Notes of Computational Complexity — Chapter 0

ArtsEpiphany

June 3, 2022

Abstract

这是我整理的《Computational Complexity — A Modern Approach》读书笔记的 Chapter 0 部分，主要介绍了一些预备性的数学知识。

本节内容较为简单，是后续内容的基础。如果愿意，欢迎阅读接下来的部分。当然，遇到不懂的地方也不要忘了可以参考原书，毕竟我的笔记可能会遗漏自以为众所周知的内容，或是有其他的问题。翻看原书或许能有所帮助。

如果发现问题请联系 ArtsEpiphany@gmail.com。

1 一些规范

- 约定 $\mathbb{Z}$ 代表整数集。
- 约定 $\mathbb{N}$ 代表非负整数集。
- 约定 i, j, k, l, m, m 均为整数。
- 约定 $\lceil x \rceil$ 为对 x 向上取整，$\lfloor x \rfloor$ 为对 x 向下取整。在取整符号被省略时（需要一个整数，但表达式为实数）均默认向上取整。
- 当我们说“对足够大的 n 怎样怎样”时，意味着存在 N 使得对于任意大于 N 的 n，此要求都符合。
- 如果 u 为字符串或向量，那么 u_i 代表 u 的第 i 个字符或分量。

2 字符串

对于一个字符集 S，基于 S 的字符串是有限的有序的由 S 中元素构成的元组。比如，对于一个常用的字符集 $\{0,1\}$，基于它的字符串就是由 0 和 1 构成的有限长的序列，比如 0111001。

- 对于正整数 n，约定 S^n 为基于 S 的长度为 n 的字符串的集合。[1]
- 约定 $S^* = \bigcup_{n\geqslant 0} S^n$，即所有基于 S 的字符串构成的集合。
- 对于字符串 x, y，约定 $x \circ y$ 或 xy 为将 x 与 y 拼接得到的字符串。
- 对于字符串 x 和正整数 k，约定 x^k 表示 k 个 x 的拼接。
- 约定 $|x|$ 表示字符串 x 的长度。

3 其他约定

- 对于一个分布 S，约定 $x \in_{\mathrm{R}} S$ 表示 x 是一个服从分布 S 的随机变量。
- 约定 U_n 为 $\{0,1\}^n$ 中所有元素的均匀分布。
- 约定 $x \odot y$ 为向量 x 与 y 的点积对 2 取模的结果（除以 2 的余数）。
- 约定 $\langle \mathbf{u}, \mathbf{v} \rangle$ 为向量 $\mathbf{u}, \mathbf{v}$ 的内积。
- 约定 $\langle x \rangle$ 为 x（可以是数字、图灵机、问题等任何可以被字符串表示的事物）在某个字符集下对应的字符串。

我们相信，我们可以使用字符串集 $\{0,1\}^*$ 来表示所有我们需要的内容，如元组、图等。我们默认讨论的输入输出都是如此的形式。

4 判定性问题

很多问题的答案为“是”或“否”，也就是说，这个问题对应的函数 f 是一个由 $\{0,1\}^*$ 到 $\{0,1\}$ 的映射。那么集合 $L_f = \{x \mid f(x) = 1\}$（所有答案为

[1] $|S^0| = 1$，元素为空串。

“是”的输入构成的集合）就被称为语言或判定性问题。计算函数 f 也就等价于判定语言 L_f。

比如，对于上一章节中的朋友问题，我们可以将其改造成一个判定性问题：输入人们之间的关系和一个整数 k，问是否存在 k 个人两两都是朋友。[2] 答案为“是”的输入的集合就构成了一个语言，我们记为 INDSET。

5 渐进符号

为了避免执着于细节，我们需要定义渐进符号。

定义 5.1 *对于两个 $\mathbb{N}$ 到 $\mathbb{N}$ 的函数 f, g，记号 $f = O(g)$ 表明存在常数 c 使得对于足够大的 n 有 $f(n) \leqslant c \cdot g(n)$；记号 $f = \Omega(g)$ 表明 $g = O(f)$；记号 $f = \Theta(g)$ 表明 $f = O(g)$ 且 $g = O(f)$；记号 $f = o(g)$ 表明对任意 $\epsilon > 0$，对于足够大的 n 有 $f(n) \leqslant \epsilon \cdot g(n)$；记号 $f = \omega(g)$ 表明 $g = o(f)$。*

这个定义中全部忽略了常数部分，这是因为在研究中我们遇到的差异通常是像之前的乘法的例子那样的量级上的差异，我们不妨常数部分。

我们可以得出一些“对于足够大的 n”的大小关系。比如我们同时对函数得到了两个上界 $O(n^2)$ 和 $O(2^n)$，那么第二个就可以被抛弃，因为它是更宽松的。

[2] 这个问题显然不比之前的问题更难，但也不比之前的问题更简单。不比之前的问题更难是因为假设我们有能力回答之前的问题（我们就知道至多可以有多少人两两都是朋友），我们就可以回答现在这个问题。不比之前的问题更简单是因为因为一旦我们能求解这个判定性问题，我们也可以回答之前的问题：首先我们可以改变 k 的值求出这个目标集合的大小 m，接着我们就可以逐个删人再询问是否仍然存在 m 个人互为朋友，如果存在则将删去的人彻底删去，否则将其保留。这样将所有人尝试删除一遍后，一定会剩下 m 个人，且他们两两都是朋友。也就是说，如果我们能够求解 A 问题，我们就能在合理的时间内求解 B 问题，就说明 B 问题不难于 A 问题。