Tutorial 1

数学归纳法溯源与公理化思维

集合论 $\Rightarrow$ 自然数 $\mathbb{N}$ $\Rightarrow$ 良序原理 $\Rightarrow$ 数学归纳法

集合论

> ZFC.7 无穷公理 Axiom of infinity
>
> $$\exists \mathbb{N} : \varnothing \in \mathbb{N} \wedge (\forall x : x \in \mathbb{N} \Rightarrow x \cup \{x\} \in \mathbb{N})$$
>
> 即存在一个集合 $\mathbb{N}$，使得空集为 $\mathbb{N}$ 的成员，并且若 x 是 $\mathbb{N}$ 的成员，则 $x \cup \{x\}$ 也是 $\mathbb{N}$ 的成员。这样的集合也叫做归纳集

由公理化的集合论可得出自然数的集合论定义

自然数

> 自然数的集合论定义
>
> 1. 定义 $\varnothing$ 为 0
> 2. 定义 n 的后继为 $n \cup \{n\}$

无穷公理确保了自然数集 $\mathbb{N}$ 存在，而上述定义也满足皮亚诺算数公理，在该定义下，自然数 n 即是包含了 n 之前所有自然数的集合

> 皮亚诺公理的非形式化叙述
>
> 1. 0 是自然数
> 2. 每个确定的自然数 x 都有一个确定的后继 x'，x' 也是自然数
> 3. 对于每个自然数 x, y，$x = y \iff x' = y'$
> 4. 0 不是任何数的后继
> 5. 任意关于自然数的命题，如果证明了它对自然数 0 是对的，又假定它对自然数 n 为真时，可以证明它对 n' 也真，那么，命题对所有自然数都真。
>
> 或：一个 **Dedekind–Peano** 结构为满足下列条件的三元组 (X, x, f)

> - X 是一个集合，x 为 X 中一个元素，f 是 X 到自身的映射
> - x 不在 f 的值域
> - f 为单射
> - 若 A 为 X 的子集且
>   - x 属于 A 且若 a 属于 A，$f(a)$ 也属于 A
>
> 则 $A = X$

良序原理

> ZFC.9 良序定理 Well-ordering theorem
>
> $$\forall X \exists R(R \text{ well-orders } X)$$
>
> 即对任意集合 X，总存在一个可良好排序 X 的二元关系 R，即 R 是 X 上的全序关系，且 X 内每个非空子集在 R 下都有一个最小元素

> **定义 A.1** 良序原理：任意非空的自然数集合必然有最小元素

良序原理对于有穷集合和无穷集合均成立

基于良序原理，即可推出数学归纳法

数学归纳法

> 超限归纳法
>
> 在良序集合中，若有一性质可从小于给定序数 α 的序数的集合传递到 α 自身，则此性质对所有序数都成立。

从良序原理可得出数学归纳法

设一组关于自然数的命题 $P(n)$

$$\{\forall k, P(1) \wedge P(2) \wedge \ldots P(k-1) \rightarrow P(k)\} \Rightarrow \forall n, P(n)$$

假设 $P(n)$ 不是对于所有自然数均成立，则 $P(n)$ 的反例集合非空，基于良序原理，存在一个最小反例 a，不妨设 $a \geqslant 2$ （即 $P(1) = \text{TRUE}$），则下列命题成立

$$\exists a \geqslant 2, P(1) \wedge P(2) \wedge P(3) \wedge \ldots P(a-1) \wedge \neg P(a) = \text{TRUE} \quad \text{(断言1)}$$

即 存在自然数不满足 $P(n) \Rightarrow$ 存在 $a \geqslant 2$ 使断言 1 成立

考虑其逆否命题，即对所有 $a \geqslant 2$，断言 1 均不成立 $\Rightarrow$ 对所有自然数，$P(n)$ 成立

$$\forall a \geqslant 2, P(1) \wedge P(2) \wedge P(3) \wedge \ldots P(a-1) \wedge \neg P(a) = \text{FALSE} \quad (\text{断言}2)$$

根据 $\neg(p \wedge \neg q) \equiv p \rightarrow q$

$$\forall a \geqslant 2, (P(1) \wedge P(2) \wedge P(3) \wedge \ldots P(a-1)) \rightarrow P(a) = \text{TRUE} \quad (\text{断言}3)$$

