Mathematics and Computation

A theory revolutionizing technology and science

Author: Avi Wigderson

1. 介绍(Intro)

"$\mathcal{P}$ *vs* $\mathcal{NP}$ *---- a gift to mathematics from computer science.*" *Steve Smale.*

本书作者重点探究数学与计算的关系，特别地，这里计算是指计算理论(ToC)，并沿着计算理论的子领域：计算复杂性理论不断展示数学与计算的魅力。

1.1 论数学与计算的相互作用(Interaction)

一切的源头："Big bang"源于 Turing 1936 的论文：

《论可计算数，及其在判定问题中的应用》

提出了重要的理论模型：图灵机(Turing Machine)；一方面讲，图灵机作为一个基本的计算的数学模型，使得我们第一次对计算任务、解决问题的算法以及所需的基本资源有了一个清晰的定义；另一方面，图灵机的优雅的定义使得其自身能够被简单地应用或逻辑上的设计到软件、硬件中并推动计算机的革新上。

但事实上，数学与计算并不是在 1936 年才第一次相遇。古代的数学源于对计算的需要，例如对自然现象的预测，管理农作物等等；当时对数学的理解仅仅是通过对数据进行计算而得到的。尽管算法(Algorithm)是 20 世纪才被定义出来的，但人们早已学会了使用算法；例如，公元前 300 年，伟大的欧几里得就提出了 GCD 算法来寻找最大公因数，并写出了著名的几何原本，中国也有著名的《九章算术》闻名于世。

在近现代，数学与计算的联系更加地加强了。文艺复兴时期，数学家们就发现了最基本的计算"原子"，公式(formulas)，那个时候，人们用公式来求解二次以及三次方程（比如求根公式来解决一元二次/三次方程），在那个时代中，人们争相竞争给出方程的求根公式，一直到 Ferrari 第一个给出了一元四次方程的求根公式；但至此之后，人们再也没有寻找到一元五次方程的求根公式。后来，Abel-Ruffini 定理向我们揭示了：五次(quintic)及更高次的代数方程没有一般的代数解法。这也可能是最早的"难结论"(hardness result)，它证明了对于一个具体的问题，在一个精确的计算模型上，不存在算法解决它。再后来。数学界的无冕之王 Hilbert 梦想着一切数学问题的答案，都有一个"机械过程"来得到（判定版本就是都有一个过程来决定真伪），Hilbert 对这种过程的追求启发了后续 Gödel，Church，Post 和 Turing 的工作，但他们的结论粉碎了 Hilbert 的美梦（证明其不可实现，哥德尔不完备定理，Halting 问题、Post 对应问题不可解等等）；不过，这些看似"坏"的消息，却也促进了计算与算法的形式定义，推动了计算机的革新。

随着计算机科学的诞生，以及计算理论稳定地不断提高，加深，多样化，过去的几十年里，数学与计算的联系越来越紧密，我们可以把这些联系分为 4 个方面：

① ToC 需要使用数学技术与结论。

很自然的可以联想到逻辑和离散数学（哥德尔定理等等），同样还有

代数和几何技术来理解近似算法，使用拓扑方法来研究分布式系统，使用数论以及代数几何来构造伪随机等等。

② 数学也需要算法与计算机。

一个简单的例子：四色问题，就是由计算机证明的；我们现在有许多对代数，群论，几何以及统计等许多数学方向的工具包来使用，帮助我们快速计算和证明；同时，现在数学证明验证和证明发现也在不断的靠计算机发展。

③ 数学对象真的存在么？非构造性的存在性证明会对数学有更加深入的理解。例如：Cantor 关于大多数实数不是代数的证明，他的证明令数学界震惊。

④ 对计算的理解也会产生更多的数学结果。

例如 hardness、combinatorial regularity lemmas and so on。

总而言之，数学与计算有着许许多多的强力纽带连接着他们，我们将会在本书的后续继续领悟他们，并且着力关注着计算复杂性理论(computational complexity theory)。

