上课之前

logic.fudan.edu.cn

上课之前

上课之前

习题课安排

上课之前

考核方式

前情提要

前情提要

本学期主要内容：

- 哥德尔第一不完全性定理
- 哥德尔第二不完全性定理

前情提要

本学期主要内容：

- 哥德尔第一不完全性定理
- 哥德尔第二不完全性定理

前情提要

哥德尔定理的数学意义：

- 希尔伯特问题
  - 1st 连续统假设问题
  - 2nd 证明算术公理的一致性
  - 10th 丢番图方程可解的判定问题
- 对“机械过程”的定义与判定问题
- 遗留问题：自然的不可判定命题

前情提要

哥德尔定理的数学意义：

- 希尔伯特问题
  - 1st 连续统假设问题
  - 2nd 证明算术公理的一致性
  - 10th 丢番图方程可解的判定问题
- 对“机械过程”的定义与判定问题
- 遗留问题：自然的不可判定命题

前情提要

哥德尔定理的数学意义：

- 希尔伯特问题
  - 1st 连续统假设问题
  - 2nd 证明算术公理的一致性
  - 10th 丢番图方程可解的判定问题
- 对“机械过程”的定义与判定问题
- 遗留问题：自然的不可判定命题

前情提要

哥德尔定理的数学意义：

- 希尔伯特问题
  - 1st 连续统假设问题
  - 2nd 证明算术公理的一致性
  - 10th 丢番图方程可解的判定问题
- 对“机械过程”的定义与判定问题
- 遗留问题：自然的不可判定命题

前情提要

哥德尔定理的数学意义：

- 希尔伯特问题
  - 1st 连续统假设问题
  - 2nd 证明算术公理的一致性
  - 10th 丢番图方程可解的判定问题
- 对“机械过程”的定义与判定问题
- 遗留问题：自然的不可判定命题

前情提要

哥德尔定理的哲学意义：

- 希尔伯特形式主义
- 希尔伯特纲领
- 哥德尔纲领
- 直觉主义数学的形式化

前情提要

哥德尔定理的哲学意义：

- 希尔伯特形式主义
- 希尔伯特纲领
- 哥德尔纲领
- 直觉主义数学的形式化

定理 (第一不完全性定理（的一个特例）)

如果 PA 是真的，即 $\mathfrak{N} \vDash \mathrm{PA}$，那么 PA 是不完全的。即存在一个算术句子 σ，$\mathrm{PA} \nvdash \sigma$ 且 $\mathrm{PA} \nvdash \neg\sigma$

证明概述

- 我们构造一个句子 σ，表示“我在 PA 中不可证”，即

$$\mathfrak{N} \vDash \sigma \Leftrightarrow \mathrm{PA} \nvdash \sigma$$

- 那么，

Case i 如果 $\mathrm{PA} \vdash \sigma$。由 $\mathfrak{N} \vDash \mathrm{PA}$，得 $\mathfrak{N} \vDash \sigma$，也即 $\mathrm{PA} \nvdash \sigma$。矛盾。故 $\mathrm{PA} \nvdash \sigma$。

Case ii 如果 $\mathrm{PA} \vdash \neg\sigma$。由 $\mathfrak{N} \vDash \mathrm{PA}$，得 $\mathfrak{N} \vDash \neg\sigma$，即 $\mathfrak{N} \nvDash \sigma$，也即 $\mathrm{PA} \vdash \sigma$，因而 $\mathfrak{N} \vDash \sigma$。矛盾。故 $\mathrm{PA} \nvdash \neg\sigma$。

证明概述

- 我们构造一个句子 σ，表示“我在 PA 中不可证”，即

$$\mathfrak{N} \vDash \sigma \Leftrightarrow \mathrm{PA} \nvdash \sigma$$

- 那么，

Case i 如果 $\mathrm{PA} \vdash \sigma$。由 $\mathfrak{N} \vDash \mathrm{PA}$，得 $\mathfrak{N} \vDash \sigma$，也即 $\mathrm{PA} \nvdash \sigma$。矛盾。故 $\mathrm{PA} \nvdash \sigma$。

