(写在前面: 本文的 Markdown 版本由计算机程序自动生成并投稿. 因此可能会有错误. 这系列是《具体数学》浅显的阅读笔记.)

§1 和式及其基本操作

要使得求和便于书写和分析, 我们最好 (跟随傅里叶) 引入如下的记号:$\sum$.

1.1. 和式与求和记号.

定义 1.1 (求和记号 $\sum$)**.** 对于一个可数的集合 $S=\{a_1,a_2,\cdots,a_n\}$, 求和实际上是将这个集合的每一个元素在某一个函数 $f:S\to X$ 作用之后, 把对应的值相加. 记作 $\sum_{i\in S} f(i)$; 表示 $f(a_1)+f(a_2)+f(a_3)+\cdots+f(a_n)+\cdots$.

例子 1.1. 考虑

$$\sum_{1\le k\le n} a_k = a_1+a_2+\cdots+a_n.$$
$$\sum_{\substack{1\le k\le 100\\ k \text{ odd}}} k^2 = \sum_{0\le k\le 49}(2k+1)^2.$$

a) 换元法 如果将 $\sum_{i\in S} f(i)$ 换为 $\sum_{k\in S} f(k)$, 求和的表示内容将不变. 这是因为仅仅是 "当前 S 中的代表" 用来表示的字母不同, 从而肯定不会影响整个映射 f 所表达的意思.

例子 1.2. 考虑

$$\sum_{1\le k\le n} a_k \xlongequal{k\text{代换为}s+1} \sum_{1\le s+1\le n} a_{s+1} \xlongequal{s\text{代换为}k} \sum_{1\le k+1\le n} a_{k+1}$$

注: 有时候 $\sum_{a\le i\le b} f(i)$ 也写作 $\sum_{i=a}^{b} f(i)$. 但是看到这样一来, 变量代换之后就得到了

$$\sum_{i=1}^{n} a_i \xlongequal{k\text{代为}k+1} \sum_{i=0}^{n-1} a_{i+1}$$

更易于出错.

此外, 我们用 $k:=k+1$ 来表示把 k 代为 $k+1$. 最后, 数列可以看做特殊的函数. 实际上书写 a_k 实际上就相当于 $f(k)$.

b) 求和的性质 尽管我们可以对可数无限个元素进行求和操作, 但是, 那样的数列我们在高等数学的级数部分就已经学过了. 这里先讨论有限项求和的情形.

定理 1.1. 设 K 是某一有限个正整数的集合, 我们有如下的三条规则:

- 常数项进出求和记号:
$$\sum_{k\in k} cf(k) = c\sum_{k\in k} f(k);$$
- 求和记号的拆分:
$$\sum_{k\in K} f(k)g(k) = \sum_{k\in K} f(k) + \sum_{k\in K} g(k);$$

- 若 $p(k)$ 应用于 K 中的每一个元素之后组成的集合依然是 K 的一个排列, 那么

$$\sum_{k\in K} f(k) = \sum_{p(k)\in K} f(p(k)).$$

上述定理的证明可以直接由定义得到.

例子 1.3. 如 $K=\{-1,0,1\}, p(k)=-k$, 由于 $p(-1)=1, p(0)=0, p(1)=-1, p$ 对 k 中每一个元素构成集合为 $\{-1,0,1\}=K$.

例子 1.4 (等差数列求和)**.** 求

$$S=\sum_{0\le k\le n}(a+bk)$$

的值.

考虑

$$\begin{aligned} S_2 &= \sum_{0\le k\le n}(a+bk) \\ &\overset{k:=n-k}{=\!=\!=} \sum_{0\le n-k\le n}(a+b(n-k)) = \sum_{0\le k\le n}(a+bn-bk) \end{aligned}$$

令 $S+S_2$ 得

$$\begin{aligned} S+S_2=2S &= \sum_{0\le k\le n}(a+bk)+\sum_{0\le k\le n}(a+bn-bk) \\ &= \sum_{0\le k\le n}(2a+bn)=(2a+bn)\sum_{0\le k\le n}1=(2a+bn)(n+1) \end{aligned}$$

那

$$S=\frac{(n+1)(2a+bn)}{2}.$$

1.2. Iverson 括号.

定义 1.2 (Iverson 括号)**.** 假设 P 是一个命题, 定义

$$[P]=\begin{cases}1, & \text{命题}P\text{为真}\\ 0, & \text{命题}P\text{为假}\end{cases}$$

这样的记号便于优化很复杂的求和操作, 并且可以把求和符号的下标转换为命题之间的操作.

例子 1.5. 求和式 $\sum_{0\le k\le n} k$ 可被改为 $\sum_k k[0\le k\le n]$.

如果 k 未指定限定条件, 我们认为 $k\in\mathbb{Z}$. 也就是说上述式子

$$\begin{aligned} \sum_k k[0\le k\le n] &= \cdots+(-3\cdot 0)+(-2\cdot 0)+(-1\cdot 0)+(0\cdot 0)+(1\cdot 1)+(2\cdot 1)+\cdots+(n\cdot 1) \\ &=(0\cdot 1)+(1\cdot 1)+\cdots+(n\cdot 1) \end{aligned}$$

例子 1.6. 如果 K 与 K' 是两个整数集合, 那么 $\forall k$,

$$[k \in K] + [k \in K'] = [k \in (K \cap K')] + [k \in (K \cup K')]$$

由此可以导出对应的和式

$$\sum_{k \in k} a_k + \sum_{k \in k'} a_k = \sum_{k \in k \cap k'} a_k + \sum_{k \in k U k'} a_k$$

这是一个很有用的命题. 下面我们会看到它的用途.

命题 1.2. $[k \in K] + [k \in K'] = [k \in (K \cap K')] + [k \in (K \cup K')]$ 对 k, k' 为可数集, $\forall k$.