1.2 计算复杂性理论

早期的计算理论着力于探究可计算理论（这也是一个很有趣的领域），即判定一个问题是否可以被一个算法解决。但很快人们就发现了，尽管有很多问题被证明为可计算的，但解决的过程仍然是不可解的，即最好的算法也需要几天甚至是数十年的时光才能计算出结果，这对于人们来说是不可接受的。这也催生出一个研究算法效率或者表现的领域：诞生于 1960s 的计算复杂性理论；它研究什么是“高效”的计算：在一个自然的计算模型下，解决一类自然的任务所需要的最小资源是多少？随着时间的推移，计算复杂性理论已经蓬勃发展，即使到目前，这个领域依旧是活跃且充满活力的。

2. 序章：计算，不可判定性以及数学知识的困境

在这一节我们将快速简洁的浏览一下导致计算复杂性产生的源头。

数学分类问题：我们经常想理解哪些对象是具有某种属性 P 而哪些没有；一些通俗的例子：哪些动物有鳃而哪些没有？哪些植物有绿叶而哪些没有？一些更广泛的数学的分类问题如下所述：

① 哪些丢番图方程有解？
② 哪个绳结是没有打上结的？
③ 哪个平面图是可以 4 着色的？
④ 哪些定理可以在皮亚诺算数系统中证明？
⑤ 哪些椭圆曲线是模的？
⑥ 哪些动态系统是混沌的？

理解与算法 一个核心的问题是，什么是理解？什么时候我们才会对分类问题的被解决而感到满意？一个主要的观察（Hilbert’s）是一个令人满意的解通常会提供一个机械过程(mechanical procedures)，它将来判定我们研究的对象是否具有某种属性 P；实际上 Hilbert 的第 1 和第 4 问题就是想探究上述观察，并且他任务这两个问题的结论都是正确的，也就是说，数学应该以这种计算的感受方式而被理解。

因此，Hilbert 便把对数学知识的理解视为通过计算过程得到答案；然而，他并没有对“计算”作形式化定义。这个任务在二十世纪初被逻辑学家所提出并在 1930s 取得相关的成绩，突破性的进展由 Gödel、Turing、Church 以及其他科学家共同从不同的出发点对计算作形式化定义（比如图灵机与 λ 演算）；目前来看，图灵的定义更加的实用并具有通用性，以至于带动了计算机的革新。正如之前所说，图灵的文章也解决了 Hilbert 的第 1 和第 4 问题，但回答恰恰相反，结论的否定的！

凭借着图灵机的力量，图灵定义了一个算法（判定程序）：一个图灵机，它对所有的输入在有限的时间内都停机(halting)；证明过程中，利用 Cantor 的对角线方法构造矛盾，我们得到这样的机器不存在，即该算法不能被计算，停机问题不可判定。这彻底粉碎了 Hilbert 的美梦！

可判定性与不可判定性 尽管不可判定性总是那么让人有一点点失望，但这些否定的结论并没有阻挡我们前进的脚步，我们会把目光集中在这些对象的子类：限制条件下的可解性；但这一部分现在不做展开。
随着研究的深入，人们发现了，即使有些问题是可判定的，也就是在有限的时间内可解，但有限的时间是多少呢？一年，十年都是有限的，这在数学的应用上是不允许的，因此便产生了高效算法与计算复杂性的领域。

高效算法与计算复杂性 一个主要的量化指标就是时间，当然随着我们后续的深入了解便可以知道，还有空间等等资源。本书的后续章节主要都是围绕这个主题而讨论，在此只阐明一个论点，后续不再赘述。

对研究对象的表示对算法过程是至关重要的，举一个简单的例子，用 1 进制和 10 进制来表示数字，在运算上显然是有差异的；同时，有人会对用离散的有限的事务来表示连续的对象而产生怀疑，这是不用担心的，你会发现，我们所使用的英文教材，每一页的文字都是有限的字母表，离散的组合构成；同样图灵机的输入也是离散有限的，也能够表示连续的事务。