Case ii 如果 $\mathrm{PA} \vdash \neg\sigma$。由 $\mathfrak{N} \vDash \mathrm{PA}$，得 $\mathfrak{N} \vDash \neg\sigma$，即 $\mathfrak{N} \nvDash \sigma$，也即 $\mathrm{PA} \vdash \sigma$，因而 $\mathfrak{N} \vDash \sigma$。矛盾。故 $\mathrm{PA} \nvdash \neg\sigma$。

证明概述

- 我们构造一个句子 σ，表示“我在 PA 中不可证”，即

$$\mathfrak{N} \vDash \sigma \Leftrightarrow \mathrm{PA} \nvdash \sigma$$

- 那么，

Case i 如果 $\mathrm{PA} \vdash \sigma$。由 $\mathfrak{N} \vDash \mathrm{PA}$，得 $\mathfrak{N} \vDash \sigma$，也即 $\mathrm{PA} \nvdash \sigma$。矛盾。故 $\mathrm{PA} \nvdash \sigma$。

Case ii 如果 $\mathrm{PA} \vdash \neg\sigma$。由 $\mathfrak{N} \vDash \mathrm{PA}$，得 $\mathfrak{N} \vDash \neg\sigma$，即 $\mathfrak{N} \nvDash \sigma$，也即 $\mathrm{PA} \vdash \sigma$，因而 $\mathfrak{N} \vDash \sigma$。矛盾。故 $\mathrm{PA} \nvdash \neg\sigma$。

证明概述

- 我们构造一个句子 σ，表示“我在 PA 中不可证”，即

$$\mathfrak{N} \vDash \sigma \Leftrightarrow \mathsf{PA} \nvdash \sigma$$

- 那么，

Case i 如果 $\mathsf{PA} \vdash \sigma$。由 $\mathfrak{N} \vDash \mathsf{PA}$，得 $\mathfrak{N} \vDash \sigma$，也即 $\mathsf{PA} \nvdash \sigma$。矛盾。故 $\mathsf{PA} \nvdash \sigma$。

Case ii 如果 $\mathsf{PA} \vdash \neg\sigma$。由 $\mathfrak{N} \vDash \mathsf{PA}$，得 $\mathfrak{N} \vDash \neg\sigma$，即 $\mathfrak{N} \nvDash \sigma$，也即 $\mathsf{PA} \vdash \sigma$，因而 $\mathfrak{N} \vDash \sigma$。矛盾。故 $\mathsf{PA} \nvdash \neg\sigma$。

关键步骤

如何在算术语言 $\mathcal{L} = (S, +, \cdot, E, 0)$ 中表达：

“我在PA中不可证”。

关键步骤

如何在算术语言 $\mathcal{L} = (S, +, \cdot, E, 0)$ 中表达：

“我在PA中不可证”。

关键步骤

如何在算术语言 $\mathcal{L} = (S, +, \cdot, E, 0)$ 中表达：

“我在PA中不可证”。

关键步骤

如何在算术语言 $\mathcal{L} = (S, +, \cdot, E, 0)$ 中表达：

“我在PA中不可证”。

为此，

我们这学期一件重要的任务是将上学期和本学期课程的主要内容，用算术的语言“说出来”。

递归论基础

为什么要讲递归论？

- 递归论严格定义了“递归的（可计算）函数/关系”
- 递归论提供了一个通用方式来刻画每个递归函数/关系
- 由此可以证明，这种刻画可以在算术语言中表达
- 由此，我们的任务转化为，证明一系列函数/关系是递归的

为什么要讲递归论？

- 递归论严格定义了“递归的（可计算）函数/关系”
- 递归论提供了一个通用方式来刻画每个递归函数/关系
- 由此可以证明，这种刻画可以在算术语言中表达
- 由此，我们的任务转化为，证明一系列函数/关系是递归的

