Mathematics and Computation

数学与计算

一个革命化科技与科学的理论

Avi Wigderson

Princeton University Press
Princeton and Oxford

由普林斯顿大学出版社出版，威廉街41号，新泽
西州普林斯顿，08540

英国：普林斯顿大学出版社，牛津街6号，伍斯特郡
，牛津，OX20 1TR

press.princeton.edu

国会图书馆控制号：2018965993 ISBN：978-0-691
-18913-0

英国图书馆出版编目数据可获取 le

编辑：Vickie Kearn、Lauren Bucca 和 Susannah Shoemaker 制作编
辑：Nathan Carr 封面设计：Shar Batsry 和 Avi Wigderson 制作：J
cquie Poirier 宣传：Matthew Taylor 和 Kathryn Stevens 校对：Cy
estmoreland

这本书是用LATEX编写的

出版者想感谢本卷的作者提供了用于印刷本书的校对稿。

印刷在无酸纸上 ∞ 在美利坚合众国印刷
10 9 8 7 6 5 4 3 2 1

Dedicated to the memory of my father, Pinchas Wigderson (1921–1988),
who loved people, loved puzzles, and inspired me.

Ashgabat, Turkmenistan, 1943

Contents

Acknowledgments

在这本书中，我试图展示我在这个领域四十年中所获得的一些知识和理解。这些知识的主要来源是计算理论社区，这个社区是我整个期间的学术和社会家园。这个美好社区的所有成员，尤其是我的老师、学生、博士后和合作者，以及我参加的无数讲座的演讲者，都是这些知识和理解的来源，往往比我所阅读的书籍和期刊还要多。我们是一个慷慨而互动的社区，其成员乐于与他人分享自己的知识和理解，并且受到领域文化的培养，做得很好。这些互动使（仍然使）学习对我来说是一种极大的快乐体验！

更直接地说，本书的内容和呈现得益于许多仔细阅读早期草稿并就所有层面提出宝贵建设性评论的人；这使得本书变得更好。为此，我感谢Scott Aaronson、Dorit Aharonov、Morteza Alimi、Noga Alon、Sanjeev Arora、Boaz Barak、Zeb Brady、Mark Braverman、Bernard Chazelle、Neil Chriss、Tom Church、Geoffroy Couteau、Dennis Dolan、Andy Drucker、Ron Fagin、Yuval Filmus、Michael Forbes、Ankit Garg、Sumegha Garg、Oded Goldreich、Renan Gross、Nadia Heninger、Gil Kalai、Vickie Kearn、Pravesh Kothari、James Lee、Alex Lubotzky、Assaf Naor、Ryan O’Donnell、Toni Pitassi、Tim Randolph、Sasha Razborov、Tim Roughgarden、Mike Saks、Peter Sarnak、Susannah Shoemaker、Amir Shpilka、Alistair Sinclair、Bill Steiger、Arpita Tripathi、Salil Vadhan、Les Valiant、Thomas Vidick、Ben Lee Volk、Cyd Westmoreland、Edna Wigderson、Yuval Wigderson、Ronald de Wolf、Amir Yehudayoff、Rich Zemel和David Zuckerman。

非常特别的感谢Sahar Batsry，感谢她在封面设计上对许多不断发展的想法进行了细致入微的工作，感谢Edna和Yuval Wigderson在LATEX、英语、图形以及许多其他超出我能力的技术问题上的大力帮助，最后还要感谢Eyal和Einat Wigderson带来的美食（以及许多其他）乐趣。

我感谢普林斯顿大学出版社出版这本书，特别是内森·卡尔、维基·基恩、萨苏珊娜·肖梅克，尤其是西德·韦斯特莫兰德，她对此书的校对工作简直令人惊叹。

在线资源简化了许多书籍撰写的方面。我想特别感谢谷歌学术、维基百科和arXiv存储库对我极大的实用性！

一些本书的章节是对我从我的调查 [Wig06] 中取材的修订和扩展，而该调查又反过来使用了本卷中与Goldreich共同进行的调查的部分 [GBGL10]。

最后但同样重要的是，我要感谢汤姆和罗塞琳·内尔森允许我使用他们在爱达荷州太阳谷的美丽家园——这本书的大部分内容都是在过去几个夏天在那个宁静的环境中写成的。

Mathematics and Computation

1 Introduction

"$\mathcal{P}$ versus $\mathcal{NP}$ — a gift to mathematics from computer science"

–Steve Smale

这里只是我们将在这本书中探讨的冰山一角：找到1,000位整数质因数需要多少时间？事实是：（1）我们甚至无法粗略估计答案：它可能不到一秒，也可能超过一百万年，并且（2）今天存在的几乎所有电子商务和互联网安全系统都基于这样一个信念：这需要超过一百万年！

消化这个单个例子已经开始阐明*computational com- plexity theory*的概念革命，本书致力于此。它说明了数学家们研究数千年的纯数论问题，如何成为一项价值万亿美元的产业的基石，几乎所有的人、公司和国家都对其至关重要。提取这种新颖的意义和效用依赖于使上述问题精确化，并将(2)的非正式陈述转化为数学定理。这反过来又需要正式定义诸如*algorithm, efficiency, secret*和*randomness*等概念，以及其他几个*proof*的新概念。解决（现在至关重要的）挑战（1），即证明分解的难度或提出替代方案，与$\mathcal{P}$与$\mathcal{NP}$的巨大谜团密切相关。在这个情节的最后一转中，我们的计算路径出现了一个分叉，通过这个分叉，(1)的答案可能根本取决于我们是否允许经典或量子物理为我们的计算机提供动力。这种新的可能性推动了学术界和产业界的大量投资，以尝试在物理上实现量子计算机的潜力。这也要求重新审视和重新定义上述概念，以及许多与量子力学相互作用并提出测试其基础的全新方式的物理概念，如*entanglement*和*decoherence*。

这本书您正在阅读的将探讨这个金矿的数学和智力方面。它将解释*computational complexity theory*，这个理论所创造和革命的观念，以及它与数学的众多联系和相互作用。在其半个世纪的存在中，计算复杂性理论已经发展成为一个丰富、深刻和广泛的数学理论，取得了惊人的成就和艰巨的挑战。它与大多数其他数学领域建立了强大的联系，同时也在对影响我们社会各个方面的*technological revolution*产生重大实际影响（其中互联网安全和量子计算只是"仅仅"的例子）。

计算复杂性理论是*theory of computation* (ToC)的核心子领域，并在其发展中扮演着关键角色。这一理论与物理学、生物学、数学和经济学的大师们并列，是计算驱动的新领域的核心。我在这本书的最后一章（可以首先阅读）中，对ToC进行了全景式的概述。该章节描述了计算理论所引发的智力新星及其持续塑造的过程。它揭示了ToC对所有科学、技术和社会的广泛影响，并讨论了其方法论、挑战以及在知识领域的独特地位。

以下我回顾了计算与数学之间长期互动的历史。我接着简要概述了计算复杂性的演变和本质。然后描述了本书的结构、范围和目标读者，接着按章节描述其内容。

1.1 On the interactions of math and computation

计算理论是研究计算机科学和技术形式基础的研究。这个动态且快速发展的领域横跨数学和计算机科学。它从这些母学科非常不同的特点、动机和传统中获得了巨大的益处。双方自然地从中诞生，源于图灵1936年开创性的论文"关于可计算数，及其在Entscheidungsproblem中的应用"的"大爆炸"。这是一篇由数学逻辑领域的研究生撰写的论文，其长标题可能使其显得晦涩难懂。然而，凭借图灵非凡的洞察力、阐述和动机，它成为数学建模中具有独特影响的鼓舞人心的典范。这篇论文正式定义了我们现在所说的"图灵机"形式的*algorithm*。一方面，图灵机是一种形式

计算模型的数学，首次实现了计算任务的严格定义、解决这些任务的算法以及这些任务所需的基本资源（特别是，允许图灵证明非常基本的任务是不可计算的）。另一方面，图灵机的极其优雅的定义使得其简单、逻辑的设计可以轻松地在硬件和软件中实现，从而引发了计算机革命。

这些理论和实践方面构成了ToC的双重性质，并强烈影响了该领域及其发展。在数学方面，计算的抽象概念揭示了其作为一个极其深刻和神秘的概念，以新的视角照亮了其他，通常是研究得很好的概念。在追求计算的抽象研究时，ToC的发展与其他数学领域一样。其研究人员证明定理，遵循数学文化来推广、简化和创造变体，根据基于美学和美的直觉行事。在实践方面，自动化计算的普遍适用性推动了计算机技术的快速发展，现在它主导着我们的生活。理论与实践的互动从未停止。计算机科学和工业不断发展的世界持续创造新的计算类型和属性，这需要理论建模和理解，并直接影响ToC的数学演变。相反，那里产生的思想、模型和技术反馈到实践世界。除了技术之外，对ToC的外部影响越来越大的来源是自然和科学。许多自然过程可以（并且应该）被视为信息过程，并需要类似的计算理解。在这里，理论建模、技术和新的理论问题反馈以建议实验和更好地理解科学数据。第20章中讨论了这些联系中的更多内容。

不言而喻，数学和计算并非在1936年首次相遇；它们自人类文明之初就相互联系。事实上，古代数学主要源于计算的需求，无论是预测各种自然现象、管理农作物和牲畜、制造和建筑、贸易商品还是规划未来。因此，设计数字的表示形式并开发对它们进行算术运算的高效方法变得至关重要。更普遍地说，以非常基本的方式，只有将计算过程应用于手头的数据，数学理解才能解决任何实际问题。因此，尽管算法在20世纪才得到正式定义，但数学家和科学家们持续地设计、描述和使用越来越好的算法（尽管这些算法通常是非正式解释且很少分析）来从他们的理论中得出结论。例子不胜枚举，我们仅列举一些亮点。欧几里得大约在公元前300年设计了快速的最大公约数算法[1]，以避免在化简分数时费力地分解整数。欧几里得的著名13卷 *Elements*，许多世纪以来的核心数学文本，包含了数十种计算数值和几何数量及结构的算法。在同一时期，中国数学家编纂了 *The Nine Chapters on the Mathematical Art*，其中包含了许多计算方法，包括“高斯消元法”（用于解线性方程组）。在9世纪，阿尔-花拉子米$^{-}$（算法一词即来源于此）撰写了他的书籍 *Compendious Book on Calculation by Completion and Balancing*和*On the Hindu Art of Reckoning*。这些书籍分别阐述了当时关于代数和算术问题的算法，如解二次方程和线性系统，以及进行十进制系统的算术运算。十进制系统之所以能够作为表示数字的主导方式幸存下来，正是因为这些高效算法在执行任意大数的算术运算方面的有用性。

现代时代加剧了数学与计算之间的联系。再次，举例。在文艺复兴时期，数学家发现了 *formulas*，最基本的计算配方，通过根式解决三次和四次方程。[2] 事实上，在 1500 年代初，塔塔利亚、皮奥雷、费拉里等人的著名竞赛都是关于谁有更快算法来解决三次方程。阿贝尔-鲁菲尼定理表明五次方程没有这样的公式，可能是最早的 *hardness result*：它证明了在精确的计算模型中，一个具体问题的算法不存在。牛顿的 *Principia Mathematica* 不仅是一部伟大的科学和数学理论的杰作；它也是一部计算这些理论预测的算法杰作。也许最著名的是

[1] The GCD (greatest common divisor) problem is to compute the largest integer that evenly divides two other integers.
[2] Namely, using arithmetic operations and taking roots.

最一般的是“牛顿法”，用于逼近任意度实多项式的根（实际上绕过了上述提到的阿贝尔-鲁菲尼障碍）。关于高斯的大作*Disquisitiones Arithmeticae*也是如此——它充满了算法和计算方法。一个著名的例子（在他去世后发表），是他发现的快速傅里叶变换（FFT），信号处理的核心算法，大约在150年前，J. W. 库利和J. W. 图基“官方”发现它之前。莱布尼茨、巴贝奇、洛夫莱斯等人超越了具体问题，率先明确尝试设计、构建和编程通用计算设备。最后，希尔伯特梦想将所有数学建立在计算基础上，寻求一种“机械程序”，原则上可以确定所有数学真理。他相信真理和证明是一致的（即每个真命题都有一个有效的证明），并且可以通过这样的计算程序自动找到这样的证明。将希尔伯特计划形式化为数学逻辑的探索直接导致了哥德尔、丘奇、波斯特和图灵的作品。他们的工作打破了希尔伯特的梦想（证明它们是无法实现的），但在这样做的同时，诞生了计算和算法的正式定义。一旦这些理论基础被奠定，计算机革命就开始了。

计算机科学的诞生，以及随之而来的目录（ToC）的稳步增强、深化和多样化，加深了数学与计算之间的互动，在过去几十年中这些互动呈爆炸式增长。这些互动可以大致分为四个（自然重叠的）类别。前两个类别源于一个领域使用另一个领域开发的专长；这些互动通常是单向的。接下来的两个类别更为复杂，并且高度互动。我们将在整本书中看到许多这样的互动。

- 一种交互源于ToC需要使用通用的数学技术和结果。最初，这些来源仅限于与计算机科学有自然亲和力的领域，如逻辑和离散数学。然而，随着ToC的发展，它变得更加深入和广泛，需要从不同的数学领域引进技术和结果，其中一些相当出乎意料。这些例子包括使用分析技术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，以及在构造伪随机对象中使用数论和代数几何。

- 相反类型的交互是数学使用算法（和计算机）的需求。如前所述，数学家们需要算法，并且已经发展了数百年。但在图灵之后，ToC将算法设计转化为一个全面的理论，准备被使用和应用。这个理论有维护和操作所有类型信息的通用技术，以及比较这些算法质量和分析这些算法资源的方法。同时，计算机变得可用。这种融合导致了在几乎所有领域为数学家开发特定算法和软件的巨大繁荣，包括代数、拓扑、群论、几何、统计学和其他领域的计算工具库。在另一个方面，计算机程序在数学证明验证以及证明发现方面的开发和使用正在不断增长。

- 一个更深、更根本的互动来源是保证某些数学对象存在的大量数学定理。现在，对这种好奇的反应已经变成了一种反射：*Can the object guaranteed to exist be efficiently found?* 在许多情况下，包括上面提到的，有很好的实际理由去寻求这样的程序。在更哲学的层面上，非构造性存在证明（如希尔伯特关于有限基定理的第一个证明和康托尔关于大多数实数不是代数数的证明）使数学界成员感到震惊。即使在有限设置中，存在证明也要求有更好的理解，尽管存在用于所需对象的暴力（但效率极高）搜索算法。我们在数学的各个领域都有越来越多的证据表明，即使没有任何直接、实际的需求或哲学上的愿望去寻找这样的高效算法，仍然会不可避免地导致对当前数学领域的更深入理解。这种追求提出了新的问题，揭示了新的结构，有时甚至使“被充分理解”的主题得到复兴。

[3]For the purpose of efficiently predicting the orbits of certain asteroids.

- 最终交互的来源产生于，也许令人惊讶的是，计算的研究导致了新的数学成果、理论和问题的产生，这些成果不是算法性的，而是结构性的。这些自然地源于分析算法和证明难度结果的需要。因此，诞生了全新的概率集中结果、几何关联定理、组合规律引理、等周不等式、代数恒等式、统计检验等等。这些成果激发了众多数学领域的合作，其中一些领域已经相当成熟，而另一些领域则刚刚开始萌芽。

数学和计算通过众多强大的纽带相连，其中一些超出了本书的范围。在这里，我们将见证计算复杂性领域这些相互作用的兴趣。

1.2 Computational complexity theory

研究ToC的早期几十年集中在理解哪些计算问题可以通过算法解决，哪些不能。但很快变得明显，这条分界线太粗了；许多在原则上可以解决的问题仍然难以处理，因为解决这些问题的最佳算法将在任何人关心或看到答案之前很久就终止了。这产生了对一个更加精细的理论的需求，该理论将解释不同算法的*performance*。因此，20世纪60年代诞生了计算复杂性理论，其最初的任务是以最一般的意义理解*efficient computation*：
determining the minimal amounts of natural resources (like time, memory, and communication) needed to solve natural computational tasks by natural computational models。为了应对这一挑战，该理论发展了一套强大的算法技术及其分析方法，以及一个使用*complexity classes*对计算问题进行分类的系统。这些结果还导致了关于该领域有效计算能力和限制的自然长期目标的制定。但这些发展只是故事的一半，因为算法消耗的资源只是计算的基本属性，而在不同环境中算法的效用提出了其他需要探索的属性。

随着时间的推移，以及如此多样的内部和外部动机，计算复杂性理论极大地扩展了其目标。它承担了计算建模和对各种中心概念的理解，其中一些被伟大的思想家研究了数百年，包括*secret, proof, learning, knowledge, randomness, interaction, evolution, game, strategy, coordination,*和*synchrony*等概念。这种计算视角常常为这些旧概念带来了全新的意义。[4]此外，在某些情况下，由此产生的理论在（实际上也促进了）重大的技术进步。[5]在其他情况下，这些理论构成了与其他科学互动的基础。

因此，从理解什么是可以高效 *computed* 的目标开始，涌现出许多具有深刻概念意义的长远自然目标。什么可以高效地学习？什么可以高效地证明？验证一个证明是否比找到一个证明更容易？机器知道什么？算法中随机性的力量是什么？我们能否有效地利用自然随机源？量子力学计算机的力量是什么？我们能否在算法中利用量子现象？在具有不同（可能冲突）激励的计算实体设置中，可以共同实现什么？私下里可以做到什么？计算机能否高效地模拟自然或大脑？

研究高效计算已经创造了一种强大的方法论，可以用来研究这些问题。以下是其中一些重要的原则，我们将在实践中反复看到，随着我们在本书中的进展，其含义将变得更加清晰。*Computational modeling*：揭示过程的潜在基本操作、信息流和资源。*Efficiency*：尝试最小化使用的资源及其权衡。
Asymptotic thinking：研究越来越大的对象上的问题，因为结构

[4]For example, this approach suggests that in proofs, one can decouple verification and understanding: *every* mathematical theorem can be proved in a convincing manner that nonetheless reveals absolutely no information except its validity (This will be discussed in Section 10.2.)

[5]The best example is cryptography, which in the 1980s was purely motivated by a collection of fun intellectual challenges, like playing poker over the telephone, but developed into a theory that enabled the explosive growth of the Internet and e-commerce. (This will be discussed in Chapter 18.)

经常在极限中显现自身。*Adversarial thinking*：总是为最坏的情况做准备，用一般的、计算性的限制来代替具体的和结构性的限制——这样的更严格的要求往往使事情更容易理解！*Classification*：根据它们所需的资源将问题组织成（复杂度）类别。*Reductions*：忽略无知，即使你不能有效地解决问题，也假设你可以，并探索哪些其他问题它能有效地解决。*Completeness*：识别复杂度类别中最困难的问题。[6] *Barriers*：当长时间陷入一个主要问题时，抽象出迄今为止用于解决它的所有已知技术，并试图正式论证它们不足以解决它。

这些原则彼此之间非常有效，在令人惊讶的多样化环境中尤其如此，尤其是在适当的抽象层次上应用时（我认为这确实是一个聪明的选择，并且被反复采用）。这种方法允许ToC揭示不同领域之间的隐藏联系，并创建一个美丽的结构大厦，一个在大量概念、问题、模型、资源和动机中的非凡秩序。虽然许多这些原则在数学和科学中都有应用，但我相信，这种在计算复杂性文化中根深蒂固的纪律性和系统性使用——特别是通过（适当的）归约和完备性进行问题分类——有很大的潜力进一步增强其他学术领域。

计算复杂性领域极其活跃且动态。虽然我试图描述除了基础理论之外，大多数领域的一些最新进展，但我预计最前沿将继续迅速扩展。因此，一些未解决的问题将成为定理，新的研究方向将被创建，新的未解决问题也将出现。事实上，在我写这本书的几年里，这种情况已经反复发生。

1.3 The nature, purpose, and style of this book

计算复杂性理论几乎是我40年的智力（和社会）家园。在这段时间里，我撰写了综述文章，并在该领域的各个方面发表了更多综述讲座。这本书是从这些阐述以及我计划撰写和讨论的许多其他方面发展而来的，但我从未着手去做。事实上，广度是本书的一个重要目标。此外，正如我的讲座和综述一样，在这本书中，我试图解释我们不仅做了什么，还解释了我们为什么这样做，为什么它如此重要，以及为什么它如此有趣!

本书探讨了计算复杂性理论的基础和一些主要研究方向，以及它们与其他数学分支的众多相互作用。当我们讨论以下每一章的内容时，这些计算环境的多样性得以展现。对于每个研究领域，本书专注于其试图尝试的*model*的主要方面。本书依次介绍了每个领域的概念、目标、结果和开放性问题，所有这些都从概念角度出发，提供了充足的动力和直觉。它描述了导致不同概念和结果的思想的历史和演变，并解释了它们的意义和效用。它还突出了计算复杂性理论不同子领域之间（通常是令人惊讶和意外的）丰富的联系；这一领域的统一性是该领域成功的重要部分。

为了突出概念视角，材料通常以高层次和某种非正式的方式呈现。几乎不给出证明，我专注于非正式层面上的通用证明技术和关键思想的讨论。精确的定义、定理陈述，以及当然，许多讨论主题的详细证明，可以在关于计算复杂性的优秀教科书中找到[Pap03, Gol08, AB09, MM11]。此外，出于历史原因和更详细的说明，我在每一章中提供了许多对原始论文、更专业的教科书和综述文章的参考文献。

1.4 Who is this book for?

我将这本书视为以几种不同的方式对多个受众都有用。

- 首先，这是一份邀请，面向数学、计算机科学和相关领域的本科生和研究生，让他们了解这个领域的内容，对其产生兴趣，并加入其中

[6] Namely, those which all other problems in the class reduce to in the sense mentioned above.

