

# 1-3 常用的证明方法

魏恒峰

hfwei@nju.edu.cn

2017 年 10 月 30 日





# 习题选讲

- UD (第五章) 反证法 (Contradiction)
- UD (第十七章) 数学归纳法 (Mathematical Induction)
- ES (第二十四章) 鸽笼原理 (Pigeonhole Principle)



## UD 17.14: 第二数学归纳法

使用 (第一) 数学归纳法证明第二数学归纳法。

### Theorem (Cantor Theorem)

*Let  $A$  be a set.*

*If  $f : A \rightarrow 2^A$ , then  $f$  is not onto.*

ES 24.8: Longest Monotone Subsequence (留待以后的专题)



## UD 题目 17.14: 第二数学归纳法

使用 (第一) 数学归纳法证明:

### Theorem (Second Principle of Mathematical Induction)

*For an integer  $n$ , let  $Q(n)$  denote an assertion. Suppose that*

- (i)  $Q(1)$  is true and*
- (ii) for all positive integers  $n$ , if  $Q(1), \dots, Q(n)$  are true, then  $Q(n+1)$  is true.*

*Then  $Q(n)$  holds for all positive integers  $n$ .*

## Theorem (第二数学归纳法)

$$\left[ Q(1) \wedge \forall n \in \mathbb{N}^+ \left( (Q(1) \wedge \cdots \wedge Q(n)) \rightarrow Q(n+1) \right) \right] \rightarrow \forall n \in \mathbb{N}^+ Q(n).$$

## Theorem ((第一) 数学归纳法)

$$\forall P : \left[ P(1) \wedge \forall n \in \mathbb{N}^+ (P(n) \rightarrow P(n+1)) \right] \rightarrow \forall n \in \mathbb{N}^+ P(n).$$

*Let us calculate [calculamus].*

$$P(n) \triangleq Q(1) \wedge \cdots \wedge Q(n)$$



说好的数学归纳法呢？

“标准” 证明示例。

$$P(n) \triangleq Q(1) \wedge \cdots \wedge Q(n)$$

用(第一) 数学归纳法证明  $P(n)$  对一切正整数都成立。

Proof.

By mathematical induction on  $\mathbb{N}^+$ .

Basis Step  $P(1)$

Inductive Hypothesis  $P(n)$

Inductive Step  $P(n) \rightarrow P(n+1)$

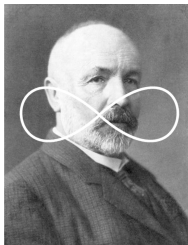
Therefore,  $P(n)$  holds for all positive integers. □



## 数学归纳法

(第一) 数学归纳法与第二数学归纳法等价。

*Q* : 为什么第二数学归纳法也被称为“强” (strong) 数学归纳法?



Georg Cantor (1845 – 1918)



David Hilbert (1862 – 1943)



Leopold Kronecker  
(1823 – 1891)



Henri Poincaré  
(1854 – 1912)



Ludwig Wittgenstein  
(1889 – 1951)

*From his paradise that Cantor with us unfolded, we hold our  
breath in awe; knowing, we shall not be expelled.*

— David Hilbert

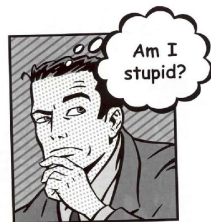
没有人能把我们从 Cantor 创造的乐园中驱逐出去。



## Theorem (Cantor Theorem)

Let  $A$  be a set.

If  $f : A \rightarrow 2^A$ , then  $f$  is not onto.



## Theorem (Cantor Theorem)

Let  $A$  be a set.

If  $f : A \rightarrow 2^A$ , then  $f$  is not onto.

Understanding this problem:

$$2^A \quad A = \{1, 2, 3\},$$

$$2^A = \{\emptyset, \{1\}, \{2\}, \{3\}, \{1, 2\}, \{1, 3\}, \{2, 3\}, \{1, 2, 3\}\}$$

Onto

$$\forall B \in 2^A \exists a \in A (f(a) = B).$$

Not Onto

$$\exists B \in 2^A \neg (\exists a \in A (f(a) = B)).$$

## Theorem (Cantor Theorem)

Let  $A$  be a set.

If  $f : A \rightarrow 2^A$ , then  $f$  is not onto.

Proof.

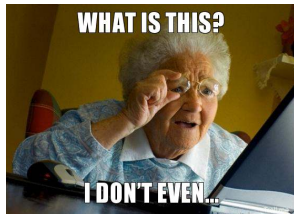
- ▶ Constructive proof:

$$B = \{x \in A \mid x \notin f(x)\}.$$

- ▶ By contradiction:

$$\exists a \in A : f(a) = B.$$

$$Q : a \in B (= f(a))?$$



## Theorem (Cantor Theorem)

Let  $A$  be a set.