为什么要讲递归论？

- 递归论严格定义了“递归的（可计算）函数/关系”
- 递归论提供了一个通用方式来刻画每个递归函数/关系
- 由此可以证明，这种刻画可以在算术语言中表达
- 由此，我们的任务转化为，证明一系列函数/关系是递归的

为什么要讲递归论？

- 递归论严格定义了“递归的（可计算）函数/关系”
- 递归论提供了一个通用方式来刻画每个递归函数/关系
- 由此可以证明，这种刻画可以在算术语言中表达
- 由此，我们的任务转化为，证明一系列函数/关系是递归的

术语与惯例

在递归论的讨论中：

- 除非特别说明，我们所说的“函数”、“关系”均指自然数上的函数、关系，我们说“集合”，指自然数集 $\mathbb{N}$ 的子集
- 在递归论中，我们说 f 是自然数上的 n 元函数，或写“$f\colon \mathbb{N}^n \to \mathbb{N}$”并不一定指 f 的定义域就是 $\mathbb{N}^n$，只是表示 $\operatorname{dom} f \subseteq \mathbb{N}^n$ 且 $\operatorname{ran} f \subseteq \mathbb{N}$

术语与惯例

在递归论的讨论中：

- 除非特别说明，我们所说的“函数”、“关系”均指自然数上的函数、关系，我们说“集合”，指自然数集 $\mathbb{N}$ 的子集
- 在递归论中，我们说 f 是自然数上的 n 元函数，或写“$f : \mathbb{N}^n \to \mathbb{N}$”并不一定指 f 的定义域就是 $\mathbb{N}^n$，只是表示 $\mathrm{dom}\, f \subseteq \mathbb{N}^n$ 且 $\mathrm{ran}\, f \subseteq \mathbb{N}$

哥德尔式的刻画——

递归函数

原始递归函数

定义 (初始函数)

我们首先定义 3 类初始函数（basic primitive recursion function）：

- 零函数（zero function）：0，这是一个常数或 0 元函数
- 后继函数（successor function）：$S(x) = x + 1$，对任意 $x \in \mathbb{N}$
- 投射函数（projection）：$\pi_i^n(x_1, \ldots, x_n) = x_i$，其中 $n \geq 1$、$1 \leq i \leq n$ 且 $x_1, \ldots, x_n \in \mathbb{N}$

原始递归函数

定义 (初始函数)

我们首先定义 3 类初始函数（basic primitive recursion function）：

- 零函数（zero function）：0，这是一个常数或 0 元函数
- 后继函数（successor function）：$S(x) = x + 1$，对任意 $x \in \mathbb{N}$
- 投射函数（projection）：$\pi_i^n(x_1, \ldots, x_n) = x_i$，其中 $n \geq 1$、$1 \leq i \leq n$ 且 $x_1, \ldots, x_n \in \mathbb{N}$

原始递归函数

定义 (初始函数)

我们首先定义 3 类初始函数 (basic primitive recursion function) :

- 零函数 (zero function) : 0 , 这是一个常数或 0 元函数
- 后继函数 (successor function) : $S(x) = x + 1$, 对任意 $x \in \mathbb{N}$
- 投射函数 (projection) : $\pi_i^n(x_1, \ldots, x_n) = x_i$, 其中 $n \geq 1$、$1 \leq i \leq n$ 且 $x_1, \ldots, x_n \in \mathbb{N}$

原始递归函数

定义 (初始函数)

我们首先定义 3 类初始函数 (basic primitive recursion function)：

- 零函数 (zero function)：0，这是一个常数或 0 元函数
- 后继函数 (successor function)：$S(x) = x + 1$，对任意 $x \in \mathbb{N}$
- 投射函数 (projection)：$\pi_i^n(x_1, \ldots, x_n) = x_i$，其中 $n \geq 1$、$1 \leq i \leq n$ 且 $x_1, \ldots, x_n \in \mathbb{N}$

原始递归函数

定义 (函数复合)