研究人员

- 其次，它应该服务于在CS理论某个领域工作的研究生和年轻研究人员，以拓宽他们的视野并加深他们对该领域其他部分及其相互关联性的理解。

- 第三，计算机科学家、数学家、其他领域的学者以及有动力的非学术界人士可以了解计算复杂性的高级概述，其广泛范围、成就和抱负。

- 最后但同样重要的是，该领域的教育工作者可以使用本书的不同部分来规划和补充各种本科和研究生课程。我希望我在这里努力呈现的该领域的概念视图、其方法论、其统一性以及其成就和挑战的美丽与激动人心的特点，将有助于启发和激发这些课程。

不同章节可能需要略有不同的背景知识，但每个章节的介绍都旨在欢迎和亲切。前两章和最后一章应该对上述大多数受众都是可访问的。

让我以一条可能对一些读者有用的建议来结束本节。快速阅读这本书可能很有吸引力。书中许多部分几乎不需要任何特定的先验知识，主要依赖于吸收所引入的定义和概念所需的数学成熟度。希望故事叙述使阅读更加容易。然而，这本书（就像这个领域本身一样）在概念上很密集，我相信在某些部分，概念和思想的密集度需要放慢速度、重读，并可能查阅相关参考以澄清和巩固材料及其在你心中的意义。

1.5 Organization of the book

以下是本书各章节内容的总结。自然，以下提到的某些概念只会在各章节中解释。在介绍性的第2章和第3章之后，其余章节几乎可以按任何顺序阅读。跨越多个章节的核心概念（除了计算本身）包括*randomness* (第7-10章)、*proof* (第3、6和10章)以及*hardness* (第5、6和12章)。

不同章节的焦点可以围绕不同的章节集合进行划分。第2章至第12章主要关注一种计算资源，即*time*，也就是单个机器（各种类型）解决问题所采取的步骤数量。第14章至第19章（以及第10章）涉及其他资源和更复杂的计算环境，其中多个计算设备之间发生交互。最后，虽然数学*modeling*几乎是每一章的重要部分，但在第15章至第19章我们将遇到的复杂计算环境中，它更为重要，在这些章节中，我们将更详细地讨论建模选项、选择和理由。第13章和第20章是独立的综述，前者关于数学与计算复杂性的具体交互，后者关于计算理论。以下是每章的简要描述（以下标题可能与章节标题不同）。

Prelude: computation and mathematical understanding
Chapter 2 是计算复杂度到来的序曲，从将 *algorithm* 的概念形式化为图灵机开始。我们讨论数学中的基本计算问题及其算法。然后，我们探讨关于数学结构类中可决问题和不可决问题之间的相关性，希望完全理解它们。

Computational complexity 101 and the $\mathcal{P}$ vs. $\mathcal{NP}$ question
Chapter 3 介绍了计算复杂性的基本概念：决策问题、时间复杂度、多项式时间与指数时间、高效算法以及类{v*}。我们定义了高效验证和类{v*}，第一个计算意义上的证明概念。我们继续讨论问题之间的高效归约和完备性的概念。然后我们引入{v*}-完备问题和{v*}与{v*}的问题。最后，我们讨论相关问题和复杂度类。对于所有这些概念，我激发了一些

做出的选择 解释它们在计算机科学、数学中的重要性 d 超出。

Different types of computational problems and complexity classes

Chapter 4 引入了新的 *questions* 类型，可以询问关于输入的分类之外的类型，包括计数、近似和搜索。我们还从算法的最坏情况性能转向平均成功或失败，并讨论了密码学中基于单向和陷阱门函数的相关概念。我解释了前一章中开发的方法如何导致其他复杂性类、归约和完备性。这开始描绘了组织在 $\mathcal{NP}$ 之上、之下和“周围”的更丰富的结构。

Hardness and the difficulties it presents

Chapter 5 讨论 *lower bounds*——证明 $\mathcal{P}$ 与 $\mathcal{NP}$ 不同的主要挑战，以及更一般地，证明某些自然计算问题是困难的。本节的核心是布尔电路模型，它是图灵机的“硬件”类似物。我们回顾了用于电路和图灵机限制形式的下界的主要技术。我们还讨论了内省障碍结果，解释了为什么这些技术似乎不足以达到真正的目标：通用模型的下界。

How deep is your proof?

Chapter 6 引入 *proof complexity*，这是 *proof* 基本概念的另一种观点。证明复杂性将计算复杂性方法应用于量化证明自然定理的难度。我们描述了各种命题证明系统——几何的、代数的和逻辑的——它们都捕捉了为证明自然重言式进行推理的不同方法和直觉。我解释了证明系统、算法、电路复杂性和 *space: the final frontier* 问题之间的联系。我们回顾了在此设置中证明下界的主要结果和挑战。

The power and weakness of randomness for algorithms

Chapter 7 使用随机性增强算法功效的评论。我们定义了概率算法及其高效解决的问题类别 $\mathcal{BPP}$。我们讨论了这类问题（以及其他许多问题），对于这些问题，尚未知道快速确定性算法，这表明随机性具有强大的能力。然而，这种直觉可能是一种错觉。接下来，我们介绍了 *computational pseudo-randomness, pseudo-random generators* 的基本概念、*hardness vs. randomness paradigm* 和 *de-randomization*。这些想法表明，随机性是脆弱的，至少在假设类似于 $\mathcal{P} \neq \mathcal{NP}$ 的困难陈述的情况下。本章以对这些想法的演变和来源、它们的惊人后果以及它们对随机性功效之外的影响的讨论结束。

Is π random?

Chapter 8 涵盖“看似随机”的确定性结构。我们讨论“抽象”的伪随机性，这是一个扩展计算伪随机性并容纳数学和计算机科学中各种自然问题的通用框架。我们定义伪随机属性并讨论在确定性中找到伪随机对象的问题。我们将看到如何将$\mathcal{P}$与$\mathcal{NP}$问题、黎曼猜想以及许多自然落入此框架的其他问题视为关于伪随机性的问题。最后，我们讨论结构对伪随机性的二分法，这是一个在各种领域证明定理的范例，并考察这一想法的范围。

Utilizing the unpredictability of the weather, stock prices, quantum effects, etc.

Chapter 9 讨论弱随机源，这是一种自然现象的数学模型，这些现象似乎有些不可预测，但可能远非完美的随机比特流。我们提出了关于概率算法如何以及是否可以使用这种弱随机性的问题，并定义了用于回答此问题的主要对象——随机性提取器。然后我们探讨了导致高效性的思想演变。

科学构造的提取器，以及这个伪随机对象在其他目的上的显著效用。

Interactive proofs: teaching students with coin tosses

Chapter 10 地址再次证明，这次是关于将随机性和交互引入证明定义的影响。这些新的证明概念产生了新的复杂度类，如$\mathcal{IP}$和$\mathcal{PCP}$，以及它们在标准复杂度类中的惊人特征。我们探讨了这种设置如何允许新的证明类型，如零知识证明和抽查证明，以及它们对密码学和近似难度的影响。

Schrödinger's laptop: algorithms meet quantum mechanics

Chapter 11 介绍了量子计算：赋予算法使用量子力学效应进行计算的能力。我们讨论了该理论模型的重要算法，它们如何激励了构建量子计算机的大规模努力，以及这项努力的状态。我们通过使用这个概念再次扩展了证明的想法，并讨论了量子证明的完备问题。这直接与量子哈密顿动力学相联系，这是凝聚态物理的一个核心领域，我们探讨了这些领域之间的一些相互作用。最后，我们讨论了交互式证明回答基本问题的能力：量子力学是否可证伪?

Arithmetic complexity: plus and times revisited

Chapter 12 离开布尔域并引入算术电路模型，这些模型使用算术运算在（大）域上计算多项式。我们回顾了该领域的主要结果和开放问题，将其与布尔电路的研究联系起来。我们阐述了Valiant的算术复杂性理论，其主要复杂性类$\mathcal{VP}$和$\mathcal{VNP}$，完备问题，行列式和永续多项式。我们讨论了一种通过代数几何和表示理论解决$\mathcal{VP}$与$\mathcal{VNP}$问题的最近方法。我们还概述了一组已知具有强下界的限制模型。

所有到目前为止的章节都集中在一种主要的计算资源上：*time*（，或者更普遍地说，是基本操作的数量）。在扩大我们的范围之前，我们先休息一下，插入一个插曲。

Interlude: vignettes of interaction between math and computation

Chapter 13 它与其它不同。它是书的中间部分，我们进行一次（技术）休息。在这个间歇中，我们展示了一系列关于复杂性与数学家在不同数学领域互动的简要调查。这些小故事涵盖了广泛的主题，以及不同类型的互动和动机。它们揭示了计算复杂性的广泛兴趣，并展示了“计算视角”下的数学。

在这段休息之后，我们转向研究更广泛的计算问题、资源、模型和属性。随着计算参与者数量的增加，完整输入（甚至被解决的问题）可能未知，以及新的效率衡量标准和成功标准的出现，对复杂计算情况建模变得更加重要、有趣且困难。

Space complexity: memory bottlenecks

到这一点为止，我们研究的主要复杂度度量是 *time*，或者更普遍地说，算法执行的基本操作数量。

Chapter 14 讨论了 *space* 复杂度：算法的内存需求。在介绍了一些该领域的基本结果和开放性问题之后，我们关注了两个具体问题，在这些问题中，可以用非常少的内存完成惊人的成就。我们首先讨论了流模型和草图技术。然后我们展示了如何仅使用常量内存来计数到任意高的数字。

Communication complexity: information bottlenecks and noise

Chapter 15 引入通信复杂度——一个极其简单、完全信息论的两方通信模型。然而，对其研究却揭示了惊人的深度和广度，这个简单模型中的基本结果最终被证明对理解一系列令人惊讶的计算模型具有重要意义。我们还讨论了这种 *interactive* 通信设置如何暗示了信息论和编码理论领域中经典问题的扩展。我们回顾了理解这些扩展的一些主要结果和挑战。

On-line computation: Is clairvoyance overrated?

Chapter 16 讨论 *on-line* 算法——需要响应连续请求或信号的响应式系统，试图优化依赖于 *future* 的目标。我们首先探讨如何通过 *competitive analysis* 的概念来建模和衡量此类算法的质量。然后，我们再次看到这种受限算法在各种情况下所具有的惊人力量，从操作系统到股市。

Learning: How do programs recognize spam, know your tastes, and beat you at chess?

Chapter 17 讨论计算学习理论以及理解和设计能够理解、应对并在不熟悉和未指定的环境中茁壮成长的系统的复杂任务。我们考察了*supervised learning*教师-学生框架中的基本建模问题和提案，旨在捕捉将给定示例泛化以形成概念的能力。我们回顾了在此环境中建模学习的两种不同方法，即“逻辑”方法“极限识别”和“统计”方法“无分布学习”。我们研究了这些环境中的某些学习算法，它们的关键方法和对其分析的见解，以及两种方法的严重局限性。

Crypto: secrets and lies, knowledge and trust

Chapter 18 涉及密码学，这是可能的最复杂的计算环境之一。互联网购物或电子投票协议的精髓是什么？在这里，许多代理及其算法相互作用以实现某些共同目标，必然使用代理的个人数据。然而，这些代理希望将信息保密，防止非法窃听者，并确保实现其目标对破坏者具有弹性。我们探讨了丰富的密码学任务和约束，以及数学建模的原则。我们将看到，当计算能力有限时，许多证明为不可能的任务（如通过电话玩扑克牌）悖论性地 *can* 可以执行。我们讨论了实现这些任务的全面理论的演变。

Distributed environments: asynchrony and symmetry breaking

Chapter 19, 在分布式计算方面，也处理交互的各方。然而，我们的重点将放在一个非常不同的障碍上：*asynchrony*。在这种情况下，参与者没有共同的时钟，必须在任意通信延迟的情况下进行计算。我们讨论了建模此类约束及其施加的限制。在这里，也发展出了一种美丽的理论，精确地描绘了哪些任务可行以及哪些不可行的边界。这部分理论建立在这些问题与拓扑学之间深刻的联系之上。

Epilogue: the nature of ToC

Chapter 20 本书以对目录和其巨大知识影响的全景式回顾作为结尾。它旨在调查该领域的许多方面，包括科学和社会方面。我们将调查目录与自然科学和社会科学之间爆炸性的联系，将计算视角融入自然和社会的理论和模型中。目录被揭示为一个独立的学术学科，对大多数其他学科至关重要。我们讨论了该领域的一些长期知识挑战和教育挑战，其社会特征以及随着其范围和使命的扩大，它将面临的日益增长的角色适应。

1.6 Notation and conventions

本书中的大多数符号都是在我们需要时引入的，这里只包括重要的渐近符号。我们还列出了一些约定和建议。

- **Undefined notions** 当提到一个对你来说不熟悉的观念但并未立即定义时，理解其含义对于理解文本并非至关重要（但你可以自由地查阅，例如在维基百科上）。

- **References** 当给出参考文献列表时，它通常是按时间顺序排列的。这并不总是从出版年份中明显看出，主要是因为我总是试图提供最全面和最新的版本。在某些情况下，这些是期刊论文（需要最长时间才能发表），有些是会议出版物（发表更快），在极少数情况下，只有电子版存在（几乎是瞬间的）。此外，当我说一个结果是“最近”的，请注意，这是相对于书籍的出版日期而言的。

- **Footnotes** 这本书中有 *many* 个脚注。在几乎所有情况下，它们都是为了丰富、详细说明或进一步解释某事而设计的。然而，文本本身应该独立于它们。因此，您可以安全地跳过脚注；这很少会影响理解。

- **iff** 简写 *iff* 将会贯穿全文，表示“当且仅当”。

- **Asymptotic notation** 至关重要的以下渐近符号与整数函数的“极限增长”相关。设 f, g 为两个整数函数。然后我们表示：

  $*$ $f = O(g)$, 如果对于某个正常数 C，对于所有足够大的 n，$f(n) \leq C \cdot g(n)$。$*$ $f = \Omega(g)$，如果对于某个正常数 c，对于所有足够大的 n，$f(n) \geq c \cdot g(n)$。$*$ $f = \Theta(g)$，如果同时满足 $f = O(g)$ 和 $f = \Omega(g)$。$*$ $f = o(g)$，如果 $f(n)/g(n)$ 随着 n 趋于无穷大而趋于零。

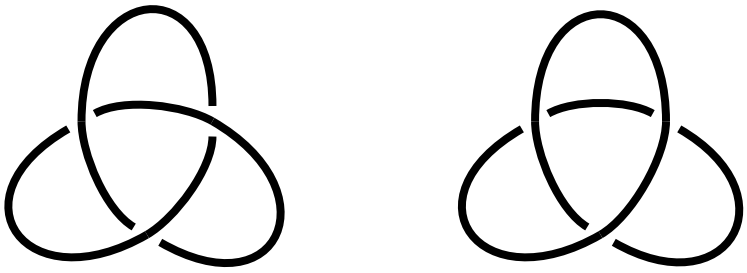
  我们称一个整数函数 f（至多）以 *polynomially* 的速度增长，如果存在常数 A, c，使得对于所有 n，有 $f(n) \leq An^c$。我们称 f（至多）以 *exponentially* 的速度增长，如果存在常数 A, c，使得对于所有 n，有 $f(n) \leq A \exp(n^c)$。

2 *Prelude*: Computation, undecidability, and limits to mathematical knowledge

本简短部分将简要回顾导致计算复杂性理论诞生的思想史。

Mathematical classification problems 哪些数学结构我们可以希望理解？考虑任何特定的数学对象类别和任何特定的相关属性。我们寻求确定哪些对象具有该属性，哪些不具有。这种非常一般的*classification problem*的例子包括以下[1]：

1. 哪些丢番图方程有解？2. 哪些结是可解的？（见图1）3. 哪些平面图是四色可分的？（见图2）4. 哪些定理可以在佩亚诺算术中证明？5. 哪些光滑流形对是同胚的？6. 哪些关于实数的初等陈述是真实的？7. 哪些椭圆曲线是模的？8. 哪些动力系统是混沌的？



NO YES

图1. 问题2的实例及其分类。左边是三叶结的图，右边是无结之一。

Understanding and algorithms 一个中心问题是我们要如何理解 *understanding*。我们何时会满意地认为我们的分类问题已经得到了合理的解决？是否存在我们永远无法解决的问题？一个中心观察（主要是由大卫·希尔伯特普及的）是，“令人满意的”解通常（明确或隐含地）提供 *mechanical procedures*，当将其应用于一个对象时，它将（在有限时间内）确定它是否具有该属性。上述希尔伯特问题1和4似乎是在这样的预期下提出的，即答案将是肯定的；也就是说，数学家将能够以这种计算方式理解它们。

[1]It is not essential that you understand every mathematical notion mentioned below. If curious, reading a Wikipedia-level page should be more than enough for our purposes here.

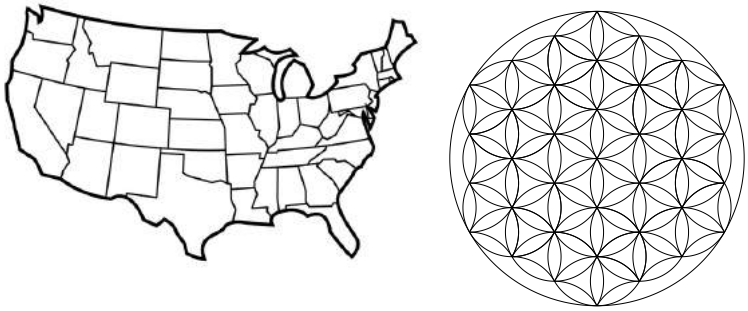


图2. 问题3的实例及其分类。两个地图都是四色可分的。

因此，希尔伯特将数学知识等同于对答案的计算访问，但从未正式定义计算。这项任务在20世纪初由逻辑学家承担，并在20世纪30年代取得了巨大的成功。哥德尔、图灵、丘奇等人的突破性发展导致了几个相当不同的计算形式定义，这些定义最终被证明在能力上是相同的。其中，图灵1936年的论文[Tur36]最具影响力.*Indeed, it is easily the most influential math paper in history*。我们在第1.1节中已经提到，在这篇极具可读性的论文中，图灵孕育了计算机科学这一学科，并引发了计算机革命，这场革命彻底改变了社会。图灵的计算模型（很快被命名为*Turing machine*）成为有史以来最伟大的智力发明之一。它优雅简洁的设计一方面，以及它的通用能力（由图灵阐明和举例说明）另一方面，立即导致了实施，并且从那时起快速的发展永远改变了地球上的生活。这篇论文是展示如何卓越的理论先于并推动显著的技术和科学进步的最有力的证明之一。但除此之外，图灵的论文还解决了上述问题1和4！在否定方面！！让我们看看。

使用图灵机，我们最终得到了计算的严格定义，为希尔伯特的问题提供了形式意义。它允许证明关于“机械过程”能做什么和不能做什么的数学定理！*Turing defined an* ***algorithm*** *(also called a* 决策过程 *) to be a Turing machine (in modern language, simply a computer program) that halts on every input in finite time*。因此，算法计算函数，作为有限对象，人们立即从康托尔式的对角线论证中看到，某些（实际上几乎是所有）函数是算法可计算的。这些函数也被称为 *undecidable*；它们没有决策过程。但图灵更进一步，表明某些具体、自然的功能，如上述希尔伯特的Entscheidungsproblem（4），是不可判定的（这也由丘奇独立证明）。图灵的简洁1页证明在图灵机上采用了一个哥德尔式的自我引用论证.[2] 这些有力地证明了图灵基本计算模型的数学价值。

Decidability and undecidability 图灵因此打破了希尔伯特的第一个梦想。问题4不可解意味着我们永远无法在希尔伯特的意义上理解哪些定理是可证明的（例如，在佩亚诺算术中）：没有算法可以区分可证明和不可证明的定理。又过了35年才同样解决了希尔伯特问题1。它的不可解性（由戴维斯、普特南、罗宾逊和马蒂亚谢维奇在1970年证明）表明，我们将永远无法以这种方式理解整数上的多项式方程：没有算法可以区分可解和不可解的方程。

一个关键因素在于这些（以及所有其他不可判定）结果中，表明这些数学结构（佩亚诺证明、整数多项式）可以 *“encode computation”* (特别是，这些看似

[2] As a side bonus, a similar argument gives a short proof of Gödel's incompleteness theorem, a fact which for some reason is still hidden from many undergraduates taking logic courses.

静态对象编码动态过程）。今天已知这适用于许多不同的数学结构，在代数、拓扑、几何、分析、逻辑等领域，尽管所研究的结构似乎与计算完全无关。这种普遍性使得每一位数学家都可能是隐藏在伪装下的计算机科学家。我们将在稍后回到这个想法的改进版本。

自然，这样的负面结果并没有阻止对这些结构和性质进行数学研究——它们仅仅建议研究给定对象的有趣子类。对于丢番图方程的特定类别，例如费马大定理和关于椭圆曲线模性质的（7）问题的解决，理解得更好。对于数论的限制逻辑（例如，普雷舍尔算术）也是如此。

决策过程（或算法）作为理解数学问题的最小要求这一概念，也导致了直接的正向结果。它表明我们应该寻找决策过程作为$a\ means$，或者作为理解问题的$first\ step$。带着这个目标，Haken [Hak61] 展示了如何从这个意义上理解结，为问题（2）设计了决策过程，确定结的性质。同样，Tarski [Tar51] 展示了封闭实数域可以这样理解，为问题（6）设计了决策过程。自然地，为了开发这些算法，需要显著的$mathematical,\ structural$理解。Haken 发展了$normal\ surfaces$的理论，Tarski 为他们的算法发明了$quantifier\ elimination$；在两种情况下，这些思想和技巧成为了他们各自领域的基石。

这些重要的例子，以及许多类似的例子，仅仅强调了几个世纪以来显而易见的事实：数学和算法理解密切相关，常常相辅相成，正如引言中所详细讨论的那样。而且，在过去的几个世纪中正确的事情，在这个世纪仍然是正确的：算法的语言与方程和公式的语言兼容，并且实际上是对它们的推广（它们是算法的特殊情况），并且是理解和解释复杂数学结构的有力语言。

许多为基本数学分类问题开发的决策过程，例如Haken和Tarski分别对问题(2)和(6)的解决方案，证明了这种算法理解的概念$in\ principle$。毕竟，它们保证的是一个将在$finite$时间内提供正确解的算法。这能满足我们吗？有限时间可能非常长，很难区分一亿年和无限。这不仅仅是对工作数学家的一个抽象问题，如我们在第1.1节中所述。事实上，Haken和Tarski的原始算法都非常慢，计算它们对适度大小对象的答案可能确实需要一亿年。这可以量化吗?

