

串

Karp-Rabin算法：串即是数

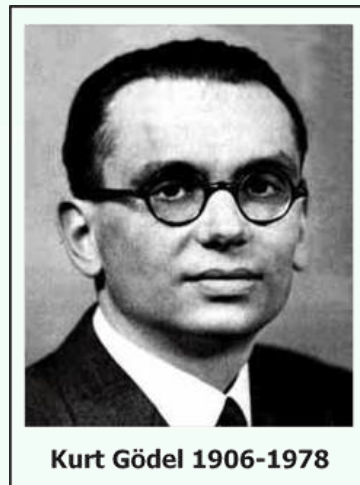
13-F1

邓俊辉

deng@tsinghua.edu.cn

All things are numbers. - Pythagoras

凡物皆数：Gödel Numbering



- ❖ 逻辑系统的符号、表达式、公式、命题、定理、公理等，均可表示为**自然数**
- ❖ 每个有限维的自然数**向量**（包括字符串），都唯一对应于某个**自然数**
- ❖ 素数序列： $p(k)$ = 第 k 个素数 = 2, 3, 5, 7, 11, 13, 17, 19, ...

$$\langle a_1, a_2, \dots, a_n \rangle \Rightarrow p(1)^{1+a_1} \times p(2)^{1+a_2} \times \dots \times p(n)^{1+a_n}$$

$$\langle \boxed{3} \quad \boxed{1} \quad \boxed{4} \quad \boxed{1} \quad \boxed{5} \quad \boxed{9} \quad \boxed{2} \quad \boxed{6} \rangle$$
$$2^{\boxed{4}} \times 3^{\boxed{2}} \times 5^{\boxed{5}} \times 7^{\boxed{2}} \times 11^{\boxed{6}} \times 13^{\boxed{10}} \times 17^{\boxed{3}} \times 19^{\boxed{7}}$$

$$\text{"gödel"} = 2^{1+7} \times 3^{1+15} \times 5^{1+4} \times 7^{1+5} \times 11^{1+12} = 139,869,560,310,664,817,087,943,919,200,000$$

- ❖ 若果真如RAM模型所假设的**字长无限**，则只需**一个**寄存器即可...

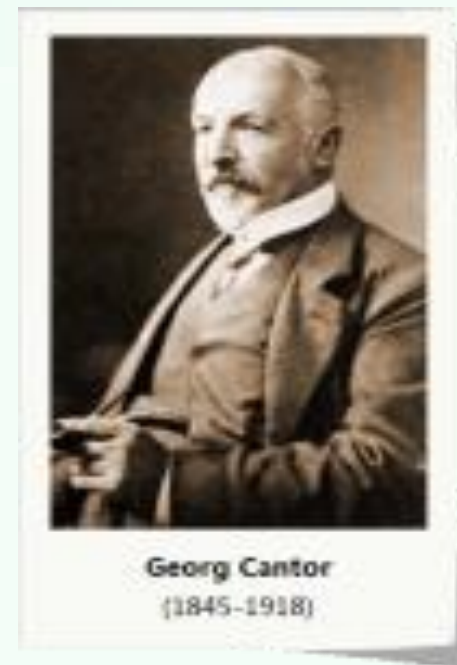
凡物皆数: Cantor Numbering

$$\text{cantor}_2(i, j) = [(i + j)^2 + 3 \cdot i + j] / 2$$

$$\text{cantor}_2(2, 3) = [(2 + 3)^2 + 3 \cdot 2 + 3] / 2 = 17$$

$$\text{cantor}_2(3, 2) = [(3 + 2)^2 + 3 \cdot 3 + 2] / 2 = 18$$

0	1	3	6	10	15
2	4	7	11	16	
5	8	12	17		
9	13	18			
14	19				
20					



$$\text{cantor}_{n+1}(a_1, \dots; a_{n-1}, a_n, a_{n+1}) =$$

$$\text{cantor}_n(a_1, \dots; a_{n-1}, \text{cantor}_2(a_n, a_{n+1}))$$

凡物皆数: Cantor Numbering

❖ 长度**有限**的字符串, 都可视作 $d=1+|\Sigma|$ 进制的自然数

"decade" = $453145_{(10)}$

// $d = 1 + ('i' - 'a') = 10$

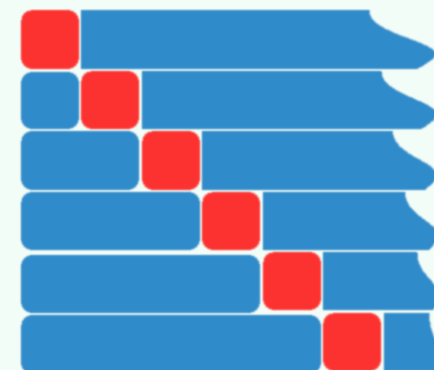
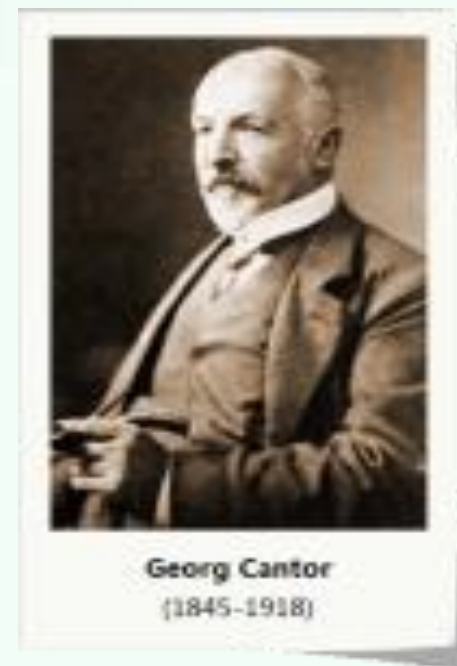
❖ 长度**无限**的字符串, 都可视作 $[0,1)$ 内的 d 进制小数

"bgahbhahbhdei..." = $0.2718281828459\dots$

"fade" = $0.614\underline{5}$ = $0.614\underline{4999\dots}$ = "fadiiii..."

❖ Cantor's Diagonal: $\aleph_0 < \aleph_1$

~~N~~



串亦为数

❖ 十进制串，可直接视作自然数 //指纹 (fingerprint) , 等效于多项式法

$P = "82818"$ $T = 271\boxed{82818}284590452353602874713527$

❖ 一般地，随意对字符编号 $\{ 0, 1, 2, \dots, d - 1 \}$ //设 $d = |\Sigma|$

于是，每个字符串都对应于一个 d 进制自然数 //尽管不是单射

$"CAT" = 2 \ 0 \ 19_{(26)} = 1371_{(10)}$ // $\Sigma = \{ A, B, C, \dots, Z \}$

$"ABBA" = 0 \ 1 \ 1 \ 0_{(26)} = 702_{(10)}$

❖ P 在 T 中出现 仅当 T 中某一子串与 P 相等 //为什么不是 “当” ?

❖ 这，不已经就是一个算法了吗? ! //具体如何实现?

❖ 问题似乎解决得很顺利，果真如此简单吗? //复杂度?