我们称 $f : \mathbb{N}^n \to \mathbb{N}$ 是 $g : \mathbb{N}^r \to \mathbb{N}$ 和 $h_i : \mathbb{N}^n \to \mathbb{N}$ ($1 \le i \le r$) 的复合 (composition)，当且仅当

$$f(x_1, \ldots, x_n) = g\Big(h_1(x_1, \ldots, x_n), \ldots, h_r(x_1, \ldots, x_n)\Big)$$

此时，我们记 $f = \mathrm{Cpo}(g, h_1, \ldots, h_r)$。当 g 是 1 元函数的时候，我们又记 $g \circ h = \mathrm{Cpo}(g, h)$，$f \circ g \circ h = \mathrm{Cpo}(f, \mathrm{Cpo}(g, h))$

例：$S \circ \pi_3^3$、$S \circ S \circ 0$

原始递归函数

定义 (函数复合)

我们称 $f : \mathbb{N}^n \to \mathbb{N}$ 是 $g : \mathbb{N}^r \to \mathbb{N}$ 和 $h_i : \mathbb{N}^n \to \mathbb{N}$ ($1 \le i \le r$) 的复合 (composition)，当且仅当

$$f(x_1, \ldots, x_n) = g\Big(h_1(x_1, \ldots, x_n), \ldots, h_r(x_1, \ldots, x_n)\Big)$$

此时，我们记 $f = \mathrm{Cpo}(g, h_1, \ldots, h_r)$。当 g 是 1 元函数的时候，我们又记 $g \circ h = \mathrm{Cpo}(g, h)$，$f \circ g \circ h = \mathrm{Cpo}(f, \mathrm{Cpo}(g, h))$

例：$S \circ \pi_3^3$、$S \circ S \circ 0$

原始递归函数

定义 (函数复合)

我们称 $f : \mathbb{N}^n \to \mathbb{N}$ 是 $g : \mathbb{N}^r \to \mathbb{N}$ 和 $h_i : \mathbb{N}^n \to \mathbb{N}$ ($1 \leq i \leq r$) 的复合 (composition)，当且仅当

$$f(x_1, \ldots, x_n) = g\Big(h_1(x_1, \ldots, x_n), \ldots, h_r(x_1, \ldots, x_n)\Big)$$

此时，我们记 $f = \mathsf{Cpo}(g, h_1, \ldots, h_r)$。当 g 是 1 元函数的时候，我们又记 $g \circ h = \mathsf{Cpo}(g, h)$，$f \circ g \circ h = \mathsf{Cpo}(f, \mathsf{Cpo}(g, h))$

例：$S \circ \pi_3^3$、$S \circ S \circ 0$

原始递归函数

定义 (函数复合)

我们称 $f : \mathbb{N}^n \to \mathbb{N}$ 是 $g : \mathbb{N}^r \to \mathbb{N}$ 和 $h_i : \mathbb{N}^n \to \mathbb{N}$ ($1 \leq i \leq r$) 的复合 (composition)，当且仅当

$$f(x_1, \ldots, x_n) = g\Big(h_1(x_1, \ldots, x_n), \ldots, h_r(x_1, \ldots, x_n)\Big)$$

此时，我们记 $f = \mathsf{Cpo}(g, h_1, \ldots, h_r)$。当 g 是 1 元函数的时候，我们又记 $g \circ h = \mathsf{Cpo}(g, h)$，$f \circ g \circ h = \mathsf{Cpo}(f, \mathsf{Cpo}(g, h))$

例：$S \circ \pi_3^3$、$S \circ S \circ 0$

原始递归函数

定义 (函数复合)

我们称 $f : \mathbb{N}^n \to \mathbb{N}$ 是 $g : \mathbb{N}^r \to \mathbb{N}$ 和 $h_i : \mathbb{N}^n \to \mathbb{N}$ ($1 \leq i \leq r$) 的复合 (composition)，当且仅当

$$f(x_1, \ldots, x_n) = g\Big(h_1(x_1, \ldots, x_n), \ldots, h_r(x_1, \ldots, x_n)\Big)$$