Efficient algorithms and computational complexity 此观点建议开发和使用一个计算标准，并通过提供该标准的算法的质量来衡量理解的质量。事实上，我们认为，对给定数学结构的更好数学理解通常与分类其属性更好的算法相辅相成。将算法的资源（特别是 $time$）的概念形式化及其有效使用是计算复杂度理论的任务，这是本书的主题，我们将在下一节开始发展。但在我们这样做之前，我想使用上述问题集来强调一些其他问题，我们在这里将不再进一步讨论。

首先，算法处理的对象的 $representation$ 变得非常重要！上述大多数问题提出的一个基本问题是数学中的连续性与计算的离散性之间的对比。算法在离散的时间步中操作有限的对象（如比特）。结、流形、动力系统等是连续对象。它们如何被描述并供算法处理？正如许多读者所知，答案各不相同，但所有这些都有有限描述。例如，我们使用结图来描述结，使用三角剖分来描述流形，以及使用符号描述或连续逼近来描述动力系统。正是这些 $discrete$ 表示在算法中确实被用于这些（和其他）$continuous$ 问题（例如，Haken 的算法所示）。请注意，这必须是这样；我们将考虑的每个连续对象都有离散表示！毕竟，数学教科书和论文是来自有限字母表的有限字符序列，就像图灵机的输入一样。而我们，作为它们的读者，否则将无法处理和讨论它们。所有

这并不贬低在寻求对连续数学结构进行描述、处理和讨论时可能出现的困难——相反，它进一步说明了数学与计算之间不可避免的联系。

让我演示算法效率可能至关重要的依赖于表示，即使对于简单的 *discrete* 结构。如果我们继续使用 *unary* 整数的编码，或者甚至罗马数字，数学（和社会）会怎样？十进制编码（或更一般地，*great* 数制）的发明是由有效的算术操作算法所驱动的！这只是一个极其基本的例子。

问题3关于平面图的四色性指向了计算与数学相互作用的另一个方面。许多读者都知道问题（3）有一个非常简单的决策过程：对每个输入回答“是”。这个平凡算法的正确性由Appel和Haken [AH89] 的4色定理的高度非平凡证明所保证。该定理表明，在每一个平面图（如地理教材中使用的，以及图2所示），每个区域都可以用一组4种颜色（例如，红色、蓝色、绿色、黄色）进行着色，使得没有两个相邻的区域被分配相同的颜色。这个数学证明是第一个使用计算机程序作为基本工具来检查大量但有限的有限图确实可以四色着色的证明。这个证明自然引发了关于此类证明价值的许多讨论和争论；随着时间的推移，随着越来越多的证明以类似的方式使用计算机（另一个著名的例子是Hales对开普勒猜想的证明 [Hal05]），这些讨论和争论变得更加激烈。我将留给读者去思考计算机生成的人类生成的证明的相对优点（以及区分两者的任务）。另一个要说明的是，问题（3）可能对某些人来说与其他问题非常不同；所有这些都是数学中众所周知的“深”问题，而（3）似乎更像是娱乐性的。事实上，在20世纪，在不知道4色定理的情况下，数学界普遍将此类问题称为“平凡的”，仅仅是因为一个尝试所有（有限多个）可能着色的平凡暴力算法就能确定答案。这又把我们直接带回到了算法的质量，这是接下来要讨论的。您可能希望在阅读第3.1节后重新审视区分“深”问题和“平凡”问题的任务。

3 Computational complexity 101: The basics, $\mathcal{P}$, and $\mathcal{NP}$

在这一章中，我们发展了计算问题的基本概念；数据表示；高效计算；问题之间的高效归约；证明的高效验证；类$\mathcal{P}$、$\mathcal{NP}$和co$\mathcal{NP}$；以及$\mathcal{NP}$-完备问题的概念。我们专注于*classification* (或决策)问题；还研究了其他类型的问题和类（枚举、近似、构造等），其中一些将在本书的后续章节中讨论。

当研究效率时，我们将重点关注*time*作为算法的主要资源。我所说的“时间”是指执行的基本操作数[1]。算法的其他资源，如内存、并行性、通信和随机性，在计算复杂性中研究，其中一些将在本书的后续部分讨论。

3.1 Motivating examples

让我们考虑以下三个分类问题。正如第二章所述，对于每个这样的分类（或决策）问题，我们得到一个对象的描述，并必须决定它是否具有所需的属性。

(1′.) 哪些形如 $Ax^2 + By + C = 0$ 的丢番图方程可以用正整数解?

(2′.) 在三维流形上哪些结绑定了 genus $\leq g$ 的曲面?

(3′.) 哪些平面图是3可着色的?

问题（1′）是第二章问题（1）的一个限制。问题（1）是不可解的，因此试图更好地理解更受限制的丢番图方程类是自然的。问题（2′）以两种方式推广了第二章问题（2）：未打结问题（2）考虑了流形 $\mathbb{R}^3$ 和 genus $g = 0$ 的特殊情况。问题（2）是可解的，因此我们可能想知道其推广（2′）是否也是可解的。问题（3′）是问题（3）的一个有趣变体。虽然每个地图都有四色着色，但并非每个地图都有三色着色；有些有，有些没有，因此这是另一个需要理解的非平凡分类问题。

大多数数学家都会倾向于认为这三个问题之间绝对没有任何联系。它们分别来自非常不同的领域——代数、拓扑学和组合数学，各自拥有完全不同的概念、目标和工具。然而，下面的定理表明这种观点可能是错误的。

Theorem 3.1. *Problems* (1′) , (2′) *and* (3′) *are* 等价.

此外，等价概念是自然且完全正式的。直观上，我们对一个问题的任何理解都可以被*simply*翻译成对另一个问题的类似理解。这种等价的正式含义将在本章展开，并在第3.9节中形式化。为了达到这一点，我们需要发展出产生这种令人惊讶结果的语言和工具。

Representation issues 我们首先通过非正式的讨论和举例来说明，如何用有限的方式来描述这些多样化的复杂数学对象，最终可以表示为一系列比特。通常存在几种不同的表示方法，并且通常将一种表示转换为另一种表示很简单。让我们讨论这三个问题中输入的表示方法。

对于问题（1′），首先考虑所有形式为$Ax^2 + By + C = 0$的方程集，其中系数为整数A, B, C。此类方程的有限表示是显然的——系数的三元组(A, B, C)，例如，每个整数以二进制表示。给定这样的三元组，决策问题是对应的多项式是否有正整数根(x, y)。令2DIO表示答案为YES的三元子集。

[1]For example, reading/writing a finite amount of data from/to memory, or performing a logical or arithmetic operation involving a finite amount of data.

有限表示问题（2′）的输入很棘手但仍然自然。输入包括一个三维流形M，其中嵌入了一个结K，以及一个整数G。有限表示可以通过三角剖分（有限个四面体及其相邻关系）来描述M。结K将被描述为沿给定四面体的边的一个封闭路径。给定一个三元组(M, K, G)，决策问题是在K所界定的表面上，其亏格是否最多为G。用$KNOT$表示答案为YES的子集。

有限表示问题（3′）的输入非平凡。让我们讨论的不是映射而是图，其中顶点代表国家，边代表国家的相邻关系（这种观点是等价的；对于一个平面图，它的图就是它的对偶图）。为了描述一个图（以使其平面性显而易见的方式），一个优雅的可能性是使用Fáry [Fár48] 的简单而美丽的定理（其他人独立发现，并且有许多证明）。它表明每个平面图在平面上都有一个*straight line*嵌入（没有交叉的边）。因此，输入可以是一个顶点坐标集合V（实际上可以是小的整数）和一个边集合E，每个边都是V中元素的配对。让$3COL$是描述3可着色地图的那些输入（V, E）的子集。

任何有限对象（整数、整数的元组、有限图、有限复形等）都可以用字母表 $\{0, 1\}$ 上的二进制序列自然表示，并且这就是它们作为算法输入的描述方式。如上所述，即使是连续对象如结也有有限描述，因此也可以用这种方式描述。[2] 我们在此不讨论诸如对象是否有唯一表示或每个二进制序列是否应该表示一个合法对象等微妙问题。只需说，在大多数自然问题中，这种输入编码的选择可以使得这些问题不是真正的难题。此外，在对象及其二进制表示之间来回转换简单且高效（以下将正式定义这一概念）。

因此，让 {v*} 表示所有有限二进制序列的集合，并将其视为所有分类问题的输入集合。实际上，{v*} 的每个子集都定义了一个分类问题。在这种语言中，给定一个二进制序列 {v*}，我们可以将其解释为整数三元组 ({v*}) 并询问相关方程是否在 2{v*} 中。这是问题 (1{v*})。我们还可以将 {v*} 解释为流形、结和整数的三元组 ({v*})，并询问它是否在集合 {v*} 中。这是问题 (2{v*})，同样也可以用 (3{v*}) 来做。

Reductions 定理3.1表明，在解决（1′）和问题（2′）之间有*simple*种翻译（双向）。更确切地说，它提供了执行这些翻译的效率可计算函数f, h: $\mathbf{I} \to \mathbf{I}$:

$(A, B, C) \in 2DIO$ iff $f(A, B, C) \in KNOT$,
和
$(M, K, G) \in KNOT$ 当且仅当 $h(M, K, G) \in 2DIO$。

因此，一个有效算法解决这些问题中的一个立即意味着对另一个有类似的解决方案。更戏剧性地讲，如果我们对拓扑学有了足够的理解来解决，比如说，结的亏格问题，这意味着我们自动获得了足够的数论理解来解决这些二次丢番图问题（反之亦然）。

翻译函数 f 和 h 被称为 *reductions*。我们捕捉 *simplicity* 在 *computational* 项中的减少，要求它将是 *efficiently* 可计算的。

相似的对偶缩减存在于3-着色问题与另外两个问题之间。如果对图论有足够的理解，就能得到一个有效的算法来确定给定的平面图是否可3着色，那么对于$KNOT$和$2DIO$也会有类似的算法。反之亦然——理解其中的任何一个都会同样解决3-着色问题。请注意，这种对等价的积极解释将所有三个问题描绘为同样“可接近”。但另一方面，它们也同样是难以处理的：如果其中任何一个缺乏这样的有效分类算法，那么其他两个也是如此。事实上，随着更好的

[2] Theories of algorithms which have continuous inputs (e.g., real or complex numbers) have been developed, for example, in [BCSS98, BC06], but will not be discussed here.

对今天这些等价的了解，似乎更有可能第二种解释是正确的：这些问题都很难理解。

当在课堂上教授这些材料，或在向不明真相的听众进行讲座时，总是很有趣地观察听众对这些遥远问题之间如此强大且出乎意料的联系感到的惊讶。我希望它对你也有类似的影响。但现在，是时候揭开神秘面纱，解释这些联系的原因了。我们开始吧。

3.2 Efficient computation and the class $\mathcal{P}$

高效算法是推动工业和经济发展日益增长部分的引擎，以及它伴随着你的日常生活。这些瑰宝嵌入到你每天使用的多数设备和应用中。在本节中，我们抽象出高效计算的一个数学概念，即多项式时间算法。我们对其进行阐述并给出此类算法的例子。

在所有后续内容中，我们关注渐近复杂度。因此，例如，我们既不关心分解数字$2^{67}-1$所需的时间（就像梅森那样关心），也不关心分解所有67位数字所需的时间，而是关注分解n-位数字的渐近行为，作为输入长度n的函数。渐近观点是计算复杂度理论固有的，我们将在本书中看到，它揭示了会被有限精确分析所掩盖的结构。我们注意到，在可计算性理论中，不存在对输入大小的依赖，其中算法只需在有限时间内停止。然而，这些领域的许多方法被引入到计算复杂度理论中——问题的复杂度类、问题之间的归约和完备问题，我们将在本书中遇到。

Efficient 计算（对于给定问题）将被认为是运行时间在任何长度为 n 的输入上被一个 *polynomial* 函数在 n 中界定的。

回忆一下，$\mathbf{I}$表示所有长度的二进制序列的集合。让$\mathbf{I}_n$表示$\mathbf{I}$中所有长度为n的二进制序列，即$\mathbf{I}_n = \{0,1\}^n$。

Definition 3.2 (该类 $\mathcal{P}$)**.** 一个函数 $f : \mathbf{I} \to \mathbf{I}$ 在类 $\mathcal{P}$ 中，如果存在一个计算 f 的算法和正的常数 A, c，使得对于每一个 n 和每一个 $x \in \mathbf{I}_n$，该算法在至多 An^c 步（即基本运算）内计算出 $f(x)$。

注意，该定义特别适用于布尔函数（其输出为 $\{0, 1\}$），这些函数捕获分类问题（通常称为“决策问题”）。我们将滥用符号，有时将 $\mathcal{P}$ 视为包含 *only* 这些分类问题的类别。观察到一个具有长输出的函数可以被视为一系列布尔函数的序列，每个输出位一个。

这个重要的定义，对比了多项式增长与（暴力）指数增长，由Cobham [Cob65]、Edmonds [Edm65b,Edm66,Edm67a] 和 Rabin [Rab67] 在20世纪60年代末提出。这些来自不同领域和动机的研究人员都试图从仅*finite*算法中正式区分出*efficient*。Edmonds的论文尤其提供了一些巧妙的多项式时间算法来解决自然优化问题。当然，非平凡的多项式时间算法在计算机时代之前就已经被发现。许多是由需要高效方法（手工计算）的数学家发现的。最古老和著名的例子当然是第一章中提到的欧几里得GCD（最大公约数）算法，该算法是为了在计算最大公约数时避免对整数进行因式分解而发明的。

两个主要选择必须在选择 $\mathcal{P}$ 来建模高效可计算函数类时做出，这些选择经常被讨论，当然需要解释。一个是将 *polynomial* 作为输入长度的时间限制，二是将 *worst-case* 要求（即，这个时间限制对所有输入都成立）作为选择。我们将在下文讨论这两个选择的动机和重要性。然而，重要的是要强调这些选择并非教条：计算复杂度理论已经考虑和研究了这些选择之外的许多其他替代方案。这包括许多比多项式更细粒度的效率界限，以及许多不同的平均情况和输入相关度量的概念，以取代最坏情况的需求。其中一些将在本书的后续章节中讨论。然而，最初的这些选择

在计算复杂性的早期，上述内容极为重要，揭示了将成为该领域坚实基础的美好结构，确立了其方法论，并指导了对更精细和更多样化替代方案的研究。

3.2.1 Why polynomial?

多项式时间表示高效计算的选取似乎任意。然而，从许多角度来看，这种特定的选择在时间上已经证明了自己的合理性。我列举一些重要的理由。

多项式代表了“缓慢增长”的函数。多项式在加法、乘法和复合运算下的闭包保持了自然编程实践中的效率概念，例如按顺序使用两个程序，或使用一个作为另一个的子程序。这种选择消除了精确描述计算模型的必要性（例如，我们是否只允许在单个数字上执行算术运算或任意整数，这并不重要，因为长加法、减法、乘法和除法有简单的多项式时间算法，这在小学就已经教授）。同样，我们也不必担心数据表示：可以在本质上任何两种自然表示之间有效地进行转换。

从实用角度来看，例如n^2的运行时间远比n^{100}更受欢迎，当然线性时间更好。事实上，多项式运行时间的常数系数对于算法在实际生活中的可行性可能至关重要。然而，似乎存在一个“小数定律”：对于自然问题，已知的多项式时间算法的指数很少超过3或4（尽管在发现时，初始指数可能是30或40）。然而，许多至今仍无法找到有效算法的重要自然问题，目前无法在$exponential$时间内解决（这当然对于小型输入数据来说也是完全不切实际的）。这种指数差距为$\mathcal{P}$的定义提供了极大的动力；将这类问题的复杂度从指数级降低到（任何）多项式级将是一个巨大的概念性改进，可能涉及新技术。

3.2.2 Why worst case?

另一个对类 $\mathcal{P}$ 定义的批评是，如果一个长度为 n 的 $every$ 输入可以在 poly(n)-时间内解决，则认为该问题是可以有效解决的。从实际的角度来看，只要我们关心的实例（例如，由我们的应用程序生成的，无论是工业还是自然）能被我们的算法快速解决，其余的可以花费较长时间。也许“典型”实例快速解决就足够了。当然，理解实践中出现的实例本身就是一个大问题，并且研究了各种典型行为模型及其算法（我们将在第4.4节中涉及这一点）。最坏情况分析的明显优势是我们不必担心哪些实例会出现——它们都将被我们所说的“有效算法”快速解决。这种观念“组合”得很好（即当一个算法使用另一个算法时）。此外，它还考虑了敌对情况，其中输入（或更普遍地，外部行为）是由一个未知的对手生成的，该对手希望减慢算法（或系统）。在密码学和纠错等领域的建模中，这种敌对行为至关重要，而最坏情况分析有助于这种建模。最后，正如提到的，这种观念最终揭示了复杂性宇宙的非常优雅的结构，这激发了平均情况和实例特定理论的更深入研究。

理解类 $\mathcal{P}$ 是核心的。在理论和实践中出现了许多需要高效解决方案的计算问题。在过去四十年中，开发了众多算法技术，并能够解决这些问题中的许多（例如，参见教科书 [CLR01, KT06]）。这些技术推动了我们现在视为理所当然的超快速家用计算机应用，如网络搜索、拼写检查、数据处理、计算机游戏图形和快速算术，以及跨行业、商业、数学和科学使用的重型程序。但是，还有更多问题（其中一些我们很快就会遇到），可能具有更高的实际和理论价值，仍然难以捉摸。目前，解决这个基本数学对象 $characterizing$——有效可解问题类 $\mathcal{P}$——的挑战远远超出了我们的能力。

我们以一些来自不同领域的具有数学意义的非平凡问题示例结束本节。在每个例子中，都需要数学和计算理解之间的相互作用，这对于

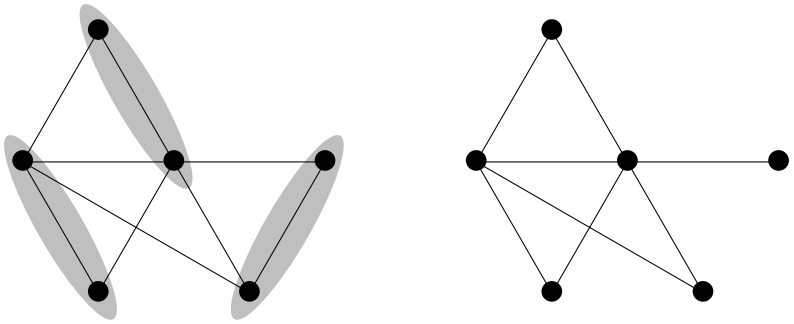


图3. 一个具有完美匹配（左侧；匹配已显示）和一个不具有完美匹配的图（右侧）。

这些算法的发展显而易见。大多数例子本质上是基础的，但如果某些数学概念不熟悉，请随意忽略该示例（或者可能更好，查阅其含义）。

3.2.3 Some problems in $\mathcal{P}$

- **Perfect matching.** 给定一个图，测试它是否有一个*perfect matching*，即所有顶点的配对，使得每一对都是图的边（见图3）。Edmonds [Edm65b] 的巧妙算法可能是$\mathcal{P}$中第一个非平凡算法，如上所述，本文在突出$\mathcal{P}$作为重要的研究类别方面发挥了核心作用。图中匹配的结构是组合数学中最研究广泛的课题之一（例如，参见[LP09]）。

- **Primality testing.** 给定一个整数，确定它是否为素数.[3] 高斯实际上向数学界呼吁寻找一个有效的算法，但这花了两个世纪才解决。Agrawal、Kayal和Saxena [AKS04] 这项最近成就及其历史在Granville的[Gre05]中被美妙地叙述。当然，没有必要详细阐述素数在数学（甚至在流行文化）中的核心地位。

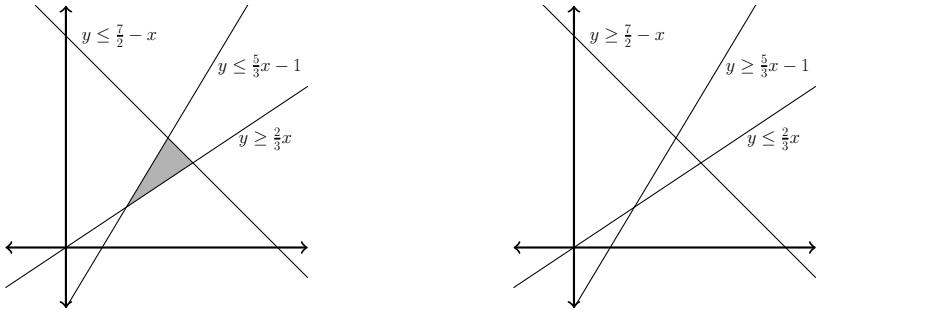
- **Planarity testing.** 给定一个图，确定它是否为*planar*。也就是说，它是否可以在平面上嵌入而不出现任何交叉的边？（尝试确定图3和图5中的图是否满足此条件。）已经发现了一系列用于解决此基本问题的*linear*时间算法，始于Hopcroft和Tarjan [HT74]的论文。

- **Linear programming.** 给定一组多变量的线性不等式，确定它们是否相互一致，即是否存在满足所有不等式的实数值变量（一个简单示例如图4所示）。这个问题及其优化版本在应用中非常有用。它捕捉了许多其他问题（例如，寻找零和博弈的最优策略）。用于为其提供高效算法的凸优化技术（[Kha79]，[Kar84]）实际上做了更多（例如，参见Schrijver的书籍[Sch03]）。

- **Factoring polynomials.** 给定一个有理系数的多项式，求其在 $\mathbb{Q}$ 上的不可约因子。Lenstra、Lenstra 和 Lovász 在 [LLL82] 中开发的工具（主要关于 $\mathbb{R}^n$ 中的格的短基）有许多其他应用（参见第 13.8 节）。

- **Hereditary graph properties.** 给定一个有限图，测试它是否是某个固定闭包的子图族成员.[4] 随后有一个多项式时间算法（在一般情况下具有巨大的指数）

[3] For example, try to determine the answer for $X-1$ and $X+1$, where $X = 6797727 \times 2^{15328}$.
[4] Namely, removing a vertex (or edge), and contracting an edge leave a graph in the family.



(a) A *feasible* linear program (b) An *infeasible* linear program

图4. 两个变量和三个不等式的两个线性规划。

Robertson 和 Seymour 的巨著结构理论 [RS95] 以及这类族的一个有限基定理。[5]

- **Permutation group membership.** 给定一个包含 n 元素排列的列表，第一个能否由其余的生成？[6] 在 [Sim70, FHL80] 中开发的非交换高斯消元技术触发了算法群论的发展，该理论被群论家广泛使用，特别是导致了测试图同构的突破 [Bab15]。[7]

- **Hyperbolic word problem.** 给定任何由生成元和关系表示的 *hyperbolic* 群，以及生成元中的一个词 w，检查 w 是否代表单位元素。Gromov 的几何技术，包括此类群的凯莱图上的等周界 [Gro87]，允许多项式时间算法（以及更多）。对于一般的有限呈现群，这个问题是不可解的。

3.3 Efficient verification and the class $\mathcal{NP}$

让 $C \subset \mathbf{I}$ 是一个分类问题。[8] 具体来说，我们被给定为输入一个描述数学对象的二进制序列 $x \in \mathbf{I}$ (，并应该确定是否 $x \in C$。将 C 视为定义一个 *property* 是方便的；具有该属性的 $x \in C$ 是对象，而没有该属性的 $x \notin C$ 是对象。如果我们有一个有效的算法用于 C，我们就可以简单地将其应用于 x 并知道 x 是否具有属性 C。但如果我们没有这样的算法，下一步最好的选择是什么？一个答案是，一个 *convincing proof*，它 $x \in C$ 。在正式定义它之前，让我们看看几个激励性的例子。

第一个例子是F. N. Cole在一次讲座中的著名轶事，题为“关于大数的分解”，在1903年美国数学学会会议上发表。他没有说一句话，走到黑板前，写下

$$2^{67} - 1 = 147573952589676412927 = 193707721 \times 761838257287,$$

然后对右侧的整数进行长乘法运算，以得到左侧的整数：梅森第67个数（被猜想为素数）。观众中没有人有任何问题。

那里发生了什么？Cole证明了数字$2^{67} - 1$是*composite*。事实上，我们可以看到，对于任何形式的（正确）断言$x \in COMPOSITES$，都可以给出如此简短的证明，其中$COMPOSITES$表示合数集。证明可以简单地是x的分解。我们想要的功能

[5] Every such family has a finite number of excluded minors.