则若断言 3 成立，则 $P(n)$ 对所有自然数成立

由此从良序原理推出了数学归纳法

一个错误的数学归纳法证明

命题：所有的马颜色都是一样的。转换为关于自然数的命题，$P(n)$ ： n 匹马的颜色是一样的

- Base case： $P(1)$ ， 1 匹马颜色是一样的
- Induction：设 $P(n)$ 成立，对于 $P(n+1)$
  - 将 $n+1$ 匹马编号为 $h_0, h_1, h_2, \ldots, h_n$
  - 对于 $h_0, h_1, \ldots, h_{n-1}$，由 $P(n)$，他们颜色相同
  - 对于 $h_1, h_2, \ldots, h_n$，由 $P(n)$，他们颜色相同
  - 则 $h_0, h_1, h_2, \ldots, h_n$ 颜色相同，$P(n+1)$ 成立

证明错误：$P(1) \rightarrow P(2)$ 不成立

Further reading

Zermelo–Fraenkel set theory

Ordinal number

Peano axioms

Set-theoretic definition of natural numbers

极限，实数与 $\varepsilon - N$ 语言

代数系统：集合和集合上定义的封闭的二元运算

$\mathbb{N} \Rightarrow \mathbb{Z} \Rightarrow \mathbb{Q} \Rightarrow \mathbb{R} \Rightarrow \mathbb{C}$

自然数集的定义即包含了 **后继** 操作，根据后继即可定义加法

设自然数 a 的后继为 a^+，则可定义加法 $a + b^+ = (a+b)^+$

若想使减法封闭，则需将自然数集 $\mathbb{N}$ 扩展至整数集 $\mathbb{Z}$

整数集上加法，减法，乘法封闭

若想使除法封闭，则需将整数集 $\mathbb{Z}$ 扩展至有理数集 $\mathbb{Q}$

有理数集对极限不封闭

e.g.，无穷级数和

$$1-\frac{1}{3}+\frac{1}{5}-\frac{1}{7}+\frac{1}{9}-\cdots\to\frac{\pi}{4}$$

故将有理数集 $\mathbb{Q}$ 扩展至实数集 $\mathbb{R}$

而在实数集上 $x^2+1=0$ 的求解不封闭，扩展至复数集 $\mathbb{C}$

从算法的角度重新审视数学的概念

基本数学概念

自变量与单调性

对于算法分析中出现的函数，一般自变量是自然数，单调性是单调非减

取整

对于任意有理数 x ，有

$$x-1<\lfloor x\rfloor\leqslant x\leqslant\lceil x\rceil<x+1$$

对于任意整数 n ，有

$$\lceil n/2\rceil+\lfloor n/2\rfloor=n$$

且对于任意实数 $x\geqslant 0$ 和整数 $a,b>0$ ，有

$$\begin{aligned}\left\lceil\frac{\lceil x/a\rceil}{b}\right\rceil&=\left\lceil\frac{x}{ab}\right\rceil\\\left\lfloor\frac{\lfloor x/a\rfloor}{b}\right\rfloor&=\left\lfloor\frac{x}{ab}\right\rfloor\\\left\lceil\frac{a}{b}\right\rceil&\leqslant\frac{a+(b-1)}{b}\\\left\lfloor\frac{a}{b}\right\rfloor&\geqslant\frac{a-(b-1)}{b}\end{aligned}$$

取整函数可以使算法分析时的输出更加严格，如划分时中点的选择，树的高度，递归的次数等

取模

对于任意整数 a 和正整数 n ，定义取模

$$a \bmod n = a - n\lfloor a/n \rfloor$$

显然有

$$0 \leqslant a \bmod n < n$$

> 同余
>
> 两个整数 a, b 和正整数 m ，若 $a \bmod m = b \bmod m$ ，则称 a, b 对于模 m **同余**，记作
>
> $$a \equiv b \pmod m$$

多项式

> CRLS 3rd edition P.55
>
> Given a nonnegative integer d ,a **polynomial in** n **of degree** d is a function $p(n)$ of the form
>
> $$p(n) = \sum_{i=0}^{d} a_i n^i \quad (a_d \neq 0)$$