此时，我们记 $f = \mathsf{Cpo}(g, h_1, \ldots, h_r)$。当 g 是 1 元函数的时候，我们又记 $g \circ h = \mathsf{Cpo}(g, h)$，$f \circ g \circ h = \mathsf{Cpo}(f, \mathsf{Cpo}(g, h))$

例：$S \circ \pi_3^3$、$S \circ S \circ 0$

原始递归函数

定义 (函数复合)

我们称 $f : \mathbb{N}^n \to \mathbb{N}$ 是 $g : \mathbb{N}^r \to \mathbb{N}$ 和 $h_i : \mathbb{N}^n \to \mathbb{N}$ ($1 \leq i \leq r$) 的复合 (composition)，当且仅当

$$f(x_1, \ldots, x_n) = g\Big(h_1(x_1, \ldots, x_n), \ldots, h_r(x_1, \ldots, x_n)\Big)$$

此时，我们记 $f = \mathsf{Cpo}(g, h_1, \ldots, h_r)$。当 g 是 1 元函数的时候，我们又记 $g \circ h = \mathsf{Cpo}(g, h)$，$f \circ g \circ h = \mathsf{Cpo}(f, \mathsf{Cpo}(g, h))$

例：$S \circ \pi_3^3$ 、 $S \circ S \circ 0$

原始递归函数

定义 (原始递归)

对任意 $n \in \mathbb{N}$，$g : \mathbb{N}^n \to \mathbb{N}$ 以及 $h : \mathbb{N}^{n+2} \to \mathbb{N}$ 我们称 $f : \mathbb{N}^{n+1} \to \mathbb{N}$ 是从 g 和 h 经原始递归（primitive recursive）得到的，当且仅当

$$f(x_1, \ldots, x_n, 0) = g(x_1, \ldots, x_n)$$

$$f(x_1, \ldots, x_n, y+1) = h(x_1, \ldots, x_n, y, f(x_1, \ldots, x_n, y))$$

此时，我们记 $f = \mathrm{PR}(g, h)$。注意，n 可以是 0

例：从初始函数通过原始递归构造常函数：$Z(x) = 0$，令 $g = 0$，$h = \pi_2^2$

原始递归函数

定义 (原始递归)

对任意 $n \in \mathbb{N}$，$g : \mathbb{N}^n \to \mathbb{N}$ 以及 $h : \mathbb{N}^{n+2} \to \mathbb{N}$ 我们称 $f : \mathbb{N}^{n+1} \to \mathbb{N}$ 是从 g 和 h 经原始递归（primitive recursive）得到的，当且仅当

$$f(x_1, \ldots, x_n, 0) = g(x_1, \ldots, x_n)$$

$$f(x_1, \ldots, x_n, y+1) = h(x_1, \ldots, x_n, y, f(x_1, \ldots, x_n, y))$$

此时，我们记 $f = \mathrm{PR}(g, h)$。注意，n 可以是 0

例：从初始函数通过原始递归构造常函数：$Z(x) = 0$，令 $g = 0$，$h = \pi_2^2$

原始递归函数

定义 (原始递归)

对任意 $n \in \mathbb{N}$，$g : \mathbb{N}^n \to \mathbb{N}$ 以及 $h : \mathbb{N}^{n+2} \to \mathbb{N}$ 我们称 $f : \mathbb{N}^{n+1} \to \mathbb{N}$ 是从 g 和 h 经原始递归（primitive recursive）得到的，当且仅当

$$f(x_1, \ldots, x_n, 0) = g(x_1, \ldots, x_n)$$

$$f(x_1, \ldots, x_n, y+1) = h(x_1, \ldots, x_n, y, f(x_1, \ldots, x_n, y))$$

此时，我们记 $f = \mathrm{PR}(g, h)$。注意，n 可以是 0

例：从初始函数通过原始递归构造常函数：$Z(x) = 0$，令 $g = 0$，$h = \pi_2^2$

原始递归函数

定义 (原始递归)