[6] A famous special case is the question, given a color pattern of a Rubik's cube (perhaps obtained by placing colored stickers illegally), can it be sorted to monochromatic faces by legal Rubik moves?

[7] See also the exposition [HBD17] of this result.

[8] In the computer science literature, C is often called a *language*.

从这一集中提取出两个：这里的证明是 *short* 和 *easily verifiable*。实际上，这些因子的总长度大致等于输入长度，并且乘法有一个高效的算法。请注意，对于 Cole 来说，将这些因子 *find* 提出来极其困难（他说这花了“他三个年的星期天”），但这丝毫没有影响那个演示。

一个第二个例子，数学家们每天都会遇到，那就是当我们阅读一篇典型的数学期刊论文时会发生什么。在其中，我们通常找到一个（声称的）定理，然后是一个（所谓的）证明。因此，我们正在验证类型 $x \in THEOREMS$的声明，其中$THEOREMS$是所有在，比如说，集合论中可证明的陈述的集合。人们理所当然地认为，书面的证明是$short$ (页数限制)和$easily\ verifiable$ (一个审稿人可以在合理的时间内验证它)，所以至少在直观层面上$THEOREMS$具有与$COMPOSITES$相同的属性，并且这可以形式化。再次提醒，我们不在乎作者花了多长时间$find$证明。不用说，数学期刊中的定理和证明并不是真正用形式语言写成的；事实上，可以将审稿的任务解释为验证这些“半正式”的证明可以被转换为形式化的证明，从而确立它们所声称证明的陈述的真实性。

现在我们准备为 $\mathcal{NP}$ 定义一个类，它泛化这两个例子。

类 $\mathcal{NP}$ 包含所有具有 C 成员资格（即形式为 $x \in C$ 的陈述）的属性。与之前一样，我们使用多项式来定义这两个术语。对于断言 $x \in C$ 的候选证明 y 必须的长度最多是 x 长度的多项式。并且，验证给定的 y 确实证明了断言 $x \in C$ 必须在多项式时间内可验证（通过我们将称之为 V_C 的验证算法）。最后，如果 $x \notin C$，则不应存在这样的 y。让我们形式化这个定义。

Definition 3.3 (类 $\mathcal{NP}$)**.** 集合 C 在类 $\mathcal{NP}$ 中，如果存在一个函数 $V_C \in \mathcal{P}$ 和一个常数 k 使得

- 如果 $x \in C$，则 $\exists y$ 与 $|y| \leq k \cdot |x|^k$ 和 $V_C(x, y) = 1$；
- 如果 $x \notin C$，那么 $\forall y$ 我们有 $V_C(x, y) = 0$。

从一个逻辑角度来看，每个集合 C 在 $\mathcal{NP}$ 中可以被视为由 *verification process* V_C 定义的一个完整且有效的证明系统中的公理集合。

一个序列 y 被称为“说服” V_C $x \in C$ 是的，通常被称为 *witness* 或 *certificate*，因为 x 属于 C。再次强调，$\mathcal{NP}$ 的定义并不关心提出一个证人 y 的难度，而只是关心使用 y 进行高效验证 $x \in C$。如果存在证人 y (，它可以被视为由全能实体给出，或者简单地猜测。实际上，缩写 $\mathcal{NP}$ 代表“非确定性多项式时间”，其中非确定性捕捉了非确定性机器“猜测”一个证人 y (（如果存在）然后确定性验证它的能力。

尽管如此，找到证人的复杂性当然是重要的，因为它捕捉了与$\mathcal{NP}$集合相关的$search\ problem$。每个决策问题C (实际上，每个$\mathcal{NP}$中的C)验证者V_C都定义了一个与之相关的自然搜索问题：给定$x \in C$，$find$一个简短的证人y，它能“说服”V_C这一事实。这个搜索问题的正确解可以通过V_C有效地验证，按定义。

显然，如果存在证人，可以通过穷举搜索找到：因为证人是短的（对于长度-n的输入，长度为poly(n)），可以枚举所有可能的证人，并对每个证人应用验证程序。然而，这种枚举需要$exponential\ time$在n中。本章（以及本书，以及计算理论！）的主要问题是，对于$all\ \mathcal{NP}$问题，是否存在比穷举搜索快得多的算法。

虽然通常出现得更自然的是搜索问题，但研究这些问题的决策版本（即是否存在简短的证据）通常更为方便。在几乎所有情况下，决策和搜索版本在计算上是等价的.[9]

这里是一些在 $\mathcal{NP}$ 中（或者更确切地说，是属性）的问题列表（或者更确切地说，是属性），除了上面我们看到的 *THEOREMS* 和 *COMPOSITES*。请注意，其中一些是我们在为该班级提供的类似列表中问题的一些变体。

[9] A notable possible exception is the set *COMPOSITES* and the suggested verification procedure for it, accepting as witness a nontrivial factor. Note that while *COMPOSITES* $\in \mathcal{P}$ is a decision problem, the related search problem is equivalent to integer factorization, which is not known to have an efficient algorithm.

$\mathcal{P}$然而，我们不知道这些中的任何一个是 $\mathcal{P}$ 中的。对于读者来说，这是一个很好的练习（对于大多数例子来说很容易，但并非所有例子），为它们中的每一个定义具有该属性的输入的简短、易于验证的证人.[10]

某些问题在 $\mathcal{NP}$ 中:

- **Hamiltonian cycles in graphs.** 图的集合具有哈密顿回路，即通过每个顶点恰好一次的边构成的回路（见图5）。[11]
- **Factoring integers.** 整数三元组({v*})，其中x在区间[a, b]内有一个质因数。
- **Integer programming.** 许多变量的线性不等式组，具有整数解。
- **Matrix group membership.** 三元组(A, B, C)，为相同大小的可逆矩阵（例如，在$\mathbb{F}_2$上），使得A在由B, C生成的子群中。
- **Graph isomorphism.** 图同构对，即具有其顶点集之间的双射，该双射可扩展到其边集上的双射。（找出图5中哪些图对是同构的。）
- **Polynomial root**. 在 $\mathbb{F}_2$ 上次数为 3 的多元多项式，具有一个根（即，一个将变量分配给其求值为 0 的赋值）。

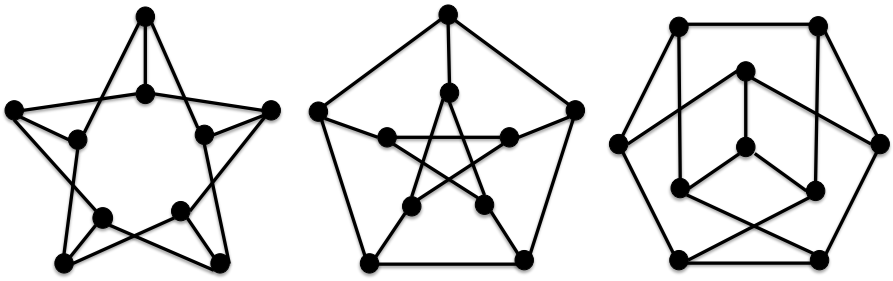


图5. 这些图中哪些是哈密顿图？哪些图的成对同构?

显然，$\mathcal{P}$中的决策问题也包含在$\mathcal{NP}$中。验证器V_C简单地取为C的有效算法，而证据y可以是空序列。

Corollary 3.4. $\mathcal{P} \subseteq \mathcal{NP}$.

但我们能否高效地解决所有 $\mathcal{NP}$ 问题？我们能否将上述提到的平凡“暴力”指数时间算法大幅改进为多项式时间，以解决所有 $\mathcal{NP}$ 问题？这就是著名的 $\mathcal{P}$ 与 $\mathcal{NP}$ 问题。

Open Problem 3.5. 是 $\mathcal{P} = \mathcal{NP}$?

$\mathcal{NP}$的定义和显式的$\mathcal{P} = \mathcal{NP}$问题（以及我们很快将了解的更多内容）最初正式出现在20世纪70年代初Cook [Coo71] 和Levin [Lev73] 的论文中，一位在美国，另一位在苏联。然而，定义和问题都早些时候以非正式的形式出现，再次独立地在东西方出现，但具有相似的动力。他们都致力于解决*tractability*问题，即解决*finite*算法的问题。

[10] The one difficult exception is Matrix Group Membership, which, if you cannot resolve yourself, peek in the beautiful [BS84].

[11] This problem is a special case of the well-known “Traveling Salesman Problem” (TSP), seeking the shortest such tour in a graph with edge lengths given.

存在，包括寻找定理的有限证明、布尔函数的简短逻辑电路、图的同构以及各种具有实际和理论兴趣的优化问题。在这些所有例子中，穷举搜索显然是一个显而易见但指数级昂贵的解决方案，目标是通过可能更巧妙（且更快）的算法来改进它，希望是多项式复杂度的（即在$\mathcal{P}$中）。关键的认识是识别出超类$\mathcal{NP}$，它如此巧妙地包含了几乎所有人们真正关心并努力解决的问题，这些问题似乎难以解决。

Sipser [Sip92] 的优秀调查描述了这一历史，并从重要的原始论文中摘录了内容。在这里，我只提到上述论文的几个先驱。在苏联，亚布洛ンス基及其学派研究了 *Perebor*，字面意思是“穷举，蛮力搜索”，而莱文的文章继续这一研究方向（参见特拉赫滕布罗特对该工作的调查 [Tra84]，包括莱文论文的修正翻译）。在西方，埃德蒙兹 [Edm66] 是第一个明确提出对短、高效可验证类型的“良好刻画”（他通过师生互动来激发这一概念，尽管比我们在第3.5节中将遇到的 $\mathcal{NP}$ 略为严格）。但早在1956年，十年前，一封引人注目的信（仅在1990年代被发现，参见 [Sip92] 中的原文和翻译）由哥德尔写给冯·诺伊曼，本质上以相当现代的语言介绍了 $\mathcal{P}$、$\mathcal{NP}$ 以及 $\mathcal{P}$ 与 $\mathcal{NP}$ 的问题（更多内容参见 [Wig10b] 的第1.2节）。特别是，哥德尔提出了克服蛮力搜索这一基本问题，举例说明它有时是非平凡的，并清楚地表明他对这一问题的意义有多么清楚。不幸的是，冯·诺伊曼当时已经因癌症去世，不清楚他是否曾回应，或者哥德尔是否有进一步的思考。有趣的是，这些早期论文对 $\mathcal{P}$ 与 $\mathcal{NP}$ 问题的解决方案有不同的预期（用他们那个时代的语言）：哥德尔推测 *THEOREMS* 可能属于 $\mathcal{P}$，而埃德蒙兹 [Edm67a] 猜测旅行商问题不在 $\mathcal{P}$ 中。

一个非常吸引人的特征是关于 $\mathcal{P}$ 与 $\mathcal{NP}$ 的问题（这曾是关于其可能迅速解决的早期乐观的来源），它可以自然地被视为计算理论中可决性问题的一个 *bounded* 类似物，我们在第2章中隐含地讨论了这个问题。为了看到这一点，将两个类中的 *polynomial-time* 界限替换为 *finite* 界限。对于 $\mathcal{P}$，类似物变成了所有具有有限算法的问题，即可决问题，有时称为 *Recursive* 问题，表示为 $\mathcal{R}$。对于 $\mathcal{NP}$，类似物是可以通过有限验证算法由有限证明者证明成员资格的属性类。这类问题被称为 *Recursively Enumerable* 或 $\mathcal{RE}$。很容易看出，第1章中提到的许多问题都属于这一类。例如，考虑第2章中定义的属性，分别是问题1和4，分别是可解的丢番图方程和可证明的佩亚诺算术定理。在前者中，多项式的整数根显然是一个可以通过有限时间评估轻松验证的有限证明者。在后者中，给定定理的佩亚诺证明是一个有限证明者，证明的推导链可以很容易地在有限时间内验证。因此，这两个问题都在 $\mathcal{RE}$ 中。我们已经知道这两个问题都是不可解的（即不在 $\mathcal{R}$ 中），因此可以得出结论 $\mathcal{R} \neq \mathcal{RE}$。

将近半个世纪的经验使我们认识到，解决 $\mathcal{P}$ 与 $\mathcal{NP}$ 的问题比解决 $\mathcal{R}$ 与 $\mathcal{RE}$ 的问题要困难得多。一个可能的类比（具有更长的历史）是解决黎曼猜想（Riemann Hypothesis）的困难，尽管我们已经知道千年以来存在无限多个素数。在这两种情况下，我们已知的是非常定性的，区分了有限和无限，而我们想要知道的是非常精确的定量版本。对于这两个问题，沿途也证明了一些弱的定量结果。素数定理是关于素数分布的比它们的无限性更精细的定量结果。在这本书中，我们将讨论关于自然问题计算复杂度的类似定量进展。在这两种情况下，长期目标似乎需要更深入的理解各自领域，以及更好的工具和技术。顺便提一下，在第8章中，我们将讨论 $\mathcal{P}$ 与 $\mathcal{NP}$ 问题与黎曼猜想之间的一种完全不同的类比。

3.4 The $\mathcal{P}$ vs. $\mathcal{NP}$ question: Its meaning and importance

你应该关心 $\mathcal{P}$ 与 $\mathcal{NP}$ 的问题吗？前几节明确指出，这是一个计算机科学中非常重要的课题。它也是一个精确的数学问题。那么，它对数学的重要性如何？对于一些数学家来说，这个问题出现在七个问题的列表中

粘土千年大奖难题[CJW06]，与黎曼猜想和庞加莱猜想（后者已被解决）一样，可能是关注的充分理由。毕竟，这些问题是在2000年由顶尖数学家选出的下个千年的主要挑战，每个问题的解决方案都附带一百万美元的奖金。

在这个部分，我希望解释$\mathcal{P} = \mathcal{NP}$问题为何不仅在克莱数学问题中独一无二，而且在所有曾经被提出过的数学问题中都是独特的，它在巨大的实用和科学重要性以及深刻的哲学内容方面。用一个（非常非正式、耸人听闻的）简单来说，可以总结如下：

Can we solve all the problems we can "legitimately" hope to solve?

在皇家"我们"可以代表任何人或所有人，代表着人类对知识和理解的一般追求的地方。特别是，这个关于$\mathcal{P} = \mathcal{NP}$问题的表述明确地解决了数学家提出的现有和未来猜想以及开放问题的可能性（至少是关于数学对象分类的问题）。

为了支持对$\mathcal{P} = \mathcal{NP}$问题的这种总体解释，让我们尝试从高层次和直观的角度理解这两个重要类别所占据的问题。事实上，我们已经在直觉上识别了类别$\mathcal{P}$，它是对我们能够解决的所有问题（例如，在我们的有生之年中，有效地）的良好近似。因此，接下来我们开始直觉地识别$\mathcal{NP}$作为所有"有趣"问题的良好近似：那些我们真正投入努力尝试解决的问题，相信我们可能能够解决。请注意，任何对这个解释的论点都必须解释为什么不可解问题（显然不在$\mathcal{P}$中）在这个意义上并不是真正"有趣"的。

这个所有（甚至大多数，甚至非常多的）"有趣"问题都可以用数学方法识别的想法无疑是大胆的。让我们慢慢考虑它。我们提醒大家，这次讨论主要是哲学性质的，我在这里提出的论点是不精确和非正式的，代表了我个人的观点。我鼓励读者对这些论点提出质疑，但我同时也挑战你们思考，这里发现的反例是否典型或例外。在本节之后，我们将很快回到数学的坚实基础上！

因此，哪些问题占据了$\mathcal{NP}$? 类$\mathcal{NP}$证明极为丰富。在数学、优化、人工智能、生物学、物理学、经济学、工业等领域，有数以千计的$\mathcal{NP}$问题自然产生，这些问题的有效解决方案将使我们受益匪浅。
What is common to all these possibly hard problems, which nevertheless separates them from certainly hard problems (like undecidable ones)?

为了探索这一点，考虑一个相关问题是值得的：*What explains the abundance of so many natural, important, diverse problems in the class $\mathcal{NP}$?* 毕竟，这个类别是由计算理论家定义为技术、数学概念的。探究$\mathcal{NP}$定义的直观意义，我们会看到它捕捉了许多人类努力的任务 *for which a successful completion can be easily recognized*。考虑以下职业，以及他们面临的典型任务（这个列表将非常肤浅，但仍然具有教育意义）：

- **Mathematician:** 给定一个数学命题，为其构造一个证明。
- **Scientist:** 给定一些现象的数据集，找到一个解释它的理论。
- **Engineer:** 给定一组约束（关于成本、物理定律等），提出一个满足这些约束的设计（如发动机、桥梁、笔记本电脑等）。
- **Detective:** 给定犯罪现场，找出"谁干的"。

考虑这些众多任务可能具有的共同特征。我声称在几乎所有情况下，当我们看到一个好的[12]解决方案时（或者我们至少相信我们能），我们可以说"我们可以告诉"它是好的。简单来说，*would you embark on a discovery process if you didn't expect to recognize what you set out to find?* 这样做是有好处的

[12]In this context, "good" may mean "optimal," or "better than previous ones," or "publishable," or any criterion we establish for ourselves.

读者应认真考虑这个陈述并尝试寻找反例。当然，在不同的环境中，上面的“我们”可能指的是学术界的成员、各种产品的消费者，或者不同审判中的陪审团。我在主题讲座之后有很多有趣的讨论，特别是关于科学家甚至艺术家是否真的处于所描述的心理状态。我相信他们是。在我看来，在这些情况下，是否向他人展示（或不展示）我们的作品的决定通常遵循对我们工作的这种“良好性测试”的应用。因此，我们着手进行任何这样的任务时，我们（隐含地或明确地）期望我们提出的解决方案（或创造）本质上要承担我们能够测试的良好的证明负担；即，要像$\mathcal{NP}$的定义一样，是*short*和*efficiently verifiable*。

$\mathcal{NP}$的丰富性源于这样一个简单的事实：这样的任务比比皆是，它们的数学表述确实是一个$\mathcal{NP}$问题。对于所有这些任务，*finding*解决方案的快速性至关重要，因此$\mathcal{P}$与$\mathcal{NP}$问题的重要性显而易见。$\mathcal{P} = \mathcal{NP}$的可能性带来的巨大影响同样明显：因为$\mathcal{P}$代表的是可以高效解决的问题，我们得出结论，所有这些任务的每一个实例都可以被高效解决.[13]对人类最紧迫问题的最优解决方案——医疗、社会、工业、科学、数学等等——将立即生成（这在Fortnow的流行书籍[For13]中有详细讨论和许多例子）。对一个精确数学问题的肯定回答是实现这一乌托邦的关键！我相信，这种普遍的承诺似乎将$\mathcal{P}$与$\mathcal{NP}$问题与其他所有曾经被问过的数学问题区分开来。

可以基于几个理由对上述强烈陈述表示怀疑。首先，虽然人类考虑的大多数问题可能是这种性质（即它们的解决方案容易识别），但使用 $\mathcal{P} = \mathcal{NP}$ (if true) 来找到这些解决方案需要高效的识别程序能够完全形式化地指定。我对这一要求的态度是，对于许多重要问题（尤其是在数学、科学和工程领域），这样的程序已经存在。对于其他重要问题，如果 $\mathcal{P} = \mathcal{NP}$ 被证明，将会有巨大的动力将直观的识别程序转化为形式化的程序以便使用。另一个怀疑可能是，$\mathcal{P} = \mathcal{NP}$ 提供的多项式时间算法在实践中可能太慢（例如，因为多项式时间界限或涉及的常数太大）。这确实可能发生，并将反过来提供巨大的动力来提高算法的效率。如第3.2节所述，目前这些问题的算法大多是穷举的指数时间算法，一个多项式时间算法（即使是不高效的）也必须代表重大的新理解，这现在应该被微调以变得更加高效。

因此，我们应该相信$\mathcal{P} = \mathcal{NP}$，以及这样的乌托邦是可实现的吗？人们觉得$\mathcal{P} = \mathcal{NP}$不太可能的一个（心理）原因是，上述任务往往需要一定程度的*creativity*或*ingenuity*，而人们并不期望一个简单的计算机程序具备这些。我们钦佩怀尔斯对费马大定理的证明；牛顿、爱因斯坦和达尔文的科学理论；金门大桥和金字塔的设计；有时甚至赫尔克里·波洛和玛普尔小姐对谋杀案的分析，正是因为它们似乎需要一种并非每个人都能跨越的飞跃，更不用说简单的机械装置了。我倾向于不同意这种特定的直觉。请注意，这些都是*specific*发现，即解决特定（高度重要）实例上的通用问题。我认为没有理由认为计算机不能做到这一点，因为毕竟人脑（以及所有自然）只是运行高效的算法，就像计算机一样（到目前为止，我们对自然的理解中没有任何东西与这一观察相矛盾，尽管有无数的猜测、写作和信仰与此相反）。所以当我们最终理解大脑的算法过程时，我们确实可能能够自动化这些特定成就的发现，也许还有许多其他成就。事实上，在人工智能的许多前沿领域最近取得的进展表明，计算机最终几乎在每一项任务上都会超越人类。

但是问题是，我们（人类或计算机）能否自动化它们 *all*。对于 *every* 这样一个验证解容易的任务，找到解也容易吗？如果 $\mathcal{P} = \mathcal{NP}$，那么答案是肯定的，这种普遍存在且可验证的创造力可以完全自动化，例如在 *every* 的情况下。大多数计算机科学家（包括我自己）认为，以下更平凡的实证原因并非如此。来自工业和学术界的数百万人的尝试

[13] In fact, a much larger class of problems, which include some without any clear way of recognizing solutions, will also become easy to solve.

已经投入证明$\mathcal{P} = \mathcal{NP}$。这项艰巨的努力大部分是无意识的，由众多独立项目组成（主要由各种应用的潜在利润所驱动，但也受到数学好奇心的驱使）以寻找*specific*优化问题的有效算法。正如第3.8节所述，如果其中*any*之一成功，那么就会得出$\mathcal{P} = \mathcal{NP}$的证明。然而，它们都失败了。这是否是足够的证据？很难说，但以下是当前普遍持有的信念：

Conjecture 3.6. $\mathcal{P} \neq \mathcal{NP}$.

