.

But, we also need to compute with natural numbers. At the very least, we should be able to define the arithmetic operations $+$, $\times$, etc.

Proofs vs computation

We saw that the principle of induction is a very powerful tool in proving universally quantified statements about natural numbers.

That is why we need another principle to help us with computation of natural numbers.This is the so-called principle of recursion which in fact can be proved from the principle of induction!

Recursion theorem

Theorem

Let A be a set. For all $a \in A$ and all $g : \mathbb{N} \times A \to A$, there is a unique function $f : \mathbb{N} \to A$ such that

1. $f(0) = a$
2. $f(\mathrm{succ}(n)) = g(n, f(n))$ *for all* $n \in \mathbb{N}$.

Recursion theorem

Theorem

Let A be a set. For all $a \in A$ and all $g : \mathbb{N} \times A \to A$, there is a unique function $f : \mathbb{N} \to A$ such that

1. $f(0) = a$
2. $f(\mathrm{succ}(n)) = g(n, f(n))$ *for all* $n \in \mathbb{N}$.

Proof.

Theorem 4.1.2 (Recursion theorem) Page 145. $\square$

Recursion theorem

Theorem

Let A be a set. For all $a \in A$ and all $g \colon \mathbb{N} \times A \to A$, there is a unique function $f \colon \mathbb{N} \to A$ such that

1. $f(0) = a$
2. $f(\mathrm{succ}(n)) = g(n, f(n))$ *for all* $n \in \mathbb{N}$.

Proof.

Theorem 4.1.2 (Recursion theorem) Page 145. □

Since for every function g such function f is uniquely determined, we write $\mathrm{rec}(g)$ for it.

Recursion theorem

Theorem

Let A be a set. For all $a \in A$ and all $g\colon \mathbb{N} \times A \to A$, there is a unique function $f\colon \mathbb{N} \to A$ such that

1. $f(0) = a$
2. $f(\mathrm{succ}(n)) = g(n, f(n))$ *for all* $n \in \mathbb{N}$.

We have

$$\mathrm{rec}(g)(0) = a$$

Recursion theorem

We have

$$\mathrm{rec}(g)(0) = a$$

$$\mathrm{rec}(g)(1) = \mathrm{rec}(g)(\mathrm{succ}(0)) = g(0, \mathrm{rec}(g)(0)) = g(0, a)$$

Recursion theorem

We have

$$\mathrm{rec}(g)(0) = a$$

$$\mathrm{rec}(g)(1) = \mathrm{rec}(g)(\mathrm{succ}(0)) = g(0, \mathrm{rec}(g)(0)) = g(0, a)$$

$$\mathrm{rec}(g)(2) = \mathrm{rec}(g)(\mathrm{succ}(1)) = g(1, \mathrm{rec}(g)(1)) = g(1, g(0, a))$$

Recursion theorem

We have

$$
\begin{aligned}
&\mathrm{rec}(g)(0) = a \\
&\mathrm{rec}(g)(1) = \mathrm{rec}(g)(\mathrm{succ}(0)) = g(0, \mathrm{rec}(g)(0)) = g(0, a) \\
&\mathrm{rec}(g)(2) = \mathrm{rec}(g)(\mathrm{succ}(1)) = g(1, \mathrm{rec}(g)(1)) = g(1, g(0, a)) \\
&\vdots
\end{aligned}
$$

Recursion, practically!

In order to specify a function $f\colon \mathbb{N} \to A$, it suffices to define $f(0)$ and, for given $n \in \mathbb{N}$, assume that $f(n)$ has been defined, and define $f(\mathrm{succ}(n))$ in terms of n and $f(n)$.

Overview

Addition by recursion

We define additions of natural numbers as a function $+\colon \mathbb{N} \times \mathbb{N} \to \mathbb{N}$.

Addition by recursion

We define additions of natural numbers as a function $+\colon \mathbb{N} \times \mathbb{N} \to \mathbb{N}$.

This means for every $m \in \mathbb{N}$, we have to define a function $m + (-)\colon \mathbb{N} \to \mathbb{N}$.

Addition by recursion

We define additions of natural numbers as a function $+ \colon \mathbb{N} \times \mathbb{N} \to \mathbb{N}$.
This means for every $m \in \mathbb{N}$, we have to define a function $m + (-) \colon \mathbb{N} \to \mathbb{N}$.
We define the latter by recursion: consider the function $g \colon \mathbb{N} \times \mathbb{N} \to \mathbb{N}$ given by the assignment $g(i, j) = \mathrm{succ}(j)$.

Addition by recursion

We define additions of natural numbers as a function $+ : \mathbb{N} \times \mathbb{N} \to \mathbb{N}$.

This means for every $m \in \mathbb{N}$, we have to define a function $m + (-) : \mathbb{N} \to \mathbb{N}$.

We define the latter by recursion: consider the function $g : \mathbb{N} \times \mathbb{N} \to \mathbb{N}$ given by the assignment $g(i, j) = \mathrm{succ}(j)$.

Choose $a = m$ in the recursion theorem. Therefore,

Addition by recursion

We define additions of natural numbers as a function $+ \colon \mathbb{N} \times \mathbb{N} \to \mathbb{N}$.