对任意 $n \in \mathbb{N}$，$g : \mathbb{N}^n \to \mathbb{N}$ 以及 $h : \mathbb{N}^{n+2} \to \mathbb{N}$ 我们称 $f : \mathbb{N}^{n+1} \to \mathbb{N}$ 是从 g 和 h 经原始递归（primitive recursive）得到的，当且仅当

$$f(x_1, \ldots, x_n, 0) = g(x_1, \ldots, x_n)$$

$$f(x_1, \ldots, x_n, y+1) = h(x_1, \ldots, x_n, y, f(x_1, \ldots, x_n, y))$$

此时，我们记 $f = \mathrm{PR}(g, h)$。注意，n 可以是 0

例：从初始函数通过原始递归构造常函数：$Z(x) = 0$，令 $g = 0$，$h = \pi_2^2$

原始递归函数

定义 (原始递归)

对任意 $n \in \mathbb{N}$，$g : \mathbb{N}^n \to \mathbb{N}$ 以及 $h : \mathbb{N}^{n+2} \to \mathbb{N}$ 我们称 $f : \mathbb{N}^{n+1} \to \mathbb{N}$ 是从 g 和 h 经原始递归（primitive recursive）得到的，当且仅当

$$f(x_1, \ldots, x_n, 0) = g(x_1, \ldots, x_n)$$

$$f(x_1, \ldots, x_n, y+1) = h(x_1, \ldots, x_n, y, f(x_1, \ldots, x_n, y))$$

此时，我们记 $f = \mathrm{PR}(g, h)$。注意，n 可以是 0

例：从初始函数通过原始递归构造常函数：$Z(x) = 0$，令 $g = 0$，$h = \pi_2^2$

原始递归函数

定义 (原始递归函数)

- 初始函数都是原始递归函数
- 如果 $g : \mathbb{N}^r \to \mathbb{N}$ ($r \geq 1$)、$h_i : \mathbb{N}^n \to \mathbb{N}$ ($1 \leq i \leq r$) 是原始递归函数，那么 $\mathrm{Cpo}(g, h_1, \ldots, h_r)$ 是原始递归函数
- 如果 $g : \mathbb{N}^n \to \mathbb{N}$ 和 $h : \mathbb{N}^{n+2} \to \mathbb{N}$ ($n \in \mathbb{N}$) 是原始递归函数，那么 $\mathrm{PR}(g, h)$ 是原始递归函数
- 没有其它原始递归函数

原始递归函数

定义 (原始递归函数)

- 初始函数都是原始递归函数
- 如果 $g : \mathbb{N}^r \to \mathbb{N}$ ($r \geq 1$)、$h_i : \mathbb{N}^n \to \mathbb{N}$ ($1 \leq i \leq r$) 是原始递归函数，那么 $\mathrm{Cpo}(g, h_1, \ldots, h_r)$ 是原始递归函数
- 如果 $g : \mathbb{N}^n \to \mathbb{N}$ 和 $h : \mathbb{N}^{n+2} \to \mathbb{N}$ ($n \in \mathbb{N}$) 是原始递归函数，那么 $\mathrm{PR}(g, h)$ 是原始递归函数
- 没有其它原始递归函数

原始递归函数

定义 (原始递归函数)

- 初始函数都是原始递归函数
- 如果 $g : \mathbb{N}^r \to \mathbb{N}$ ($r \geq 1$)、$h_i : \mathbb{N}^n \to \mathbb{N}$ ($1 \leq i \leq r$) 是原始递归函数，那么 $\mathrm{Cpo}(g, h_1, \ldots, h_r)$ 是原始递归函数
- 如果 $g : \mathbb{N}^n \to \mathbb{N}$ 和 $h : \mathbb{N}^{n+2} \to \mathbb{N}$ ($n \in \mathbb{N}$) 是原始递归函数，那么 $\mathrm{PR}(g, h)$ 是原始递归函数
- 没有其它原始递归函数

原始递归函数

定义 (原始递归函数)