为了过渡到讨论$\mathcal{P} \neq \mathcal{NP}$的世界，我提到了$\mathcal{P} = \mathcal{NP}$世界的一个重要*negative*后果，这可能使其看起来不如表面上那么乌托邦。在这个世界里，每个代码都可以被破解，实际上使我们所知的所有互联网安全和电子商务应用都变得无效。确实，正是$\mathcal{P} \neq \mathcal{NP}$的可能性催生了具有众多应用、每天都被所有人使用的*complexity-based*密码学。某种意义上，存在（特定、结构化的）硬问题，其解决方案可以轻松验证，这使得从未见过面的各方之间可以创建无法破解的代码（例如，在线购物所必需的），以及许多其他看似不可能的任务。

非常引人注目的是，那些无法解决的问题实际上有应用！因此，我们可能生活在这个$\mathcal{P} \neq \mathcal{NP}$的世界中，也有其优势。所需的困难性质在第4.5节中讨论，其全部效用在第18章关于密码学的讨论中得到了充分展示。第7.2节讨论了困难性和随机性之间的密切联系。

给定这次讨论，人们可能会想知道为什么证明这一点如此困难，即$\mathcal{P} \neq \mathcal{NP}$——似乎很明显，搜索比验证要困难得多。我们将在第5章讨论尝试这样做以及遇到的困难。在此之前，在本章中，我们开发了一种减少和完备性的方法，使我们能够识别*hardest*中的$\mathcal{NP}$问题，这些问题可能也是任何困难证明的目标。尽管这些发展和理解具有启发性和重要性，但它们似乎仍然使我们远离$\mathcal{P}$与$\mathcal{NP}$的解决。

虽然我在这里强调了这个问题的重要性，但其解决（无论是$\mathcal{P} \neq \mathcal{NP}$还是$\mathcal{P} = \mathcal{NP}$）只是故事的开始，而不是结束。正如所讨论的，这些类别相当粗糙，只是众多有趣类别中的两个。如果我们开发出解决$\mathcal{P}$与$\mathcal{NP}$的技术，人们希望它们能够被进一步细化，以更精确地确定解决特定问题所需的投资计算资源！

许多关于 $\mathcal{P}$ 与 $\mathcal{NP}$ 问题的讨论和其他观点，以及其意义和重要性，都出现在上述所有计算复杂性文本和调查中，以及 Aaronson 新发表的调查 [Aar16b] 中。

我们将有很多关于这个问题的讨论，但首先我们绕个弯，转向讨论一个与数学有强烈联系的相关问题：$\mathcal{NP}$ 与 co$\mathcal{NP}$ 的问题。

3.5 The class co$\mathcal{NP}$, the $\mathcal{NP}$ vs. co$\mathcal{NP}$ question, and efficiently characterizable structures

我们已讨论了高效计算和高效验证。现在让我们转向定义和讨论属性的高效 *characterization*。请注意，在组合数学、图论和优化中寻找“良好”的描述（一些成功的，如图中的完美匹配和欧拉回路，一些失败的，如图中的哈密顿回路和着色），对于阐明本章中概念和类别的定义和重要性至关重要。许多这些，以及正式定义“良好”概念（不仅在描述中，也在算法中）的焦点，可以追溯到Edmonds早期的优化论文，主要是[Edm66]。

修复属性 $C \subseteq \mathbf{I}$。我们已经有了解释 s

- $C \in \mathcal{P}$ 如果容易计算一个对象 x 是否具有属性 C,
- $C \in \mathcal{NP}$ 如果容易证明对象 x 具有属性 C,

我们现在加上

- $C \in \text{co}\mathcal{NP}$ 如果容易证明对象 x *does not have* 的属性 erty C,

在本文中，我们正式定义该类如下。

Definition 3.7 (类 co$\mathcal{NP}$)**.** 集合 C 在类 co$\mathcal{NP}$ 中，当且仅当其补集$\bar{C} = \mathbf{I} \setminus C$ 在 $\mathcal{NP}$ 中。

例如，所有素数的集合 $PRIMES$ 在 co$\mathcal{NP}$ 中，因为它的补集 $COMPOSITES$ 在 $\mathcal{NP}$ 中。同样，非汉密尔顿图的集合在 co$\mathcal{NP}$ 中，因为它的补集，即所有汉密尔顿图的集合，在 $\mathcal{NP}$ 中。

虽然类 $\mathcal{P}$ 的定义是对称的，[14] 但类 $\mathcal{NP}$ 的定义是 *asymmetric*。拥有一个给定的对象具有属性 C 的良好证书，绝不自动意味着拥有一个给定的对象确实具有这种属性的良好证书。

确实，当我们能够同时做到这两者，即拥有集合及其补集的良好证书时，我们正在实现数学理解结构的一个神圣目标，即*necessary and sufficient*条件，有时被称为*characterization*或*duality theorem*。正如我们所知，这样的特征化是罕见的。当我们坚持（正如我将要做的）证书还必须是*short, efficiently verifiable*的，[15]这样的特征化就更加罕见。这导致以下猜想。

Conjecture 3.8. $\mathcal{NP} \neq \text{co}\mathcal{NP}$

请注意，这个猜想意味着 $\mathcal{P} \neq \mathcal{NP}$。我们将在第6章关于证明复杂性的章节中详细讨论这个猜想的改进。

尽管这类*efficient*特征（即在$\mathcal{NP} \cap \text{co}\mathcal{NP}$中同时存在的属性）短缺，但它们并非微不足道地存在。Edmonds [Edm66] 提出了这个类别，他称之为具有*good*特征的难题。以下是遵循（分别）Menger、Dilworth、Farkas、von Neumann 和 Pratt 的重要定理的一些典型例子。我对大多数的$\mathcal{NP}$和co$\mathcal{NP}$证据进行了非正式解释，这些证据可以被认为是有效可验证的。当然，关键在于，对于这些难题中的每一个，*every*实例都拥有这样一个证据：具有该属性或违反它。

Efficient duality theorems: problems in $\mathcal{NP} \cap \text{co}\mathcal{NP}$

- **Graph k-connectivity.** 图的集合，其中 *every* 对顶点通过（给定数量）k 个不相交路径连接。这里 NP-证人是每对顶点之间的一组这样的 k 路径，而 co$\mathcal{NP}$-证人是 *cut* 个 $k-1$ 个顶点的集合，其移除会断开图中的一些对。

- **Partial order width.** 有限偏序集（poset），其最大的成对不可比较元素集*antichain* (至少有（给定数）w个元素。这里，$\mathcal{NP}$证人是一个包含w个元素的反链，而共$\mathcal{NP}$证人是将给定的偏序集划分为$w-1$ *chains* (全序集)的划分。

- **Linear programming.** 一致线性不等式系统。在这里，一个 $\mathcal{NP}$ 证人是一个满足所有不等式的点。一个 co$\mathcal{NP}$ 证人是不等式的线性组合，产生矛盾 $0 > 1$.[16]

- **Zero-sum games.**17有限零和博弈（由实支付矩阵描述）中，第一个玩家至少可以赢得给定的v (一些值)。在这里，$\mathcal{NP}$证人是对第一个玩家的策略（即对行的概率分布），这保证了她的收益为v，而共$\mathcal{NP}$证人是对第二个玩家的策略（即对列的概率分布），这保证了他的支付少于v。

[14] Having a fast algorithm to determine whether an object has a property C is equivalent to having a fast algorithm for the complementary set $\bar{C}$. In other words, $\mathcal{P} = \text{co}\mathcal{P}$.

[15] There are many famous duality theorems in mathematics that do not conform to this strict efficiency criterion (e.g., Hilbert's Nullstellensatz).

[16] This duality generalizes to other convex bodies given by more general constraints, like *semi-definite* programming. Such extensions include the Kuhn-Tucker conditions and the Hahn-Banach theorem.

[17] This problem was later discovered to be equivalent to linear programming.

- **Primes.** 素数。这里 co$\mathcal{NP}$-见证很简单：输入的两个非平凡因子。鼓励读者尝试找到 $\mathcal{NP}$-见证：一个简短的素性证书。这只需要非常基础的数论知识.[18]

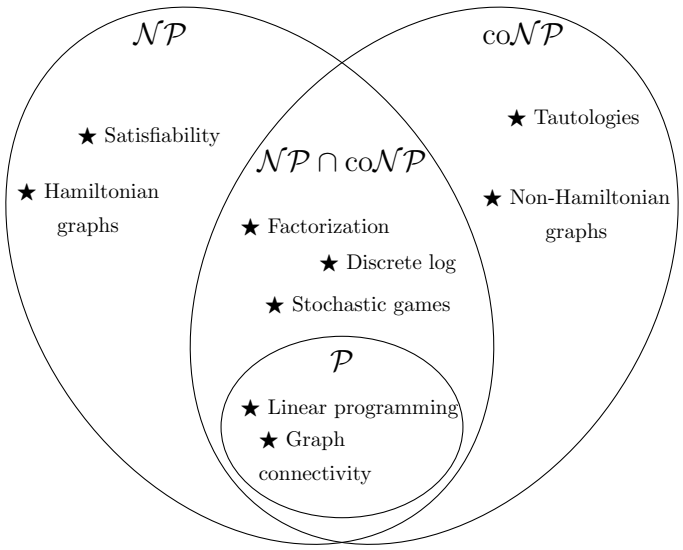


图6. $\mathcal{P}$，$\mathcal{NP}$和co$\mathcal{NP}$。

已知的 $\mathcal{P}$、$\mathcal{NP}$ 和 co$\mathcal{NP}$ 的关系在图6中描述。上述 $\mathcal{NP} \cap$ co$\mathcal{NP}$ 中的问题示例是为了说明一个观点。在它们被发现时，似乎只有对这些结构的特征描述引起了兴趣；不清楚是否也寻求了这些问题的有效算法。然而，随着时间的推移，所有这些问题都变成了 $\mathcal{P}$，它们的解决方案进入了有效算法的殿堂。著名的例子包括 Khachian [Kha79] 的椭球法和对线性规划的 Karmarkar [Kar84] 的内点法，以及 Agrawal、Kayal 和 Saxena [AKS04] 为素数设计的突破性算法.[19]

这个故事有什么寓意吗？只有一点，有时当我们对结构有一个有效的描述时，我们可以期待更多：有效的算法。事实上，通向一个难以捉摸的有效算法的自然垫脚石可能首先是对结构进行有效的描述。

我们能期待这种魔法始终发生吗？$\mathcal{NP} \cap$ 是 co$\mathcal{NP} = \mathcal{P}$ 吗？我们在 $\mathcal{NP} \cap$ co$\mathcal{NP}$ 中没有太多例子，这些例子的问题已经抵抗了有效的算法。其中一些著名的，如整数分解和离散对数，[20] 来自于 *one-way* 函数，这些函数是密码学的基础（我们在第4.5节中讨论了这些）。请注意，尽管它们并不被认为是困难的，但人类实际上是在它们的不可解性上建立了电子商务的安全性。另一个著名的例子是，对于 $\mathcal{NP}$ 和 co$\mathcal{NP}$ 的成员资格高度非平凡（分别在 [Lac15] 和 [HLP99] 中证明）的是 *unknottedness* 问题，即测试一个结图是否代表平凡结。一个非常不同的例子是康登在 [Con92] 中研究的 Shapley 的 *stochastic games*，对于这个例子，还没有已知的有效算法。然而，我们已经看到，许多最初被证明属于 $\mathcal{NP} \cap$ co$\mathcal{NP}$ 的问题最终被发现属于 $\mathcal{P}$。从这么少的例子中很难进行概括，但普遍的看法是这两个类别是不同的。

[18] Hint: Roughly, the witness consists of a generator of Z_p^*, a factorization of $p-1$, and a recursive certificate of the same type for each of the factors.

[19] It is interesting that assuming the Generalized Riemann Hypothesis, a simple polynomial-time algorithm was given 30 years earlier by Miller [Mil76].

[20] Which have to be properly defined as decision problems.

Conjecture 3.9. $\mathcal{NP} \cap \text{co}\mathcal{NP} \neq \mathcal{P}$

请注意，这个猜想意味着 $\mathcal{P} \neq \mathcal{NP}$。
我们现在回到开发主要机制，这将帮助我们研究以下问题：*efficient reductions* 和 *completeness*。

3.6 Reductions: A partial order of computational difficulty

在这个部分，我们处理将我们尚未找到有效解决方案的问题的计算难度相关联的问题。

回忆一下，我们可以将任何分类问题（在有限描述的对象上）视为我们的输入集$\mathbf{I}$的子集。有效的归约提供了一种自然的部分顺序，可以捕捉这些问题的相对难度。请注意，归约是计算可计算性和递归理论中的主要工具，从其中发展出了计算复杂性。在那里，归约通常是简单的*computable*函数，而计算复杂性的焦点是*efficiently computable*函数。虽然我们在这里关注时间效率，但该领域研究了许多其他资源；在归约中限制这些资源与在算法中限制它们一样富有成效。以下关键定义在图7中展示。

Definition 3.10 (高效归约)**.** 设$C, D \subset \mathbf{I}$为两个分类问题。$f\colon \mathbf{I} \to \mathbf{I}$是从$C$到$D$的一个高效归约，当且仅当$f \in \mathcal{P}$并且对于每个$x \in \mathbf{I}$，我们有$x \in C$当且仅当$f(x) \in D$。在这种情况下，我们称$f$为从$C$到$D$的*efficient reduction*。如果存在从C到D的高效归约，我们写$C \leq D$。

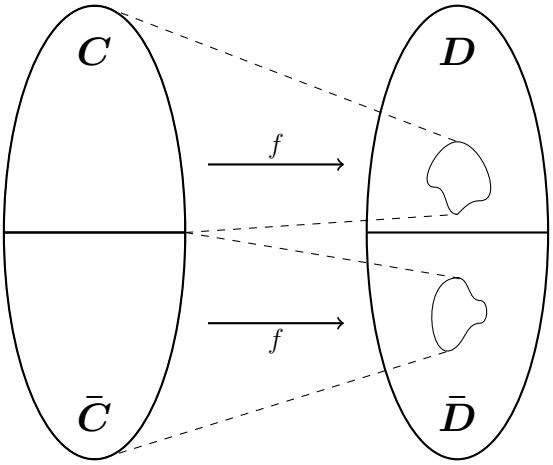


图7. 两个分类问题之间减法的一个示意图。

因此，如果 $C \leq D$，则解决分类问题 C 在计算上并不比解决 D（难多少（在运行时间上至多项式因子）。[21] 如果我们同时拥有 $C \leq D$ 和 $D \leq C$，那么 C 和 D 在计算上是等价的（再次，至多项式因子）。这为定理3.1中的“等价”一词赋予了形式意义。

高效计算的定义（更确切地说，两个多项式的复合是一个多项式）使得我们可以立即观察到关于高效归约的有用性。请你自己验证这两点！首先，确实 $\leq$ 是 *transitive*，因此定义了分类问题上的一个偏序。其次，可以组合（如图8所示）一个问题的有效算法和从第二个问题到第二个问题的有效归约，以获得第二个问题的有效算法。具体来说，如果 $C \leq D$ 和 $D \in \mathcal{P}$，那么也 $C \in \mathcal{P}$。

[21] In particular, if $C \in \mathcal{P}$ then it is not much harder than trivial problems D (e.g. D might ask to distinguish sequences starting with 0 from those starting with 1). The reduction in this case would simply solve C.

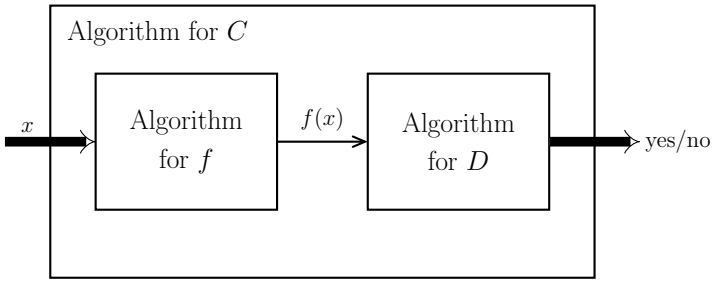


图8. 通过组合一个减少和一个算法来创建一个新的算法。

如所述，$C \leq D$ 表示解决分类问题 C 在计算上并不比解决 D 难多少。在某些情况下，可以用（模糊的）术语“数学上”来代替“计算上”。为了在数学上具有实用性并允许更好地从另一个问题的角度理解一个问题，可能需要比仅仅高效可计算更多的 f 的性质。例如，我们可能希望 f 是一个线性变换，或者一个低次多项式映射，实际上在某些情况下（如我们将在第 12 章中看到的那样）这是可能的。当两个分类问题（看似无关）之间的这种联系可以证明时，它意味着一个领域的技巧可以转移到另一个领域。

高效缩减将看似无关的概念联系起来，其力量将在后续章节中展开。我们将看到，它们不仅可以关联分类问题，还可以关联诸如从困难到随机性；平均情况到最坏情况难度；证明长度到计算时间；以及最后但同样重要的是，电子交易的安全性到分解整数的难度等这样多样化的概念。从某种意义上说，*efficient reductions are the backbone of computational complexity*。事实上，鉴于多项式时间缩减可以做所有这些奇迹，难怪我们很难描述类别$\mathcal{P}$!

3.7 Completeness: Problems capturing complexity classes

我们现在回到分类问题。通过有效的约简提供的难度偏序，使我们能够定义给定问题类中的 *hardest* 问题。设 $\mathcal{C}$ 为任何分类问题集合（即，$\mathcal{C}$ 的每个元素都是 C 的子集 $\mathbf{I}$）。在本章中，我们主要关注类 $\mathcal{C} = \mathcal{NP}$。但本书稍后我们将看到这个重要思想在其他 *complexity classes* (中反复出现，即定义由解决它们所需资源的问题类，如 $\mathcal{NP}$) 。

Definition 3.11 (硬度与完备性).一个问题D被称为$\mathcal{C}$-*hard*，如果对于每一个$C \in \mathcal{C}$，我们都有$C \leq D$。如果我们进一步有如果$D \in \mathcal{C}$，那么D被称为$\mathcal{C}$-*complete*。

换句话说，如果 D 是 $\mathcal{C}$-完备的，那么它是类 $\mathcal{C}$ 中最难的问题：如果我们设法高效地解决 D，那么我们也就自动解决了 $\mathcal{C}$ 中所有其他问题。

它不是先验地清楚一个给定的类有任何完备问题！这可能是计算复杂性的奇迹，许多自然*complexity*类都有！请注意，一个给定的类可能有多个完备问题，并且根据定义，它们都具有本质上相同的复杂性。如果我们设法证明其中*any*个不能被有效地解决，那么我们就自动地证明了*all*个。

它是平凡的（并且无趣的），在类 $\mathcal{P}$ 中的每个问题实际上在我们的定义下都是 $\mathcal{P}$-完全的。当我们发现这样的通用问题在那些我们没有有效算法的问题类别中时，它才变得有趣。到目前为止，所有这些类别中最重要的一个就是 $\mathcal{NP}$。

3.8 $\mathcal{NP}$-completeness

如前所述，Cook [Coo71] 和 Levin [Lev73] 的开创性论文定义了 $\mathcal{NP}$，有效可归约性和完备性，但他们的最高成就发现了 *natural* $\mathcal{NP}$-完备问题。

Definition 3.12 (问题 SAT). 布尔公式是关于布尔变量（可以在 $\{0,1\}$ 中取值）的逻辑表达式，其中连接词 $\wedge, \vee, \neg$ (代表 AND、OR、NOT)，例如，$(x_1 \vee x_2) \wedge (\neg x_3)$。令 SAT 表示所有 *satisfiable* 布尔公式（即变量布尔赋值求值为 1 的那些公式）的集合。

例如，以下公式是不可满足的:

$$(\neg x_1 \vee x_2) \wedge (\neg x_2 \vee x_3) \wedge (\neg x_3) \wedge (x_1 \vee x_3 \vee x_4) \wedge (\neg x_4 \vee x_1),$$

当下面的公式可满足时:

$$(\neg x_1 \vee x_2) \wedge (\neg x_2 \vee x_3) \wedge (x_1 \vee x_3 \vee x_4) \wedge (\neg x_4 \vee x_1).$$

我们现在到达计算复杂性的一个基础定理，揭示了满足布尔公式的这个看似简单的问题的重要性。

Theorem 3.13 [Coo71, Lev73]. *SAT is $\mathcal{NP}$-complete.*

回忆那个陈述的意义。对于 *every* 设置 $C \in \mathcal{NP}$，存在一个有效的约简 f: $\mathbf{I} \to \mathbf{I}$，具有以下性质：对于每个序列 $z \in \mathbf{I}$，我们有 $z \in C$ 当且仅当序列 $f(z)$ 编码一个可满足公式。

这个定理的证明（即，构造约简算法 f）带来一个额外的奖励，结果证明它极其有用：它将证人映射到证人.[22] 更精确地说，对于任何证明证人 y，它通过某个验证器 V_C) 证明 $z \in C$ (，相同的约简 f 将证人 y 转换为满足公式的变量 $f(z)$ 的布尔赋值。换句话说，这个约简不仅在不同的问题 *decision* 之间转换，还在相关的问题 *search* 之间转换。

一些关于定理3.13证明的讨论是必要的。当然，可以看出SAT包含在$\mathcal{NP}$中；一个令人满意的赋值是一个易于验证的证人。困难的部分是证明SAT的$\mathcal{NP}$难性。当然，证明不能单独考虑每个问题$C \in \mathcal{NP}$。证明的精髓是一个*generic*转换：对一个C的验证者V_C的描述进行操作，并模拟它在输入z和假设的证人y上的计算，以有效地创建一个布尔公式$f(z)$ (，其变量是y)的位。构建的公式简单地测试(z, y)上V_C的计算的有效性，并检查这个计算输出1。在这里，算法（例如，作为图灵机描述）的局部性和步骤的简单性起着核心作用：检查V_C计算的每一步的一致性本质上相当于几个位上的常数大小公式。

总结来说，SAT 捕捉了整个类 $\mathcal{NP}$ 的难度。特别是，$\mathcal{P}$ 与 $\mathcal{NP}$ 的问题现在可以表述为关于 *one* 问题复杂性的一个问题，而不是无限多个问题。

Corollary 3.14. *$\mathcal{P} = \mathcal{NP}$ iff $SAT \in \mathcal{P}$.*

一个手头有一个完整问题（如 SAT）的巨大优势是，现在，为了证明另一个问题（比如说，$D \in \mathcal{NP}$）是 $\mathcal{NP}$-完全的，我们只需要设计一个从 SAT 到 D (的缩减，即证明 $SAT \leq D$)。我们已知对于 *every* $C \in \mathcal{NP}$，我们有 $C \leq SAT$，而 $\leq$ 的传递性负责处理其余部分。

这个想法在下一篇开创性的论文中被有力地使用，即Karp的论文[Kar72]。在他的论文中，Karp列出了来自逻辑、图论、调度和几何学的21个问题，并证明了它们是$\mathcal{NP}$-完全的。这是首次展示了$\mathcal{NP}$-完全问题广泛谱系的演示，并引发了一个寻找更多此类问题的行业。几年后，Garey和Johnson [GJ79] 出版了关于$\mathcal{NP}$-完全性的书籍，其中包含来自科学、工程和数学各个分支的数百个此类问题。如今，已知这些问题有数千个。我们很快将讨论这个概念的意义和重要性，但首先我给出一些$\mathcal{NP}$-完全问题的例子，以及它们之间联系的本质。

[22] We will see one crucial use of this property in the proof of Theorem 10.5 on *zero-knowledge proofs*.