This means for every $m \in \mathbb{N}$, we have to define a function $m + (-) \colon \mathbb{N} \to \mathbb{N}$.

We define the latter by recursion: consider the function $g \colon \mathbb{N} \times \mathbb{N} \to \mathbb{N}$ given by the assignment $g(i, j) = \mathrm{succ}(j)$.

Choose $a = m$ in the recursion theorem. Therefore,

$$
\begin{aligned}
&\mathrm{rec}(g)(0) = m \\
&\mathrm{rec}(g)(1) = \mathrm{rec}(g)(\mathrm{succ}(0)) = g(0, \mathrm{rec}(g)(0)) = g(0, m) = \mathrm{succ}(m) \\
&\mathrm{rec}(g)(2) = \mathrm{rec}(g)(\mathrm{succ}(1)) = g(1, \mathrm{rec}(g)(1)) = \mathrm{succ}(\mathrm{succ}(m)) \\
&\mathrm{rec}(g)(\mathrm{succ}(n)) = g(n, \mathrm{rec}(g)(n))
\end{aligned}
$$

Addition by recursion

We now define $m + (-) : \mathbb{N} \to \mathbb{N}$ to be $\mathrm{rec}(g) : \mathbb{N} \to \mathbb{N}$.

Addition by recursion

We now define $m + (-) : \mathbb{N} \to \mathbb{N}$ to be $\mathrm{rec}(g) : \mathbb{N} \to \mathbb{N}$.

$$m + 0 = m \tag{1}$$

$$m + \mathrm{succ}(n) = \mathrm{succ}(m + n) \tag{2}$$

Addition by recursion

We now define $m + (-) : \mathbb{N} \to \mathbb{N}$ to be $\mathrm{rec}(g) : \mathbb{N} \to \mathbb{N}$.

$$m + 0 = m \tag{1}$$

$$m + \mathrm{succ}(n) = \mathrm{succ}(m + n) \tag{2}$$

Therefore,

$$m + 1 = m + \mathrm{succ}(0) = \mathrm{succ}(m + 0) = \mathrm{succ}(m) \tag{3}$$

Addition by recursion

We now define $m + (-) : \mathbb{N} \to \mathbb{N}$ to be $\mathrm{rec}(g) : \mathbb{N} \to \mathbb{N}$.

$$m + 0 = m \tag{1}$$

$$m + \mathrm{succ}(n) = \mathrm{succ}(m + n) \tag{2}$$

Therefore,

$$m + 1 = m + \mathrm{succ}(0) = \mathrm{succ}(m + 0) = \mathrm{succ}(m) \tag{3}$$

In particular,

$$1 + 1 = \mathrm{succ}(1) = \mathrm{succ}(\mathrm{succ}(1)) = 2$$

Combining recursion and induction

Proposition

For every natural numbers m and n, we have $m + n = n + m$.

Proof.

Let m to be an arbitrary natural number. we use induction on n to prove that $1 + n = n + 1$ for all $n \in \mathbb{N}$.

Combining recursion and induction

Proposition

For every natural numbers m and n, we have $m + n = n + m$.

Proof.

Let m to be an arbitrary natural number. we use induction on n to prove that $1 + n = n + 1$ for all $n \in \mathbb{N}$.

When $n = 0$, by equations (1) and (2), we have
$1 + 0 = 1 = \mathrm{succ}(0) = \mathrm{succ}(0 + 0) = 0 + \mathrm{succ}(0) = 0 + 1.$

Combining recursion and induction

Proposition

For every natural numbers m and n, we have $m + n = n + m$.

Proof.

Let m to be an arbitrary natural number. we use induction on n to prove that $1 + n = n + 1$ for all $n \in \mathbb{N}$.

When $n = 0$, by equations (1) and (2), we have

$1 + 0 = 1 = \mathrm{succ}(0) = \mathrm{succ}(0 + 0) = 0 + \mathrm{succ}(0) = 0 + 1.$

Suppose that $1 + n = n + 1$. We want to show that $1 + \mathrm{succ}(n) = \mathrm{succ}(n) + 1.$

Combining recursion and induction

Proposition

For every natural numbers m and n, we have $m + n = n + m$.

Proof.

Let m to be an arbitrary natural number. we use induction on n to prove that $1 + n = n + 1$ for all $n \in \mathbb{N}$.
When $n = 0$, by equations (1) and (2), we have
$1 + 0 = 1 = \mathrm{succ}(0) = \mathrm{succ}(0 + 0) = 0 + \mathrm{succ}(0) = 0 + 1$.
Suppose that $1 + n = n + 1$. We want to show that $1 + \mathrm{succ}(n) = \mathrm{succ}(n) + 1$.
But, by definition of function $m + -$ for $m = 1$,

$$1 + \mathrm{succ}(n) = \mathrm{succ}(1 + n) = \mathrm{succ}(n + 1) = \mathrm{succ}(\mathrm{succ}(n)) = \mathrm{succ}(n) + 1$$

The last two equations above follow from equation (3). □

Overview

1. Recursion
2. Applications of recursion theorem
3. Recursion in Lean