- 初始函数都是原始递归函数
- 如果 $g : \mathbb{N}^r \to \mathbb{N}$ ($r \geq 1$)、$h_i : \mathbb{N}^n \to \mathbb{N}$ ($1 \leq i \leq r$) 是原始递归函数，那么 $\mathrm{Cpo}(g, h_1, \ldots, h_r)$ 是原始递归函数
- 如果 $g : \mathbb{N}^n \to \mathbb{N}$ 和 $h : \mathbb{N}^{n+2} \to \mathbb{N}$ ($n \in \mathbb{N}$) 是原始递归函数，那么 $\mathrm{PR}(g, h)$ 是原始递归函数
- 没有其它原始递归函数

原始递归函数

定义 (原始递归函数（自下而上）)

- 初始函数都是原始递归函数
- 如果 $g : \mathbb{N}^r \to \mathbb{N}$（$r \geq 1$）、$h_i : \mathbb{N}^n \to \mathbb{N}$（$1 \leq i \leq r$）是原始递归函数，那么 $\mathrm{Cpo}(g, h_1, \ldots, h_r)$ 是原始递归函数
- 如果 $g : \mathbb{N}^n \to \mathbb{N}$ 和 $h : \mathbb{N}^{n+2} \to \mathbb{N}$（$n \in \mathbb{N}$）是原始递归函数，那么 $\mathrm{PR}(g, h)$ 是原始递归函数
- 没有其它原始递归函数

原始递归函数

定义 (原始递归函数（自上而下）)

我们定义原始递归函数（primitive recursive function）

类PRF为包涵初始函数且在 Cpo 和 PF 运算下封闭的最小的类

原始递归函数

回忆，对于递归定义的类，我们往往可以用数学归纳法来证明其中所有元素都具有某性质

引理

每个原始递归函数 $f : \mathbb{N}^n \to \mathbb{N}$ 都是全函数，即 $\operatorname{dom} f = \mathbb{N}^n$

Proof.

对原始递归函数构造序列归纳

原始递归函数

回忆，对于递归定义的类，我们往往可以用数学归纳法来证明其中所有元素都具有某性质

引理

每个原始递归函数 $f : \mathbb{N}^n \to \mathbb{N}$ 都是全函数，即 $\operatorname{dom} f = \mathbb{N}^n$

Proof.

对原始递归函数构造序列归纳

原始递归函数

回忆，对于递归定义的类，我们往往可以用数学归纳法来证明其中所有元素都具有某性质

引理

每个原始递归函数 $f : \mathbb{N}^n \to \mathbb{N}$ 都是全函数，即 $\operatorname{dom} f = \mathbb{N}^n$

Proof.