3.9 Some $\mathcal{NP}$-complete problems

我们再次强调，所有 $\mathcal{NP}$-完全问题在非常强的意义上是等价的。任何解决一个问题的算法都可以简单地转换成同样高效的[23]算法来解决任何其他问题。

我们最终准备好查看第3.1节中我们动机示例等价性的定理3.1的证明。以下三个定理可以得出结论。

Theorem 3.15 [AM75]. *The set* $2DIO$ *is* $\mathcal{NP}$*-complete.*
Theorem 3.16 [AHT06]. *The set* $KNOT$ *is* $\mathcal{NP}$*-complete.*
Theorem 3.17 [Kar72, Sto73]. *The set* $3COL$ *is* $\mathcal{NP}$*-complete.*

回忆一下，为了证明一个集合的 $\mathcal{NP}$-完备性，必须证明两件事：它属于 $\mathcal{NP}$，并且它是 $\mathcal{NP}$-难。在几乎所有 $\mathcal{NP}$-完备问题中，证明 $\mathcal{NP}$ (的成员资格，即存在简短证书)，是容易的。当然，给定地图的候选 3-着色是简短且容易检查的。对于 $2DIO$，可以很容易地看出，如果存在 $Ax^2 + By + C = 0$ 的正整数解 (x, y)，那么实际上存在一个简短的解，[24] 事实上是一个长度（以比特为单位）与 A, B, C 的长度成线性关系的解。因此，简短证据只是一个根 (x, y)。但是 $KNOT$ 是一个例外，对于具有小亏格的结的简短证据需要 Haken 的正常表面算法理论，大大增强（甚至在 $\mathbb{R}^3$ 中获得无结的简短证书也是困难的；参见 [HLP99]）。让我们讨论这些 $\mathcal{NP}$-完备性结果意味着什么，首先是关于三个集合之间的关系，然后是关于每个集合的单独讨论。

这些问题的完备性证明是通过从（SAT的变体）进行归约得出的。这些归约的离散、组合性质可能会让人怀疑这些问题的计算等价性是否意味着在拓扑学和数论之间等实际“技术转移”的能力。尽管如此，既然我们已经知道了它们的等价性，也许可以在这些问题之间找到更简单、更直接的归约。此外，我们再次强调，归约不仅转换了证据。具体来说，对于任何实例，比如（M, K, G）$\in KNOT$，如果我们使用这个归约将其转换为实例（A, B, C）$\in 2DIO$，并且（要么纯粹靠运气，要么是那个方程的特殊结构）找到了一个整数根，那么同样的归约将把这个根转换回一个描述包围结K的类G流形的过程。如今，数学中已知许多这样的 $\mathcal{NP}$-完备问题，对于某些配对，这种等价性在数学上可能是有意义和有用的（就像某些计算问题的配对一样）。

让我们讨论上述三种简化中最简单的一种，即从 SAT 到 $3COL$。如果你从未见过，这应该是个谜：这两个问题谈论的是不同的世界，一个是逻辑，另一个是图论。这两个问题都很困难，但简化应该很容易（即高效可计算）。这个简化以及几乎所有其他简化的关键是计算的局部性！这在 SAT 中当然是显而易见的；一个公式是由布尔门组成的，每个门都执行一个简单、局部的操作。然而，$3COL$ 感觉更像是一个更全局的性质.[25] 这个简化的想法是关注输入公式的单个门。我们将找到一个适用于每个门的简化，并将产生的微小（“小工具”）图组合起来，模仿输入公式规定的结构。让我们详细阐述这个想法。

这里是如何将（平凡，1门）公式 $x \vee y$ 的可满足性问题转化为图3着色问题。实际上，我们将方程 $x \vee y = z$ 转换为图3着色陈述，使用图9中所示的装置图。检查它是否满足以下条件：在 *every* 中，具有颜色 $\{0, 1, 2\}$ 的图的合法3着色中，标记为 x, y, z 的顶点的颜色将从 $\{0, 1\}$ 中选择，这将满足方程 $x \vee y = z$。也可以轻松地为门 $\wedge$，$\neg$ 构造这样的装置。现在，为了完成约简，算法按以下方式进行。给定一个任意公式作为输入，它命名其线，为每个门构建一个装置图，并在其中识别适当的顶点以生成输出图。通过构造，它只有在给定的公式是可满足的情况下才是3可着色的。这本质上

[23] As usual, up to polynomial factors.
[24] Hint: If (x, y) is a root, so is $(x + B, y - A(2x + B))$.
[25] Consider, for example, a cycle on n vertices, where n is odd; it requires 3 colors, but if we remove any edge, it can be 2-colored.

在[Kar72]中的减少，但我们还没有完成：上面的设备图不是平面的（因此输出图也不是平面的）。然而，Stockmeyer [Sto73] 提供了另一个设备，可以在不改变图的3可着色性的情况下消除平面嵌入中的交叉。鼓励读者找到这样的设备。有了这个，定理3.17的证明就完成了。

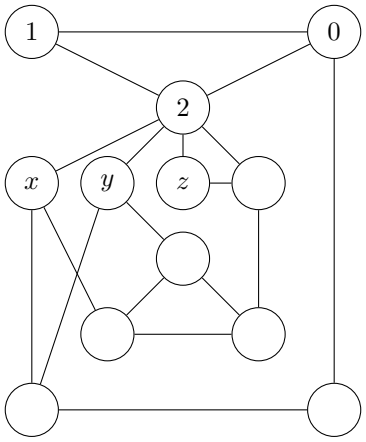


图9. 从SAT到3COL的减少背后的装置。

我们现在列出一些不同性质的$\mathcal{NP}$-完备问题，以感受这一现象的广度。其中一些已在Karp的原文章[Kar72]中出现。再次，Garey和Johnson的书中[GJ79]中可以找到数百个，而现在，已知的有数万个：

- **Hamiltonian cycle.** 给定一个图，是否存在一个简单循环，其边恰好通过每个顶点一次?
- **Subset-sum.** 给定一个整数序列 $a_1, \ldots, a_n$ 和 b，是否存在一个子集 J 使得 $\sum_{i \in J} a_i = b$?
- **Integer programming.** 给定一个在 $\mathbb{R}^n$ (中的多面体，由其边界超平面) 确定，它是否包含一个整数点?
- **Clique.** 给定一个图和一个整数 k，是否存在 k 个顶点，使得所有顶点两两互为相邻?
- **Quadratic equations.** 给定一个最多为2次的多元多项式方程组，在有限域（例如，$\mathbb{F}_2$）上，它们是否有公共根?
- **Shortest lattice vector.** 给定一个在 $\mathbb{R}^n$ 中的格 L 和一个整数 k，L 的最短非零向量是（欧几里得）长度 $\leq k$ 吗?

3.10 The nature and impact of $\mathcal{NP}$-completeness

$\mathcal{NP}$-完整性是一个独特的科学发现——似乎没有精确定义的科学概念能够在如此多的科学和工程领域普遍存在！我们从它在计算机科学本身的最直接影响开始，然后转向数学，再到科学以及更远的地方。随着你继续阅读本书，其中一些讨论将变得更加有意义（并且令人印象深刻），在最后一章中更是如此。更多内容可以在，例如，Papadimitriou关于该主题的回顾性文章[Pap97]中找到。有趣的是，那篇论文报告说，电子搜索（在当时是新的）揭示了包含短语“$\mathcal{NP}$-complete”的数千篇科学和数学论文；今天，20多年后，这个数字要大得多!

如上所述，从Karp的论文[Kar72]开始，$\mathcal{NP}$-完备性结果在计算机科学的每个角落和子领域中迅速爆发。这很容易解释。在大多数个体中

计算机科学领域，从学者到行业程序员，都在忙于寻找解决众多计算问题的有效算法。如何证明找不到这样的算法呢？在没有证明不可解性的技术的情况下，最好的办法是证明手头的计算问题是$\mathcal{NP}$-完全的（或$\mathcal{NP}$-难），这意味着找到这样的有效算法将意味着找到许多其他问题的有效算法，而许多其他问题未能解决。简而言之，未能证明$\mathcal{P} = \mathcal{NP}$是一个非常有力的借口，$\mathcal{NP}$-完全性是一个出色的困难证明。该领域的每个专业人士都知道这一点！虽然$\mathcal{NP}$-完全性是一个负面结果（基本上表明我们想要的是不可能的），但这些负面结果产生了积极影响。因为当你宣布问题为$\mathcal{NP}$-完全时，问题并没有消失，仍然需要解决方案，因此为这些问题开发了较弱的解决方案概念。例如，对于优化问题，人们试图找到好的近似算法。此外，鉴于$\mathcal{NP}$-完全性仅捕获最坏情况下的难度，人们开发了在“平均”情况下表现良好的算法和似乎在“实践中出现”的输入上表现良好的启发式算法。这个方向需要更多的理论，特别是在*deep networks*最近的实际成功远未得到理解的情况下。为不同放松有效可解性的情况开发了各种质量标准和模型，导致了类似的复杂性理论，使研究人员能够论证困难；这些模型中的一些将在本书的后面讨论。一个主要的这种理论，能够论证近似问题的$\mathcal{NP}$-完全性，并在许多情况下精确地确定有效近似的极限，将在第10.3节中讨论。

下一个受到$\mathcal{NP}$-完备性影响的领域是数学。经过一些延迟，$\mathcal{NP}$-完备性定理开始在大多数数学学科中显现，包括代数、分析、几何、拓扑、组合数学和数论。这种“入侵”可能看起来令人惊讶，因为数学家们提出的大多数问题都不是算法性的。然而，各种对象的存 在性定理要求对这些对象有“明确”的描述。此外，在许多领域，实际上需要找到这样的对象。数学充满了各种各样的构造，这些构造在很久以前是手工完成的，现在则由许多对进步至关重要的计算机程序库完成，因此，它们的效率也是至关重要的。像计算机科学家一样，数学家们采用了多项式时间算法的概念，作为定义“有效”和“明确”的第一步。因此，$\mathcal{NP}$-完备的描述和构造问题对于限制实现这些属性的希望极为有用。此外，在数学中，这样的$\mathcal{NP}$-完备性结果意味着所研究结构的基本“数学复杂性”。例如，如第3.5节所述，一个$\mathcal{NP}$-完备属性的 有效描述将意味着$\mathcal{NP} = \text{co}\mathcal{NP}$，因此（按照我们今天的理解），这样的描述不太可能存在。正如在计算机科学中一样，在数学中也是如此，这样的坏消息产生了好的结果，促使数学家们朝着更富有成效的方向发展：细化或专门化所研究的属性，考虑各种近似概念，或者简单地满足于充分必要条件，这些条件不是互补的（如用于描述所需）。

科学中 {v*} 的存在及其影响，通过以下事实得到证明：在生物学、化学、经济学、神经科学、电气工程和其他领域，这些明显关于计算的结果（通常由计算机科学家证明）不是由生物学家、化学家、经济学家、神经科学家、电气工程师等证明的。此外，这些结果还发表在这些领域的科学期刊上。数字令人震惊：搜索包含“$\mathcal{NP}$-完备”或“$\mathcal{NP}$-完备”等关键词的论文，发现这些学科中有 *hundreds* 篇这样的论文，还有成千上万篇在论文正文中提到这些术语。为了获得这些结果，这些成千上万的科学家需要学习计算复杂性的概念和证明方法，这对大多数人来说通常是一门外语（例如，数学定理很少出现在科学文章中）。

这种现象需要解释！确实，有两个问题需要回答。是什么解释了这些不同学科中$\mathcal{NP}$-完备性的丰富性？为什么他们的科学家们还要费力去证明这些计算定理？

一个重要的观察是，科学家们经常研究过程并试图建立解释和预测它们的模型。几乎可以说是定义上的，这些是计算过程，即由一系列 *simple, local steps* 组成，就像图灵机一样，尽管它们在计算机中操作的不是比特而是可能的神经元

在大脑中，细胞中的蛋白质，物质中的原子，鱼群中的鱼，或星系中的星星。换句话说，许多模型只是描述自然界用于生成某些过程或行为的*algorithms*。典型的$\mathcal{NP}$-完备性结果通常指的是某个自然过程特定模型的预测极限。以下是一些说明性的例子。在一些现有模型中，计算以下量是$\mathcal{NP}$-完备的：给定泡沫将沉降的最小表面积（在物理学中），一定分子的最小能量配置（例如，在生物学中的蛋白质折叠），以及某些均衡状态的最大社会福利（在经济学中）。让我们探讨这些结果对模拟自然现象的意义。

我们将做出两个自然假设，这似乎是完全良性的。首先，即$\mathcal{P} \neq \mathcal{NP}$。然后$\mathcal{NP}$-完备性意味着没有有效的算法可以计算所需的数量（例如，在刚刚提到的例子中），至少对于 *some* 实例来说是这样。其次，自然过程本质上是有效的算法，而我们执行以提取这些数量的测量也是如此。这两个假设相互冲突，这似乎至少暗示了两个可能的结论之一。一种可能性是，该模型（对于该模型已证明$\mathcal{NP}$-完备性）在描述现实方面是错误的（或不完整的）。另一种可能性是，“困难”的实例在自然界中根本不会发生.[26] 在这两种情况下，$\mathcal{NP}$-完备性要求我们更好地理解（例如，对当前模型进行细化），描述算法可以有效地解决实例的特征（以及这些条件与我们在自然界中看到的一致性的论点）。

这个想法导致一些研究人员提出，我们的基本猜想$\mathcal{P} \neq \mathcal{NP}$应该被视为一个*law of nature*。也许首次明确出现这种情况的引用是Volker Strassen为Les Valiant在Nevanlinna奖上的颂词[Str86]中的这句话[27]：
“The evidence in favor of Cook’s and Valiant’s hypotheses is so overwhelming, and the consequences of their failure are so grotesque, that their status may perhaps be compared to that of physical laws rather than that of ordinary mathematical conjectures.” 注意，在当时，这个数学猜想对科学的价值并没有像今天这样被充分理解。如何精确地将这个数学陈述表述为一条相关的自然法则当然仍然值得讨论。一种直观的愿望是让它扮演与热力学第二定律在科学中扮演的相同角色：科学家们会非常谨慎地提出违反它的模型。Scott Aaronson的一个建议是关于现实世界的一个更强有力的陈述：*There are no physical means to solve $\mathcal{NP}$-complete problems in polynomial time.* 在[Aar05]中讨论了许多实际尝试的可能物理手段。

仍然存在$\mathcal{NP}$-完备性在CS、数学以及几乎所有科学领域的普遍存在的神秘性。回顾过去，这是所有这些学科中不可判定性普遍性的放大（并且更加相关）的表现。这两个现象都可以通过以下事实来解释：计算（如上所述，被视为通过一系列简单、局部步骤演化的任何过程）是如此普遍。同样，对具有许多部分的系统（无论是期望的性质还是观察到的性质，这些通常是此类计算的结果）的描述通常是通过整个系统的小子系统的简单、局部约束集来给出或建模的。事实上，对于几乎所有的约束选择，如果系统是无限的，则给定实例的相互满足是不可判定的，如果系统是有限的，则是$\mathcal{NP}$-完备的。在更罕见的情况下，它们会导致（分别）可判定或多项式时间内可解的问题。理解这些现象，并划分可处理/不可处理障碍上的约束类型，是活跃的研究领域，将在第4.3节中进一步讨论。

结论这一部分，我们注意到$\mathcal{NP}$-完备性的另一个重大影响。即，它为许多其他计算通用性的概念树立了榜样。$\mathcal{NP}$-完备性证明是一个极其灵活和可扩展的概念，允许许多变体，这使得能够在其他（主要是计算，但不仅限于此）环境中捕捉通用性。它导致了使用非常不同的资源界限解决问题的类别的定义。在大多数情况下，这些类别也被证明具有完备问题，在自然归约下捕捉整个类别的难度，并具有上述好处（一些例子将在第4.1节中讨论，然后）

[26] For example, it is quite possible that over billions of years of evolution, only proteins that are easily and efficiently foldable survived, and others became extinct.

[27] In this quote Cook’s hypothesis is $\mathcal{P} \neq \mathcal{NP}$, and Valiant’s hypothesis is what became known as $\mathcal{VP} \neq \mathcal{VNP}$ which we will discuss in Chapter 12.

后续章节）。计算复杂性的整个演变、算法理论以及理论计算机科学中的大多数其他领域，都受到了强大的归约和完备性方法论的指导。这种进展产生了多种在不同环境和工具中关于 *intractability* 的理论，以及理解和可能遏制或规避它的方法（尽管我们大多数情况下仍然无法证明不可解性）。这种强大方法论揭示的结构向其他学科传递了重要信息。

我感到这条信息对科学的影响最为强烈。我将在第20章中详细阐述这个观点，并在本段中总结。本章的讨论建议将计算复杂性方面整合到自然过程的*every*模型中。换句话说，科学家们将不仅考虑事物演化的机制以及物理量的影响方式，还要考虑在该演化过程中消耗的相关计算资源量。这样做将增加约束条件，可能引导科学家们找到更好的模型，自然界实际上可以“执行”的模型。当在科学模型中考虑计算约束成为标准做法时，科学实践的方式将发生革命。虽然这种范式转变需要时间才能在科学界传播，但它正在进行中！上述许多科学论文中提到的$\mathcal{NP}$-完备性只是开始，既说明了复杂性对科学的重要性，也说明了科学家们接受它的意愿。但除此之外，在越来越多的作品中（实际上，主要是计算机科学家与其他领域科学家的合作作品），我们看到将计算方面整合到自然过程描述中的模型，以及这种融合如何导致全新的科学洞察。代表这一趋势的书籍、综述和文章包括[NRTV07, EK10, Val00, Kar11, HH13, Val13, CLPV14, Pap14]等众多其他作品。例如，[HH13]，由物理学家Daniel Harlow和Patrick Hayden撰写，提供了一个基于复杂性理论的解释，可能是解决著名的“黑洞防火墙悖论”的唯一途径。关于这些令人兴奋的发展的更多内容可以在第20章中找到。

4 Problems and classes inside (and around) $\mathcal{NP}$

本章涉及计算复杂性研究中 $\mathcal{P}$ 与 $\mathcal{NP}$ 问题的不同方面和变体，以及这两个主要类别的邻近复杂性类别，以及关于它们的中心问题。第一部分简要列举了几种不是分类问题的问题类型，这些类型主要导致包含 $\mathcal{NP}$ 的类别。其余部分致力于在 $\mathcal{NP}$ 中最简单和最困难问题之间的潜在广阔宇宙中的问题和类别，即 $\mathcal{P}$ 和 $\mathcal{NP}$-完备之间的宇宙。我们讨论了该宇宙中问题中间复杂度的程度，以及 *constraint satisfaction problems* (CSP)，对于这些可能没有这样的中间地带。最后两部分从另一个角度讨论了相同的宇宙，这由计算复杂性的平均情况分析和密码学的更严格的计算需求所激发。

4.1 Other types of computational problems and complexity classes

存在许多其他类型的计算问题，它们不属于类别 $\mathcal{NP}$，自然产生，并在理论和实践中都得到了深入研究。其中最自然的一些类型是：

- **Optimization problems.** 修复一个 $\mathcal{NP}$ 问题并在解（证）上定义一个成本函数。给定一个输入，找到它的 *best* 解（例如，找到最大的团，最短路径，最小能量配置）。优化问题的自然放松是要求一个近似解，即找到一个“接近”最佳解的解。允许理解有效近似极限的复杂性理论是计算复杂性中最激动人心的进展之一，始于 $\mathcal{PCP}$ 定理 10.6。这些问题在 4.3 节和 10.3 节中讨论。

- **Quantified problems.** 一个用于$\mathcal{NP}$ (的完整集合，通过缩减)来表征它，是SAT，即所有在变量x中的公式F的集合，其中$\exists x\, F(x)$。同样，co$\mathcal{NP}$的完整集合是所有在变量x中的公式F的集合，其中$\forall x\, F(x)$。从这些例子中推广，允许交替使用多个量词（和变量集合），就像在一阶逻辑中做的那样，可以得到新的复杂度类。例如，考虑满足$\exists x\, \forall y\, F(x,y)$的变量$x, y$中的公式$F$的集合，并使用它来定义一个所有问题（如果这让你想起了形式为“白方2步将死”的国际象棋谜题，你就有正确的直觉）都能有效归约到它的类（称为Σ_2）。类Π_2由满足$\forall x\, \exists y\, F(x,y)$的公式$F$类似定义。这些自然扩展到任何固定自然数$k$ (的Σ_k, Π_k类；co$\mathcal{NP} = \Pi_1$；以及当没有量词)时自然地扩展到$\mathcal{P} = \Sigma_0 = \Pi_0$类，具有明显的包含关系（$\Sigma_k \subseteq \Sigma_{k+1}$，$\Pi_k \subseteq \Pi_{k+1}$）。虽然在第一阶逻辑中，每个额外的量词严格增加描述能力是一个定理，但在计算复杂性（例如，$\Sigma_k \neq \Sigma_{k+1}$）中的类似情况只是一个猜想（扩展$\mathcal{P} \neq \mathcal{NP}$）。这些类的并集称为$Polynomial\ Hierarchy$，表示为$\mathcal{PH}$。其研究始于Stockmeyer[Sto76]。这个类包含许多自然问题，其确切复杂度未知。其中最有趣和迷人的是$MCSP$，即最小电路大小问题[1]，其状态在[AH17]中进行了综述。

- **Counting problems.** 修复一个 $\mathcal{NP}$ 问题。给定一个输入，找到它的 *number* 解决方案（证据）。许多枚举组合数学和统计物理学中的问题都属于这一类。在这里，计数问题的自然放松也是近似：计算一个接近实际计数的数字。这些问题的自然家园是一个称为 #$\mathcal{P}$ 的类别。这个类别中最自然的完整问题是 #SAT，它要求计算给定公式的满足赋值数量（更一般地，典型 $\mathcal{NP}$-完整分类问题的计数版本是 #$\mathcal{P}$-完整）。对于它来说，一个显著完整的问题是评估 $Permanent$ 多项式，[2] 或者等价地，计算给定二部图的完美匹配数量。因此，即使是简单分类问题的计数版本（例如，测试是否存在完美匹配）也可以

[1] The input to this problem is a Boolean circuit, and the problem is to determine if there exists a smaller circuit computing the same function. We will formally define circuits in Section 5.2.