一个多项式是渐近正的 $\iff a_d > 0$ ，对于任何渐近正的 d 阶多项式 $p(n)$ ，有 $p(n) = \Theta(n^d)$

对于任意实数 $a \geqslant 0, n^a$ 单调增，对于任意实数 $a \leqslant 0, n^a$ 单调减

对数

计算机科学中的对数一般默认以 2 为底，有三类算法操作与对数 $\log n$ 有密切的联系

- 折半：划分规模为 n 的问题（D&C recursion）一般经过 $\log n$ 次划分可将问题降至常数规模
- 二叉树：
  - n 个结点的完美二叉树高度为 $\lfloor \log n \rfloor$
  - 完美二叉树与折半的关系：叶结点层的结点数是结点树的一半，折半相当于减少一层
  - 二叉树的层数划分是对正整数的等价类划分，划分依据是 $\lfloor \log n \rfloor$
- 二进制编码：整数 n 转换成二进制的比特数为 $\lfloor \log n \rfloor + 1$

阶乘

对于自然数 n

$$n! = \begin{cases} 1 & \text{if } n = 0 \\ n \cdot (n-1)! & \text{if } n > 0 \end{cases}$$

> Stirling's approximation
>
> $$n! = \sqrt{2\pi n}\left(\frac{n}{e}\right)^n \left(1 + \Theta\left(\frac{1}{n}\right)\right)$$

与阶乘相关的函数渐近增长率：

$$\begin{aligned} n! &= o(n^n) \\ n! &= \omega(2^n) \\ \log n! &= \Theta(n \log n) \end{aligned}$$

阶乘常常与排序算法的分析有关，一个 n 个元素的输入所有排列的种数即为 $n!$

常用级数求和

多项式级数：

$$\begin{aligned} \sum_{i=1}^{n} i &= \frac{n(n+1)}{2} \\ \sum_{i=1}^{n} i^2 &= \frac{1}{3}n(n+\frac{1}{2})(n+1) \\ \sum_{i=1}^{n} i^k &= \Theta(\frac{1}{k+1}n^{k+1}) \end{aligned}$$

几何级数：

$$\sum_{i=0}^{k} ar^i = a\left(\frac{r^{k+1}-1}{r-1}\right) = \Theta(r^k)$$

算数几何级数：

$$\sum_{i=1}^{k} i \cdot 2^i = (k-1)2^{k+1} + 2$$

调和级数：

$$\sum_{i=1}^{k} \frac{1}{i} = \ln k + \gamma + \varepsilon_k$$

期望值，指标随机变量，期望的线性特征

> **定义 2.1** 指标随机变量
>
> $$X_i = I\{事件\ e_i\} = \begin{cases} 1, & 事件发生 \\ 0, & 事件不发生 \end{cases}$$

显然指标随机变量的期望值就等于事件发生的概率

算法的代价可用 critical operation 的指标随机变量和表示，而算法的平均情况时间复杂度即是算法代价的数学期望。基于期望的线性特征，可以显著的简化求数学期望的过程

> 期望的线性特征：给定任意随机变量 $X_1, X_2, X_3, \ldots, X_k$ 及其线性函数 $h(X_1, X_2, X_3, \ldots, X_k)$ ，则有
>
> $$E[h(X_1, X_2, \ldots, X_k)] = h(E[X_1], E[X_2], \ldots, E[X_k])$$

蛮力算法

重点：如何从蛮力的解法到优化的解法

具体细节涉及课后习题答案故不在此处给出

微博名人问题

给定 n 个人，不关注所有人且被其他所有人关注的人称为名人，能够做的操作只有询问两个人其中一个是否关注另一个人，得到是或否的答案

- 蛮力算法：关注是一种**二元关系**，关于二元关系的问题总能得到一个时间复杂度为 $O(n^2)$ 的算法（即遍历所有的二元关系）
- 线性时间解：提示：每次询问操作可排除一个“非名人”

常见项问题特例

给定 n 个元素，求其中出现次数至少为 $\left\lceil \frac{n}{2} \right\rceil$ 的元素

- 蛮力算法：遍历数组，检查每个元素，时间复杂度 $O(n^2)$
- 线性时间解：提示：常见项的出现次数不会少于其他所有元素的出现次数之和