1.3. 常见的求和方法.

a) 成套方法 这个方法类似于在解答微分方程的时候, 首先求解特解, 然后求解通解. 在这里我们不再赘述.

b) 扰动法 要计算 $S_n = \sum_{0 \le k \le n} a_k$, 可以有两种方法改写

命题 1.3. 扰动法是指

$$\begin{aligned} \boxed{S_n + a_{n+1}} = \sum_{0 \le k \le n+1} a_k &= a_0 + \sum_{1 \le k \le n+1} a_k \\ &\overset{k:=k+1}{=\!=\!=} a_0 + \sum_{1 \le k+1 \le n+1} a_k \\ &\boxed{= a_0 + \sum_{0 \le k \le n} a_k} \end{aligned}$$

例子 1.7. 用上述的方法计算等比数列. $S_n = \sum_{0 \le k \le n} ax^k$.

$$\begin{aligned} S_n + ax^{n+1} &= ax^0 + \sum_{0 \le k \le n} ax^{k+1} \\ &= ax^0 + x \sum_{0 \le k \le n} ax^k \\ &= ax^0 + xS_n \end{aligned}$$

对于 $x \neq 1$, 有

$$S_n = \sum_{k \neq 1}^{n} ax^k = \frac{a - ax^{n+1}}{1 - x}$$

例子 1.8 (等差数列乘等比数列)**.** 计算

$$S_n = \sum_{0 \le k \le n} k \cdot 2^k.$$

按照上面的方法,

$$\begin{aligned} S_n+(n+1)2^{n+1} &= 0+\sum_{0\le k\le n}(k+1)2^{k+1} \\ &= 0+\sum_{0\le k\le n}k\cdot 2^{k+1}+\sum_{0\le k\le n}2^{k+1} \\ &= 2S_n+2^{n+2}-2 \end{aligned}$$

解出 S_n, 就是

$$S_n=\sum_{0\le k\le n}k2^k=(n-1)2^{n+1}+2.$$

例子 1.9. 从上面的推导中, 我们知道 $\sum_{k=0}^n x^k=\frac{1-x^{n-1}}{1-x}$, 两边对 x 求导, 就有

$$\begin{aligned} \sum_{k=0}^n kx^{k-1} &= \frac{(1-x)\left(-(n+1)x^n\right)+1-x^{n+1}}{(1-x^2)} \\ &= \frac{1-(n+1)x^n+nx^{n+1}}{(1-x)^2}. \end{aligned}$$

同样也可以得到上一式子的结果.

c) 求和因子

例子 1.10. 由递推关系

$$\begin{aligned} T_0 &= 0 \\ T_n &= 2T_{n-1}+1 \end{aligned}$$

倘若两端同时除以 2^n, 就得到

$$\begin{aligned} T_0/2^0 &= 0 \\ T_n/2^n &= T_{n-1}/2^{n-1}+1/2^n \end{aligned}$$

令 $S=T_n/2^n$, 得 $\begin{cases} S_0=0 \\ S_n=S_{n-1}+2^{-n} \end{cases}$, 即 $S_n=\sum_{k=1}^n 2^{-k}$, 为一等比数列.

对于更为一般的式子, 如 $a_nT_n=b_nT_{n-1}+c_n$, 可变为 $S_n=S_{n-1}+S_nc_n$ 的形式.

1° 方法: 使两边同时乘以求和因子 S_n:

$$\underbrace{\boxed{s_na_nT_n}}_{:=S_n}=\underbrace{S_nb_nT_{n-1}}_{\text{必须得是}S_{n-1}=S_{n-1}a_{n-1}T_{n-1}}+S_nc_n$$

由此 $S_n=S_0a_0\cdot T_0+\sum_{k=1}^n S_kc_k=S_1b_1T_0+\sum_{k=1}^n S_kc_k$, 那么

$$T_n=\frac{1}{S_na_n}\left(S_1b_1T_0+\sum_{k=1}^n S_kc_k\right)$$

.

2° 寻找 S_n 的方法: 由于上式要满足 $S_n b_n T_{n-1} = S_{n-1} a_{n-1} T_{n-1}$, 代入展开, 有

$$\begin{aligned} s_n &= \frac{s_{n-1}a_{n-1}}{b_n} \\ &= \frac{s_{n-2}a_{n-1}a_{n-2}}{b_n b_{n-1}} = \cdots \\ &= \frac{a_{n-1}a_{n-2}\cdots a_1}{b_n b_{n-1}\cdots b_2}. \end{aligned}$$

因此我们就这样找到了求和因子.

例子 1.11. 解由快速排序带来的递归式:

$$\begin{aligned} &C_0 = 0 \\ &C_n = n + 1 + \frac{2}{n}\sum_{k=0}^{n-1} C_k \quad , n > 0. \end{aligned}$$

将 C_n 两侧同乘以 n, 得

$$nC_n = n^2 + n + 2\sum_{k=0}^{n-1} C_k \quad (n > 0).$$

令 $n := n - 1$, 有

$$(n-1)C_{n-1} = (n-1)^2 + (n-1) + 2\sum_{k=0}^{n-2} c_k.$$

上面两式相减, 有 $nC_n - (n-1)C_{n-1} = 2n + 2C_{n-1}$, 即

$$\begin{aligned} &C_0 = 0 \\ &nC_n = (n+1)C_{n-1} + 2n. \end{aligned}$$

将上述 $a_n = n, b_n = n+1, c_n = 2n$, 得 $s = \frac{(n-1)\cdots 1}{(n+1)\cdots 3} = \frac{2}{n(n+1)}$.
得

$$C_n = 2(n+1)\sum_{k=1}^{n} \frac{1}{k+1} = 2(n+1)H_{n+1}.$$