[2] A sibling of the Determinant, which is discussed in Chapter 12.

#$\mathcal{P}$-完备。这一发现，类#$\mathcal{P}$的定义以及枚举问题的复杂性理论研究源于Valiant的论文[Val79a,Val79c]。Toda [Tod91] 的一个令人惊讶的基本结果有效地将（上述）量化问题减少到计数问题（用符号表示，$\mathcal{PH} \subseteq \mathcal{P}^{\#\mathcal{P}}$）。

- **Strategic problems.** 给定一个（完全信息，两人）博弈，找到一个给定玩家的最优策略。等价地，给定游戏中的一个位置，找到最佳走法。经济学和决策理论中的许多问题，以及玩好象棋，都属于这一类。这些问题的自然归属是使用多项式内存（但可能是指数时间）可解决的问题类$\mathcal{PSPACE}$。事实上，许多这样的游戏（适当扩展到任意大小的游戏族，以允许渐近性，并将走法数量限制为“棋盘大小”的多项式）对$\mathcal{PSPACE}$是完备的。这种将计算中的基本内存（或空间）资源以量词交替（即游戏策略）的形式进行表征，也源于[Sto76]，并且显然扩展了上述有界交替游戏（它定义了$\mathcal{PH}$）。多项式空间的一个主要、令人惊讶的理解是[Sha92]的结果$\mathcal{IP} = \mathcal{PSPACE}$。它将$\mathcal{PSPACE}$确立为所有具有有效*interactive proofs* (的问题的家园，这是“书面证明”的一个重要扩展，如第10.1节所述，由$\mathcal{NP}$)捕获。

- **Total $\mathcal{NP}$ functions.** 这些是搜索问题，试图找到存在（如局部最优解、不动点、纳什均衡）的对象，并由小证人证明。在许多这样的问题中，输入是一个指数级大的图（可能是有权重的，可能是有向的）。任务是找到一个具有某些简单属性的顶点，其存在由组合原理保证。例如，每个有向无环图都有一个汇点（因此任务是找到一个），或者每个无向图都有偶数个奇数度顶点（因此任务是，给定这样一个顶点，找到另一个）。在启动这项研究的研究论文中，帕帕季米特里欧 [Pap94] 定义了几个复杂性类，每个类都由这样一个原理捕获。这些类位于（与）$\mathcal{P}$和$\mathcal{NP}$相关的搜索问题之间。一个重要例子是$\mathcal{PLS}$类，对于多项式局部搜索，其完整问题是在加权有向图中找到一个局部最小值。另一个例子是$\mathcal{PPAD}$类，其自然完整问题是（给定函数的离散版本）计算一个不动点。在给定的两人游戏中计算纳什均衡显然属于这个类别，因为纳什定理（每个游戏都有一个这样的均衡）的证明简单地遵循布劳威尔的不动点定理。一个主要结果 [DGP09,CDT09] 证明了逆命题：证明找到纳什均衡是这个类的一个完整问题。这些问题及其复杂性在[BCE+95]中通过证明复杂性的框架进行了研究，我们将在第6章中讨论这个主题。

图10显示了这些类别之间的一些已知包含以及其中的一些问题。请注意，尽管SAT和$CLIQUE$是$\mathcal{NP}$-完备的，而*Perfect Matching*在$\mathcal{P}$中，它们的计数版本都在#$\mathcal{P}$中，而且确实，这三个都是此类（$Permanent$是完美匹配的计数问题）的完备问题.[4]

我将不在此处详细阐述这些重要问题和类别的家族。其中一些将在后续章节中提及，但我不将系统地发展它们的复杂性理论。请注意，高效归约和完备性的方法以与分类问题相同的方式阐明它们的大部分计算复杂性。

4.2 Between $\mathcal{P}$ and $\mathcal{NP}$

我们已经看到 $\mathcal{NP}$ 包含了大量的问题，但在难度方面，我们看到的几乎所有问题都落入两个等价类之一：$\mathcal{P}$，这些都可以高效解决，以及 $\mathcal{NP}$-完全。当然，如果 $\mathcal{P} = \mathcal{NP}$，这两个类是相同的。但假设 $\mathcal{P} \neq \mathcal{NP}$，还有其他什么吗？ Ladner [Lad75] 证明了以下结果。

[3]For example, via a program computing the neighbors of any given vertex.
[4]We oversimplified the figure a bit; technically, we only know that $\mathcal{PH} \subseteq \mathcal{P}^{\#\mathcal{P}}$, rather than $\mathcal{PH} \subseteq \#\mathcal{P}$.

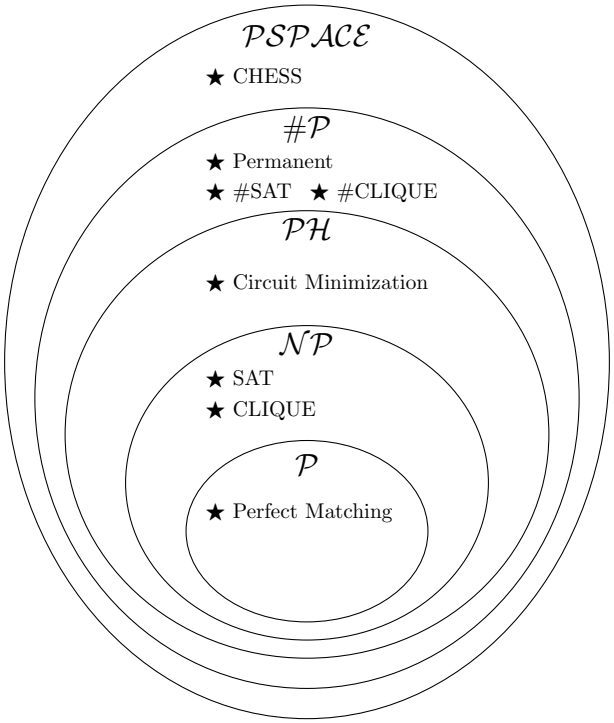


图10。在$\mathcal{P}$和$\mathcal{PSPACE}$之间。据我们所知，所有这些类别可能相等。

Theorem 4.1 [Lad75]. *If $\mathcal{P} \neq \mathcal{NP}$, then there are infinitely many levels of difficulty in $\mathcal{NP}$. More precisely, there are sets $C_1, C_2, \ldots$ in $\mathcal{NP}$ such that for all i, we have $C_i \leq C_{i+1}$, but $C_{i+1} \not\leq C_i$.*

因此，在 $\mathcal{P}$ 和 $\mathcal{NP}$-complete 之间有很多“暗物质”。但是，是否存在介于这些类别之间的 *natural* 决策[5]问题？我们只知道非常珍贵的少数几个候选人：以下列表中的那些（其中一些也在第 3.5 节中讨论过）以及少数其他问题。我们将依次讨论它们，在列出它们之后。

- **Integer factoring.** 给定一个整数，找出它的质因数（决策版本可能要求求第 i 位第 j 个质数的 i 位）。
- **Stochastic games.** 三位玩家，白方、黑方和自然，在一个顶点标记为玩家名称的有向图上移动一个标记。在每一步，标记可以被标记该顶点的玩家沿着从该顶点出发的边移动到另一个顶点。自然的移动是随机的，而白方和黑方则采取战略性的玩法。给定一个标记的图和标记的起始节点和目标节点，白方是否有策略能保证标记以概率 $\geq 1/2$ 达到目标节点?
- **Knot triviality.** 给定一个描述结的图（例如，见图1），这是平凡结吗?
- **Approximate shortest lattice vector.** 给定一个（基）格 L 在 $\mathbb{R}^n$ 中和一个整数 k，L 的最短向量是否有（欧几里得）长度至多 k，或者至少 kn? （保证这个最小长度不在 $[k, kn]$ 中。）
- **Graph isomorphism.** 给定两个图，它们是否同构? 也就是说，是否存在一个双射关系在

[5] We discussed search problems in this gap in Section 4.1.

它们的顶点保留边? [6]

- **Circuit minimization.** 这个问题描述中出现的概念将在第5章中形式化。直观上，它要求找到计算固定大小输入的函数的最快程序。更正式地说，给定一个布尔函数 f 的真值表和一个整数 s，是否存在一个大小不超过 s 的布尔电路来计算 f? 关于这个问题的“中间状态”的一些证据可以在 [AH17] 及其参考文献中找到。

目前我们无法排除找到解决这些问题的有效算法的可能性，因此其中一些实际上可能在$\mathcal{P}$中。但我们有很好的形式化理由相信它们不是$\mathcal{NP}$-完备的。这很有趣；我们已经看到，如果一个问题是$\mathcal{NP}$-完备的，那么它不是*easy*(，即在$\mathcal{P}$)中，如果我们相信$\mathcal{P} \neq \mathcal{NP}$。我们有什么迹象表明一个问题是一致地*not*，即它不是$\mathcal{NP}$-完备)? 好吧，例如，如果问题在$\mathcal{NP} \cap \text{co}\mathcal{NP}$中，并且我们相信$\mathcal{NP} \neq \text{co}\mathcal{NP}$，那么这个问题不能是$\mathcal{NP}$-完备的。在上面的两个论证中，复杂类的不太可能的崩溃（$\mathcal{P} = \mathcal{NP}$和$\mathcal{NP} = \text{co}\mathcal{NP}$）给我们（可能在不同置信水平上）关于特定问题复杂性的一个迹象。在没有关于它们复杂性的明确定理的情况下，这多少是令人满意的。特别是，我们现在可以更好地解释为什么上述问题不太可能是$\mathcal{NP}$-完备的。

前四个问题都在 $\mathcal{NP} \cap \text{co}\mathcal{NP}$ 中。对于分解问题（的决策问题），这是显而易见的。对于随机博弈，这个结果在 [Con93] 中得到证明。在 [AR05] 中解决了格问题。结问题很特殊：它是两个包含都高度非平凡的罕见例子。$\mathcal{NP}$ 的成员资格在 [HLP99] 中得到证明（并且在 [Lac15] 中再次以非常不同的方式证明）。$\mathcal{NP}$ 的成员资格首先在 [Kup14] 和 [7] 中在广义黎曼假设（GRH）的条件下得到证明，最近在 [Lac16] 中给出了一个不需要未证明假设的不同证明。

图同构，虽然在 $\mathcal{NP}$ 中未知是否在 co$\mathcal{NP}$ 中，因此我们不能使用相同的逻辑直接排除其可能的 $\mathcal{NP}$-完备性。然而，可以应用非常相似的逻辑。图非同构有不同类型的简短、高效的证明，称为 *interactive proof*，在第 10 章中讨论。使用这个，可以证明如果图同构是 $\mathcal{NP}$-完备的，它将导致在 4.1) 节中定义的多项式时间层次结构 $\mathcal{PH}$ (的惊人崩溃。当然，随着上面提到的最近关于图同构的准多项式时间算法 [Bab15]，我们有更多更好的理由相信它不能是 $\mathcal{NP}$-完备的。

最后问题，电路最小化（有几种变体），甚至比前四个问题更神秘。关于其可能容易（在$\mathcal{P}$中）和可能困难（$\mathcal{NP}$-完备）的“不可能”后果，已经发表了大量论文。关于这个主题的最新综述是[All17]。

找到其他类似的自然例子（或者更好的是，例子类别）将增强我们对差距 $\mathcal{NP} \setminus \mathcal{P}$ 的理解。考虑到上述例子，我们预计数学比工业更有可能是它们的来源。然而，对于某些大型自然问题类别，我们知道或相信它们 *must* 表现出这种二分性：该类别中的每个问题要么在 $\mathcal{P}$ 中，要么是 $\mathcal{NP}$-完备的。这些是约束满足问题类别，我们将在下一节中讨论。

4.3 Constraint satisfaction problems (CSPs)

本节包含，除了算法和复杂度的极其一般和核心猜想之外，还有数十年来坚持不懈的故事以及为克服这些问题所需的令人惊讶的数学来源。

4.3.1 Exact solvability and the dichotomy conjecture

大量自然问题可以表述为在变量集上满足大量约束。求解某些域上的线性方程组或多项式方程组，满足布尔公式，

[6] The recent breakthrough of Babai [Bab15] gives a quasipolynomial-time algorithm for this problem (namely of complexity roughly $\exp((\log n)^{O(1)})$), bringing it very close to $\mathcal{P}$. See also the exposition [HBD17] of this result.

[7] It may seem mysterious what the GRH has to do with knots, and I encourage you to look at the paper to find out.

图着色是许多例子中的一部分。在这里，我们将对 *local* 和 *uniform* 约束集合感兴趣。“局部性”意味着每个约束作用于一个常数的变量数。“均匀性”意味着所有约束都是同一类型。例如，在平面地图的3色着色中，变量对应于区域的目标颜色（每个可以取三个值之一，例如，红色、绿色、蓝色），每个相邻区域的每一对都规定一个约束：这两个区域的颜色必须不同。

让我们给出一个更正式的定义。固定局部性参数)、变量)的可能值字母表Σ (和定义满足约束的元组集合)的关系$R \subseteq \Sigma^k$ (，我们用CSP(k, Σ, R)表示以下计算问题。给定一个从n变量集合中提取的k-元组集合，是否存在从Σ^n变量集合中分配的变量赋值，使得它满足*all*约束（即分配给每个给定元组的值在关系R中）？这些都是只有一个关系的关系约束问题。更一般地，我们可以允许有多个关系而不是一个，我们将它们都称为“关系约束问题”。

例如，对于图的三色问题，存在一个（二元）关系（$a \neq b$）在三个颜色（形式上，这个关系包含三个对$R_{3col} = \{(1,2),(1,3),(2,3)\}$）的字母表$\{1, 2, 3\}$上。CSP($3, \{1, 2, 3\}, \mathrm{R}_{3\mathrm{col}}$)问题的一个实例是在$n$变量上可以取这三个值之一的约束集合（$x_i \neq x_j$）。约束中出现的对是此$n$-顶点图中的边。另一个例子，考虑布尔可满足性的$3 - SAT$问题。这里需要引入八个3元关系，一个用于否定或未否定单个子句中变量的八个文字组合（例如，（$a \vee b \vee c$），（$\bar{a} \vee b \vee c$））。此问题的一个实例是在n变量x_i上取布尔值$\{0, 1\}$的约束集合。也就是说，它是一个标准的$3 - SAT$实例。

Feder和Vardi [FV98] 提出的一个大胆猜想，称为 *Dichotomy conjecture*，断言对于这个庞大的问题集，不存在 Ladner定理第4.2节所述类型的中间复杂度级别——每个CSP要么属于 $\mathcal{P}$，要么是 $\mathcal{NP}$-完全的。

Conjecture 4.2 (二分猜想 [FV98])。每个CSP要么在 $\mathcal{P}$ 中，要么是 $\mathcal{NP}$-完全的。

这个猜想由Schaefer [Sch78] 在所有二进制字母上的CSP子类中得到了证明。他的证明实际上描述了哪些约束（即关系）会产生容易的CSP，哪些会产生困难的CSP。Bulatov和Jeavons在一般情况下的进一步工作[BJ01]试图通过定义每个特定CSP的关系的某种代数属性来寻找这种描述。这种基础关系的属性是具有非平凡 *polymorphisms*：这些是结合关系R中有限序列组的函数，以另一个关系中的序列R (。有关更多信息，请参阅[BKW17]。令人惊讶的是，非平凡多态的存在可以区分，例如，像*linear equations*和*disjunctions*这样的关系，使第一个CSP变得容易（在$\mathcal{P}$中）和第二个变得困难（$\mathcal{NP}$-完全）。请注意，这种描述需要开发容易情况下的元算法和用于证明困难性的元归约，因为它们同时为无限多个CSP推导出每种类型的结果。Bulatov [Bul06] 在三元字母上的关系上实施了这一代数计划，并在这种情况下也证明了猜想。

当这本书付印时，Rafey、Kinne和Feder [**?**] 以及Bulatov [Bul17] 和Zhuk [Zhu17] 宣布了完全二分猜想。我们将其陈述为一个定理，证明了上述猜想。为这个重要的算法结果所需发展的令人瞩目的 *algebraic* 理论，或许是一个关于高效算法（和约简）来源多样性的教训。

Theorem 4.3 (二分法定理 [Bul17, Zhu17])**.** *Every CSP is either in $\mathcal{P}$ or is $\mathcal{NP}$-complete.*

存在许多CSPs的扩展及其相关问题。在第11.2节中，我们将讨论*quantum Hamiltonians*，它们构成了上述“经典”CSPs的量子类比。它们在物理学中自然出现，并引发关于算法、完备性、证明、近似、二分法等问题和结果的自然计算复杂性。

返回经典CSP，对满足赋值数量（而不是存在一个）的*counting*有主要兴趣。这类问题在代数组合数学和统计学中至关重要

[8] To me polymorphisms are very reminiscent of the combinations of tuples of accepting computations to a new one in the *Fusion Method*, described in [Wig93] and in Section 5.2.4 on Natural Proofs of circuit lower bounds. But I know of no formal connection.

物理学。在其中一个广泛的理论中，可以证明强二分定理（现在是在 $\mathcal{P}$ 和 #$\mathcal{P}$-完备性之间）——参见蔡和陈的论文[CC12]及其书籍[CC17]中的结果和历史记载。我们将详细讨论解决CSPs *approximately* 的问题。

4.3.2 Approximate solvability and the unique games conjecture

一个关于CSPs的非常自然的问题是如何高效地找到好的*approximate*解，即满足许多给定约束的变量分配。例如，在上文中提到的图3-着色问题中，很容易满足2/3的边约束（注意，随机着色平均来说可以做到这一点）。[9]我们还知道，满足所有这些约束是$\mathcal{NP}$-难。最佳*approximation ratio*可达到的高效性是什么？例如，我们能获得99%吗?

巨大的数学领域优化研究这样的问题（不仅限于CSPs），寻求产生良好近似解的高效算法，对于最优解难以找到的问题。几十年来，人们不知道如何论证近似问题（例如，上述3-着色问题）的难度（例如，$\mathcal{NP}$-难度）。这种状况随着概率可验证证明（PCPs）的发明和第10.3节讨论的革命性PCP定理10.6发生了巨大变化。这个定理和随后的进展不仅使得证明许多问题的这种难度结果成为可能，甚至对于某些问题，还能够精确指出*precisely*有效近似能力的极限。例如，尽管满足1/2给定集合线性方程组的2模下的一个分数是$\mathcal{P}$ (找到这样一个算法！)，满足$1/2+\epsilon$分数（对于任何$\epsilon>0$）最终被证明是$\mathcal{NP}$-难度。

这个问题在复杂性上存在如此尖锐的过渡似乎非常令人惊讶。然而，一个类似的二分猜想表明，对于 *every* CSP，这确实如此。这个猜想的真实性似乎取决于以下引人注目的计算问题。它的$\mathcal{NP}$-完备性状态（是还是不是？），以及更一般地，它的精确计算复杂性，在过去十年中是这个领域最引人入胜和最重要的两个问题。这个问题是由Khot [Kho02] 提出的，他（出于某种原因）称之为*Unique Games*问题。我现在将解释这个问题，并简要说明为什么它如此核心。第10.3节将讨论这个问题的惊人起源。

Unique Games: 修复 $\epsilon>0$ 和整数 m。问题 UG(ϵ, m) 如下。输入是 n 变量 $x_1, x_2, \ldots, x_n$ 在 $\mathbb{Z}_m$ 上的线性方程组，每个方程有 *two* 个变量。如果存在一个分配满足方程的 $1-\epsilon$ 分之，算法必须回答“是”，如果没有任何分配满足超过 ϵ 分之的方程，则回答“否”（如果这两种情况都不成立，任何答案都是可接受的）。[10] 注意，每个独特游戏问题都是一个简单的 CSP（有 m^3 个关系，每个 $a, b, c \in \mathbb{Z}_m$ 一个，每个都是一个形式为 $a \cdot x_i + b \cdot x_j = c$ 的变量对的线性约束）。

在他的论文中，Khot提出了这个问题是$\mathcal{NP}$-完全的。它通常被称为独特游戏猜想（UGC）。

Conjecture 4.4 (UGC [Kho02]{v*37}. 对于每个 {v*38} 0，存在 {v*39} 使得 UG({v*40}) 是 {v*41}-难。

注意，UG问题可以被视为一个近似问题——为了解决这个问题，只需要将满足方程的最大数量近似到小于$(1-\epsilon)/\epsilon$的因子即可。这将区分“是”和“否”实例。Khot证明了改进某些已研究问题的某些近似算法至少与解决UG问题一样困难。自从他的论文发表以来，已经发现了更多这样的缩减。唯一游戏问题似乎是一种新的完全问题，捕捉了许多设置中有效近似的极限。当Raghavendra [Rag08] 证明，假设UGC，存在一个单一、简单的有效元算法（基于半定规划），对于每个约束满足问题，除非 $\mathcal{P}=\mathcal{NP}$，否则都能达到最佳可能的近似比时，这一点得到了强有力的证明。请注意，Raghavendra的结果可以表述为一个（条件）二分定理。

Theorem 4.5 [Rag08]. *Assume UGC. Then for every CSP, there is a constant ρ such that approximating it to within approximation ratio ρ is in $\mathcal{P}$, but approximating it to any better ratio $\rho+\epsilon$ is $\mathcal{NP}$-hard for every $\epsilon>0$.*

[9] Try finding an efficient deterministic algorithm that will produce such a 3-coloring.
[10] Such a problem is called a “promise problem,” where algorithms can err on instances not satisfying the promise.