对原始递归函数构造序列归纳

原始递归函数

例：

加法函数：$f_+(x,y) = x + y$

莱布尼兹式定义：$x + 0 = x$, $x + (y + 1) = (x + y) + 1$

我们希望：

$$f_+(x, 0) = x = \pi_1^1(x)$$

$$f_+(x, y + 1) = (x + y) + 1 = S(x + y)$$

令 $g = \pi_1^1$, $h = S \circ \pi_3^3$

原始递归函数

例：

加法函数：$f_+(x,y) = x+y$

莱布尼兹式定义：$x+0 = x$, $x+(y+1) = (x+y)+1$

我们希望：

$$f_+(x,0) = x = \pi_1^1(x)$$

$$f_+(x,y+1) = (x+y)+1 = S(x+y)$$

令 $g = \pi_1^1$, $h = S \circ \pi_3^3$

原始递归函数

例：

加法函数：$f_+(x,y) = x+y$

莱布尼兹式定义：$x+0 = x$, $x+(y+1) = (x+y)+1$

我们希望：

$$f_+(x,0) = x = \pi_1^1(x)$$

$$f_+(x,y+1) = (x+y)+1 = S(x+y)$$

令 $g = \pi_1^1$, $h = S \circ \pi_3^3$

原始递归函数

例：

加法函数：$f_+(x,y) = x+y$

莱布尼兹式定义：$x+0 = x$, $x+(y+1) = (x+y)+1$

我们希望：

$$f_+(x,0) = x = \pi_1^1(x)$$

$$f_+(x,y+1) = (x+y)+1 = S(x+y)$$

令 $g = \pi_1^1$, $h = S \circ \pi_3^3$

原始递归函数

例：

加法函数：$f_+(x,y) = x + y$

莱布尼兹式定义：$x + 0 = x$, $x + (y + 1) = (x + y) + 1$

我们希望：

$$f_+(x,0) = x = \pi_1^1(x)$$

$$f_+(x,y+1) = (x+y)+1 = S(x+y)$$

令 $g = \pi_1^1$, $h = S \circ \pi_3^3$

原始递归函数

例：

加法函数：$f_+(x,y) = x + y$

莱布尼兹式定义：$x + 0 = x$, $x + (y + 1) = (x + y) + 1$

我们希望：

$$f_+(x,0) = x = \pi_1^1(x)$$

$$f_+(x,y+1) = (x+y)+1 = S(f_+(x,y))$$

令 $g = \pi_1^1$, $h = S \circ \pi_3^3$

原始递归函数

例：

加法函数：$f_+(x,y) = x+y$

莱布尼兹式定义：$x+0 = x$, $x+(y+1) = (x+y)+1$

我们希望：

$$f_+(x,0) = x = \pi_1^1(x)$$

$$f_+(x,y+1) = (x+y)+1 = S(f_+(x,y))$$

令 $g = \pi_1^1$, $h = S \circ \pi_3^3$

原始递归函数

例：

加法函数：$f_+(x,y) = x + y$

莱布尼兹式定义：$x + 0 = x$, $x + (y + 1) = (x + y) + 1$

我们希望：

$$f_+(x,0) = x = \pi_1^1(x)$$

$$f_+(x,y+1) = (x+y)+1 = S(f_+(x,y))$$

令 $g = \pi_1^1$, $h = S \circ \pi_3^3$

原始递归函数

引理

下列是原始递归函数：

- 0 元函数：$0, 1, 2, \ldots$
- 常函数 $c_k^n(x_1, \ldots, x_n) = k$
- 乘法 $x \cdot y$、幂 x^y、连乘 $x!$

原始递归函数

引理

下列是原始递归函数：

- 0 元函数：$0, 1, 2, \ldots$
- 常函数 $c_k^n(x_1, \ldots, x_n) = k$
- 乘法 $x \cdot y$、幂 x^y、连乘 $x!$

原始递归函数

引理

下列是原始递归函数：

- *0* 元函数：$0, 1, 2, \ldots$
- 常函数 $c_k^n(x_1, \ldots, x_n) = k$
- 乘法 $x \cdot y$、幂 x^y、连乘 $x!$

原始递归函数

引理

下列是原始递归函数：

- 非零检测（*non-zero test function*）：

$$\sigma(x) = \begin{cases} 0 & \text{若 } x = 0 \\ 1 & \text{若 } x \neq 0 \end{cases}$$

原始递归函数

引理

下列是原始递归函数：

- 零检测（*zero test function*），又称否定函数（*negation function*）：

$$\delta(x) = \begin{cases} 1 & \text{若 } x = 0 \\ 0 & \text{若 } x \neq 0 \end{cases}$$

原始递归函数

引理

下列是原始递归函数：

- 前驱函数（*predecessor function*）：$\mathrm{pred}(0) = 0$，$\mathrm{pred}(x+1) = x$
- 截断减法（*Monus*）：

$$x \dot{-} y = \begin{cases} 0 & \text{若 } x < y \\ x - y & \text{否则} \end{cases}$$

原始递归函数

引理

下列是原始递归函数：

- 前驱函数（*predecessor function*）：$\mathrm{pred}(0) = 0$，$\mathrm{pred}(x+1) = x$
- 截断减法（*Monus*）：

$$x \dot{-} y = \begin{cases} 0 & \text{若 } x < y \\ x - y & \text{否则} \end{cases}$$

习题：

7.1.1 - 7.1.3

证明：幻灯片中所定义的原始递归函数类与教材中所定义的仅差 0 元函数