If  $f : A \rightarrow 2^A$ , then  $f$  is not onto.

对角线论证 (Cantor's diagonal argument) (以下仅适用于可数集合  $A$ ).

$a$	$f(a)$					
	1	2	3	4	5	...
1	1	1	0	0	1	...
2	0	0	0	0	0	...
3	1	0	0	1	0	...
4	1	1	1	1	1	...
5	0	1	0	1	0	...
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	...

$$B = \{0, 1, 1, 0, 1\}$$

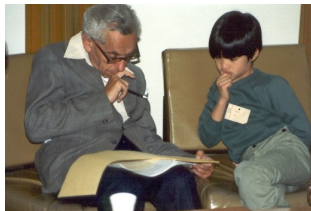


# 补充思考题



## 存在性证明 (Existence Proof)

1. 构造性证明 (Constructive proof)
2. 反证法 (By contradiction)
3. 概率法 (Probabilistic Method)



Paul Erdős (1913 – 1996)

## Theorem (Dov Jarden (1953))

$$\exists a, b \in \mathbb{R} \setminus \mathbb{Q} : a^b \in \mathbb{Q}.$$

$$\sqrt{2} \in \mathbb{R} \setminus \mathbb{Q} \quad (\text{UD: Theorem 5.2})$$

Proof.

$$\sqrt{2}^{\sqrt{2}}$$

$$(\sqrt{2}^{\sqrt{2}})^{\sqrt{2}}$$



**Q** : 这是构造性证明吗？这是反证法吗？

# Lossless Compression Algorithms



## Theorem (“No Free Lunch” Theorem)

*Any lossless (file) compression algorithm that makes some files shorter must necessarily make some files longer.*

Understanding this problem:

File  $f$ : a string of bits of a finite length  $|f|$

Compression Alg.: a function  $\mathcal{C}$

$$\mathcal{C} : F \rightarrow F$$

Lossless:  $f$  is injective

$$\mathcal{C}(f_1) = \mathcal{C}(f_2) \implies f_1 = f_2$$

$$\boxed{\forall \mathcal{C} : (\exists f \in F : |\mathcal{C}(f)| < |f|) \rightarrow (\exists f \in F : |\mathcal{C}(f)| > |f|)}$$

## Theorem (“No Free Lunch” Theorem)

*Any lossless (file) compression algorithm that makes some files shorter must necessarily make some files longer.*

$$\forall \mathcal{C} : (\exists f \in F : |\mathcal{C}(f)| < |f|) \rightarrow (\exists f \in F : |\mathcal{C}(f)| > |f|)$$

Proof.

- By contradiction

$$\exists \mathcal{C} : (\exists f \in F : |\mathcal{C}(f)| < |f|) \wedge (\forall f \in F : |\mathcal{C}(f)| \leq |f|)$$

$M$  : # of bits of a shortest file  $f$  such that  $(|\mathcal{C}(f)| = N) < (|f| = M)$

- By the pigeonhole principle

$$2^N + 1 \text{ vs. } 2^N$$



# Longest Monotone Subsequence

### Example (ES 24.8: Longest Monotone Subsequence)

Write a computer program that takes as its input a sequence of distinct integers and returns as its output the length of a longest monotone subsequence.

Understanding this problem:

Subsequence vs. substring

Monotone increasing vs. decreasing      strictly vs. non-strictly

Longest existence? uniqueness?

## ES 24.8: Longest Increasing Subsequence

- ▶ Given an integer array  $A[1 \dots n]$
- ▶ To find (the length  $L$  of) a longest increasing subsequence.

$$5, 2, 8, 6, 3, 6, 9, 7 \implies 2, 3, 6, 9$$



学生反馈： 这道题为什么放在 “Pigeonhole Principle” 这一章？



### Theorem (Erdős-Szekeres Theorem)

*Let  $n$  be a positive integer. Every sequence of  $n^2 + 1$  distinct integers must contain a monotone subsequence of length  $n + 1$ .*

$Q$ : 这道题与数学归纳法有什么关系?

B.S.  $P(1)$

I.H.  $P(n)$

I.S.  $P(n) \rightarrow P(n+1)$

$P(n)$  是什么?

$L(i)$  : the length of an LIS *ending at*  $i$ .

$$L = \max_{1 \leq i \leq n} L(i)$$

$$L(1) = 1$$

$$L(i) = 1 + \max\{L(j) : j < i \wedge A[j] < A[i]\}$$



Thank  
You!