可能公平地说，与本书中的大多数猜想不同，关于UGC的真实性没有共识。然而，UGC的研究结果出人意料地成为分析、几何、概率和其他领域的难题来源。特别是，Raghavendra和Steurer [RS10] 将其与图中的扩张联系起来，这是我们将在第8.7节和其他部分讨论的主题。关于这个问题、猜想及其联系，请参阅[Kho10, O'D14]。几年后，发生了以下情况，可能为UGC的真实性提供了更强有力的证据。

2-2 Games: Khot的原始论文[Kho02]关于UGC也包含一个相对较弱的猜想，对于约束为上述线性方程的CSP，除了它们有两个可能的值而不是一个（即具有$a \cdot x_i + b \cdot x_j \in \{c_1, c_2\}$的形式）。Khot猜想了这个问题的$\mathcal{NP}$-难性，并表明这甚至极大地提高了我们对许多自然近似问题的理解。一系列迅速发表的论文，以[KMS18]（描述了这一历史）为高潮，证明了2-2游戏问题确实是$\mathcal{NP}$-难！这些论文中的复杂证明需要分析所谓的*Grassmann graph*的扩展性质。

4.4 Average-case complexity

重要的是强调，我们在整个过程中采用的算法最坏情况分析（考虑每个长度的最坏输入的求解时间），当然不是唯一有趣的复杂度度量（也不是唯一的研究对象）。通常情况下，平均情况分析，关注“典型”算法的复杂度，研究起来更有趣。毕竟，在许多应用中，大多数时候解决一个难题可能就足够了。因此，在给定的自然输入分布*specific*下分析算法（例如，它们平均表现如何，或以高概率表现如何）是一个庞大且重要的领域，算法理论和计算复杂性理论一直认真对待。建模此类“典型”分布和为它们开发算法的一些一般方法包括算法的概率分析[Kar76]、半随机模型[FK01]、平滑分析[ST04b]、稳定实例[BL12,BBG13]、具有某些矩界限的输入分布[HS17, KS17]等。我们将在第20.7.2节中详细阐述这一努力以及理解启发式算法的挑战。

但事实是，在大多数实际应用中，通常对输入分布知之甚少，甚至一无所知。问题实例是由自然或人类生成的，它们的分布（甚至支持）很难确定。为了了解这一点，可以考虑分子生物学算法处理的基因组蛋白质集合，天体物理学算法处理的来自外太空的信号集合，Google处理的搜索请求集合，甚至工作数学家在思考问题时玩弄的数学结构集合（例如，第2章开头陈述的问题）。能否对*all*自然输入分布进行分类？应该如何构建一个成功的平均案例复杂性理论？所有$\mathcal{NP}$-完全问题在平均情况下是否都容易？我们应该如何在这个分布设置中正式定义容易和困难问题，以及我们如何比较分布问题的相对难度？

这些高度非平凡的难题首先由莱文[Lev86]解决，并在他与伊帕吉亚佐的后续工作中得到更好的理解[IL90]。让我们讨论莱文理论的一些主要定义及其微妙之处。读者有望感受到之前引入的分类、归约和完备性方法的力量，以及当将此方法应用于新环境时可能出现的困难。

由于输入分布事先未知，莱文将分类问题的概念推广到包含分布的 *distributional* 问题。这些问题具有 (C, D) 的形式，其中 $C \subseteq \mathbf{I}$ 定义了一个属性（或分类问题），与之前相同，而 D 是 $\mathbf{I}$ 上的概率分布，根据该分布选择输入。接下来要处理的是定义 *easy* 分布问题类，这是 $\mathcal{P}$ 类的分布类比，称为“dist$\mathcal{P}$”，即具有“平均”快速算法的那些问题。这个定义提出了一个重大挑战。回想一下，容易的最坏情况问题类 $\mathcal{P}$ 具有强大的鲁棒性属性（例如，它对模型或数据表示的多项式变化是不变的）。在分布设置中保证这种理想属性是困难的。采用平均多项式时间的明显定义——即当算法在给定分布下的期望运行时间随着输入长度的增长呈多项式增长时——是不行的。这个期望值可以

在多项式和指数之间变化，例如，在输入大小或每个输入的运行时间的二次变化下。莱文通过使用一种巧妙、非标准的概念（我们在这里省略）来克服这种困难，即算法在平均多项式时间内解决输入分布上的问题C的含义。然后，“简单”问题类dist$\mathcal{P}$在平均意义上定义为所有具有这种算法的分布问题(C, D)的集合。

接下来，一方面着眼于理论的实际应用，另一方面着眼于识别$complete$分布$\mathcal{NP}$问题（与其他问题一样困难），必须谨慎选择要考虑的允许输入分布的集合。不难看出，允许所有分布是毫无希望的。然而，人们希望理论包括所有实际可能发生的$reasonable$分布，如上面提到的自然和人工示例。正确的选择是可高效采样的分布。一个概率分布是$efficiently\ sampleable$，如果它是任何以独立无偏的硬币投掷为输入的有效算法的输出分布（大致上，是每个给定长度的序列上的均匀分布）。这是一个极其广泛的概率分布类别！实际上，假设自然不会执行计算上不可行的工作，这涵盖了算法将面临的all分布。一旦做出这个选择，$\mathcal{NP}$的分布对偶dist$\mathcal{NP}$就可以简单地定义为所有具有$C \in \mathcal{NP}$和D为可高效采样的分布的对(C, D)。

它还需要定义约简和完备性。这些是自然而然的；在最坏情况设置中使用的定义足够了，因为任何映射在实例 $f : \mathbf{I} \to \mathbf{I}$ 上自然扩展到分布上的映射。但是，是否存在任何完备问题，甚至更自然的问题？请注意，为了 (C, D) 成为完备问题，分布 D 必须“捕获”所有其他高效可采样的分布！这提出了巨大的技术挑战，Levin 通过证明某种分布版本的平面镶嵌问题是 dist$\mathcal{NP}$ 的完备问题而克服了这些挑战。自从 Levin 30 年前的原始论文以来，只有极少数其他合理自然完备问题被发现。展示一个真正自然完备问题仍然是一个挑战：可能是均匀分布下的 $\mathcal{NP}$-完备问题之一，或者是在自然吉布斯分布下的统计力学（如伊辛模型）中出现的问题之一。

显然，如果 $\mathcal{P} = \mathcal{NP}$，则 dist$\mathcal{P}$ = dist$\mathcal{NP}$。也许这个领域最突出的问题就是关于逆命题：自然的最坏情况难题是否意味着自然平均情况难题的存在？

Open Problem 4.6. $\mathcal{P} \neq \mathcal{NP}$是否意味着dist$\mathcal{P} \neq$ dis t$\mathcal{NP}$?

读者可以在Impagliazzo [Imp95b] 和Goldreich [Gol97] 的著作中找到更多关于这个迷人主题的细节。

4.5 One-way functions, trap-door functions, and cryptography

没有比密码学的故事更能展示计算复杂度思想和方法的强大力量，从而彻底改变我们所生活的世界的例子了。我在这里非常简要地讲述它，重点关注支撑它的特殊类型的“平均难”函数。我在本书的第18章中对此故事进行了更详细的阐述。有许多流行和技术文本阐述了密码学；我推荐Goldreich的书籍[Gol04]，以全面的技术发展其理论思想。

让我们首先阐述单向函数的动机，这些函数源于实际应用，即由Needham在20世纪60年代提出的一个密码方案（由Wilkes描述[Wil75，第91页]）。信不信由你，自那时以来访问控制并没有太大变化。一个典型的系统会要求用户提供两份信息，$login$和$password$。假设（在普遍性上损失不大）这两个都是某些长度n的序列，并且用户i的登录只是数字i。每个用户秘密地（随机或以任何其他方式）选择一个密码x_i。因此，如果你输入(i, x_i)（对于任何i），那么系统应该让你进入，否则不应该。主要问题是：系统应该如何存储其用户的密码？当然，它可以在受保护的系统文件中存储所有对(i, x_i)（当用户输入(i, z)时，检查$z = x_i$）。

但即使在20世纪60年代，黑客也闯入了系统，因此寻求更好的解决方案。纽曼提出了一种方法，可以完全避免隐藏密码文件的需要，如下所示。定义一个函数 $f : \mathbf{I} \to \mathbf{I}$。存储在密码文件中的信息将是 (i, y_i) 对，其中包含 $y_i = f(x_i)$。现在让我们看看

哪些属性 {v*} 应该具备才能使这个系统变得良好。首先，它必须在任何输入 x 上是 *easy to compute* f。毕竟，现在检查 (i, z) 是否合法需要系统检查 $f(z) = y_i$。然而，访问成对 (i, y_i) 的信息不应帮助任何人推断出 x_i 对于任何 i。特别是，f 必须是 *hard to invert*：给定 $f(x)$，如果存在，则获取 x（或任何其他 $f(x)$ 的前像应该是困难的）。这至少应该在均匀随机选择 x（的情况下以高概率成立，并且系统应该建议用户随机选择他们的密码）。

因此，我们得出了Diffie和Hellman在他们具有远见卓识、文笔优美的论文[DH76]中提出的*one-way*函数的定义，作为他们基于复杂性密码学革命性理论的基石。这是一个易于计算但平均而言难以求逆的函数。

Definition 4.7 (单向函数)**.** 一个函数 $f : \mathbf{I} \to \mathbf{I}$ 被称为 *one-way* 如果 $f \in \mathcal{P}$，但对于每一个有效算法 A，它在随机输入上计算 f 的任何前像的概率很小。也就是说，对于每一个（足够大的）n，

$$\Pr[f(A(f(x))) = f(x)] \leq 1/2,$$

在 n-位序列 x 的均匀分布上取概率。换句话说，算法 A 被随机 x 驱动，应该以高概率失败，无法产生 y 的任何逆。

我们的选择将1/2作为逆概率的上界是任意的；实际上，选择范围[exp($-n$), $1 - 1/n$]中的任何界限都会得到一个本质上等效的定义；这是通过重复*amplification*成功概率来实现的——一个我们将在随机性章节中遇到的想法，例如第7.1节。

让我们来探讨Diffie和Hellman的建议（实际上，他们将其归功于John Gill）——基于离散对数假设困难性的单向函数，模幂运算。

设 p 为一个素数，g 为 $\mathbb{Z}_p^*$ 的生成元，并定义 $\mathtt{ME}_{p,g} : \{1, 2, \ldots, p-1\} \to \{1, 2, \ldots, p-1\}$ 为 $\mathtt{ME}_{p,g}(x) = g^{x-1} \bmod p$，模 p（的模幂函数，并注意它实际上是一个排列）。注意，在每一个输入上计算 $\mathtt{ME}_{p,g}$ 都很容易（通过重复平方）。[11] 人们认为，对于 p 这样的素数，$p-1$ 有很少的因子（例如，$p - 1 = 2q$ 与素数 q），计算 $\mathtt{ME}_{p,g}$（的逆，即 *discrete logarithm* 模 p）是指数级困难的，[12] 即使平均来看也是如此（过去几十年中，许多努力寻找更好的算法都没有反驳这一信念）。可以将所有这些排列（以几种自然的方式）“粘合”成一个单一的排列 $\mathtt{ME} : \mathbf{I} \to \mathbf{I}$，即 *modular exponentiation* 排列，并猜想：

Conjecture 4.8. 模幂运算函数 ME 是一个单向函数。

这个猜想与我们已遇到的复杂度类有何关联？因为它要求平均难度，这意味着 dist$\mathcal{P} \neq$ dist$\mathcal{NP}$。事实上，任何单向函数的存在都将意味着这一点！此外，由于 ME 是一个排列，它意味着 $\mathcal{NP} \cap \text{co}\mathcal{NP} \neq \mathcal{P}$.[13] 事实上，任何单向排列都会意味着这个猜想。这些联系有望暗示计算复杂度分类系统在将不同问题的难度联系起来方面的力量，这些问题的复杂性未知，既提供又限制着各种类型困难问题的例子。

在Diffie和Hellman的论文之后，Rivest、Shamir和Adleman提出了他们的候选单射函数，模幂运算，该函数（间接地）基于整数分解的假设难度。它在密码学目的上相对于模指数运算具有重要的优势，我们将在定义它之后简要讨论。

设 p, q 为素数，$N = pq$，和 c 在模 $\phi(N) = (p-1)(q-1)$ 下可逆。定义 $\mathtt{MP}_{N,c} : \mathbb{Z}_N \to \mathbb{Z}_N$ 为 $\mathtt{MP}_{N,c}(x) = x^c \bmod N$。这同样是一个排列；如果 d 是 c 在模 $\phi(N)$ 下的逆，那么 $\mathtt{MP}_{N,d}(\mathtt{MP}_{N,c}(x)) = x$。再次，计算 $\mathtt{MP}_{N,c}$ 是容易的。人们相信在随机输入上，没有访问 N 的因子，反转它是指数级困难的。适当地将这些函数粘合在一起

[11] Namely, computing all powers $g^{2^i} \mod p$ with $i \leq \log p$.

[12] As a function of the input length, which is roughly $\log p$ bits.

[13] As these classes are defined for decision problems, proving this requires a decision version of ME. For example, given integers a, b, y, decide whether there is an x in the interval $[a, b]$ such that $\mathtt{ME}(x) = y$. As inverses exist and are unique, this classification problem is both in $\mathcal{NP}$ and co$\mathcal{NP}$. Any algorithm for it can invert ME via binary search.

模块化幂函数（实际上是一个排列）$\mathtt{MP} : \mathbf{I} \to \mathbf{I}$，我们有一个与ME的类似猜想：

Conjecture 4.9. 模块化幂函数 MP 是一个单向函数。

这个候选单射函数的显著优势在于，当有 p, q 的因子 N（时，将其 *trap-door*:[14] MP 反转变得容易。因为有了因子，可以有效地计算上述 c 的逆 d。这种性质是著名的 RSA *public-key cryptosystem* 的关键，RSA 是自发明以来大多数数字安全系统的基础。其工作方式极其简单。我（以及你、亚马逊；实际上任何人）都这样做。我随机选择两个大素数 p, q 并宣传（例如，在我的网站上）它们的乘积 N 和任何 c。如果你想给我发送一个秘密消息 x，你可以秘密地通过计算 $y = MP_{N,c}(x)$ 加密 x，然后通过任何公开渠道（例如，通过电子邮件）发送给我 y。根据猜想 4.9，没有因子 N 的人无法反转 y，但我当然可以解密你的消息，因为我有 p, q。这就是你在网上购物时信用卡号码被保护的方式；只要关于整数分解难度的计算复杂性假设是真实的，它就会得到保护！

让我详细说明上一段绝对引人注目的内容。陷阱门单射函数（我们不会正式定义）的可能存在并不仅仅是使在线购物和计算机安全（这些对社会的影响巨大）成为可能。它还允许任何两个事先不认识的人，在他人面前，建立并使用一种无人能理解的秘密语言！我们注意到，在“信息论”设置中，当计算是免费的，这样的壮举显然是不可能的。计算复杂性的基本前提——计算能力的限制和自然困难函数的存在——是绝对必要的。

这只是开始。陷阱门函数，诞生于解决秘密通信问题，最终解决了几乎所有可想象的密码学问题。在Goldwasser和Micali的奠基性论文[GM84]中为密码学奠定了形式基础之后，关于秘密和隐私的上述陈述可以数学化并得到证明，疯狂的1980年代爆发了关于其他“不可能”任务成为可能的论文。这些论文表明，基于陷阱门函数，诸如*contract signing, secret exchange, playing poker over the telephone, oblivious computation*和许多其他任务可以建立，最终 culminating in the general protocols of [Yao86,GMW87]。此外，即使是较弱的单向函数也被发现足以，并且实际上在计算上等同于诸如*private-key cryptosystems*、*pseudo-random generation* (等概念，更多内容在第7.3节中介绍)和*zero-knowledge*，更多内容在第10.2节中介绍*proofs* (。

难以在几页内公正地介绍密码学需求为复杂性理论模型和概念引入的新层次复杂性和微妙之处。我们将在第18章中更详细地讨论这些问题和结果，而在这里的讨论将相对简短。一方面，大多数问题涉及两个或多个参与者，引入了交互式 *protocols* 而不是单方 *algorithms*。对手和对抗性思维的重要性远远超出了最坏情况分析（在这种情况下，对手所能做的就是选择不良输入）到几乎任意的滥用，密码协议必须抵抗那些不遵循它们的参与者。为了在效率之外保留诸如 *knowledge* 这样的属性，需要在密码协议和原语之间进行缩减，并允许以新的方式操纵算法，远远超出了将它们应用于给定输入（例如，从任意状态反复回滚它们）的范围。这以许多方式丰富了计算复杂性。我们将在第7章中看到的一个巨大影响是对 *randomness* 的新理解，这是一个被许多学科研究数千年的概念。

Difie和Hellman在他们关于计算复杂性和$\mathcal{P}, \mathcal{NP}$的定义以及$\mathcal{NP}$-完备性的论文发表后不久。在这些早期，人们对$\mathcal{P} \neq \mathcal{NP}$很快就能被证明抱有乐观态度，并且他们有充分的理由期待函数的平均难度和单向性能够无条件地被证明。正如我们所知，这并没有成为现实，我们只能满足于*candidate*单向和陷阱门函数，并寻找其他方法来支持他们假设的难度。

[14] A notion sometimes signifying private or secret access.

[15] This notion was already suggested in [DH76], which also explained how it can be indirectly obtained (via a *key exchange* protocol) from the one-way function ME. Later, El Gamal [Elg85] showed how to obtain a public-key cryptosystem directly from ME.

对于最坏情况下的复杂性，大量具有实际重要性的$\mathcal{NP}$-完全问题的存在，以及数十年来在尝试高效解决这些问题上投入的巨大（且独立）努力，使人们有理由相信它们是困难的。对于单向函数，莱文[Lev87]构造了一个 *complete* 单向函数，即如果存在单向函数，则该函数是单向的。然而，这并不十分自然，没有人会梦想在实际密码系统中使用它。

因此，世界要继续享受密码学的益处，所依赖的是单个问题的假设难度，例如离散对数和整数分解。当然，由于它们在几乎所有计算机安全系统中都得到应用，因此（无论是好人还是坏人）都投入了巨大的努力来寻找它们的快速（或更快）算法，但到目前为止，还没有找到合理有效的算法。尝试提出替代方案也占据了密码学家的注意力。对于单向函数，我们实际上有很多候选者（实际上，几乎任何计算机程序都可能计算一个单向函数，尽管不清楚如何证明从输出中获取输入是困难的）。相比之下，对于基于陷阱门函数（公钥加密的基础），除了整数分解之外，我们几乎没有其他例子。一个突出的候选者，与上述数论问题非常不同，源于一系列作品[Ajt96, AD97, PW11]。在这里，安全性基于在随机格中找到短向量的难度，我们在第13.8节中讨论了这个问题。拥有更多陷阱门函数的候选问题是至关重要的！在第11章关于量子计算的讨论中，我们将看到离散对数和整数分解问题都有快速的 *quantum* 算法。因此，如果量子计算机真的被建造出来，当前的安全系统将变得过时，而对于这些格问题，目前（尚）没有已知的有效量子算法。

但我们所知，可能存在一个快速的经典算法来分解（并且一些数论学家坚信这一点）。在将密码学建立在稳固基础上时，可能最重要的任务是肯定回答以下问题。

Open Problem 4.10. $\mathcal{P} \neq \mathcal{NP}$（甚至dist$\mathcal{P} \neq$ dist$\mathcal{NP}$）是否意味着存在单向函数（甚至陷门函数）的存在？

证明难度（及其困难）是下一章的主题。

5 Lower bounds, Boolean circuits, and attacks on $\mathcal{P}$ vs. $\mathcal{NP}$

为了证明 $\mathcal{P} \neq \mathcal{NP}$，我们必须表明对于给定的问题，不存在有效的算法。这类结果被称为从下限限制问题的计算复杂度 *lower bound*（。在过去几十年中，出现了几种强大的证明下界的技术。它们适用于两种（非常不同的）设置。我现在描述这两种，并解释为什么它们似乎不足以证明 $\mathcal{P} \neq \mathcal{NP}$。我只简要提及第一种，对角化，而集中讨论第二种，布尔电路。

布尔电路被视为一种独立的计算模型进行研究，而不仅仅是上下文中的下界。这种所谓的非均匀电路模型与通常的均匀算法模型（例如图灵机）之间的联系并不完全清楚。对下界尝试中的电路的主要关注点是每个电路都是一个有限对象，并且可以希望（并在有限情况下成功，正如我们将看到的）使用组合方法来分析它们。

5.1 Diagonalization and relativization

对角化技术可以追溯到康托及其关于实数基数大于整数基数的论证。对角化被哥德尔用于其不完备性定理，以及图灵用于其不可判定性结果（读者可能希望回忆）。然后它被精炼以证明可计算函数的计算复杂度下界。该领域的一个典型定理是哈特曼尼斯-斯特恩斯定理*time-hierarchy* [HS65]，它本质上表明更多的时间可以带来更多的计算能力。例如，有些函数可以在时间n^3内计算，但不能在时间n^2内计算。这类论证（缩小图灵不可判定性证明）的核心是存在一个“通用算法”，它可以以效率损失很小的方式模拟其他任何算法。因此，一个n^3时间机器可以同时针对*all* n^2时间算法进行对角化，并计算一个与每个算法在某些输入上不一致的函数。对角化的更复杂应用产生了对图灵机其他重要下界（例如，参见[PPST83, For00]）。

这样的论点能否用来将 $\mathcal{P}$ 与 $\mathcal{NP}$ 分离？这取决于我们所说的“这样的论点”是什么意思。Baker、Gill 和 Solovay 的重要论文 [BGS75] 提出了这种证明技术的形式化，并表明通过这种形式化，无法获得这样的分离。这是旨在解释常见证明技术局限性的论文系列中的第一篇。他们的论点有两个部分。首先，他们指出许多关于复杂度结果的类似对角化证明的共同特征，称为 *relativization*。如果一个证明在修改所有涉及的算法（模拟机和模拟机）时仍然有效（即赋予它们解决任何固定（但任意）问题实例的自由能力），那么这个证明是相对化的。在常见术语中，这些机器都可以自由访问一个“预言机”，该预言机回答关于 S.[1] 成员的提问。在上面的例子中，一个具有访问 S 的 n^3-时间通用机可以模拟每个具有类似访问权限的 n^2-时间机器，并且也可以像原始证明那样对其进行对角化。[BGS75] 论文的第二部分表明，相对化论点不足以解决 $\mathcal{P}$ 与 $\mathcal{NP}$ 的问题。为了证明这一点，这些作者定义了两个不同的预言机，例如 S' 和 S''，它们的存在对 $\mathcal{P}$ 与 $\mathcal{NP}$ 问题的答案产生了相反的结果。也就是说，通过访问 S'，$\mathcal{NP}$-机器的“猜测”能力神奇地消失了，产生了“$\mathcal{P} = \mathcal{NP}$”，而通过访问 S''，猜测的能力被证明是指数级更强的，产生了“$\mathcal{P} \neq \mathcal{NP}$”。Fortnow [For94] 进一步讨论了相对化。

即使今天，几十年后，大多数复杂性结果都相对化。证明非相对化结果的主要方法源于第10.1节中讨论的*arithmetization*技术。这被用来证明一些相对化下界（例如，[Vin04, San09]），它们确实*not*相对化。为了限制这种技术的力量，Aaronson和Wigderson [AW09] 定义了*algebrization*，这是相对化的一个推广，它结合了使用算术化的证明。他们表明，代数化证明仍然太弱，无法解决$\mathcal{P}$与$\mathcal{NP}$问题，以及其他复杂性挑战。

[1] S can be arbitrarily hard, even undecidable.

5.2 Boolean circuits

布尔电路是另一种基本的计算模型，我们现在来探讨它。关于这个主题的优秀文本是Jukna的[Juk12]。

一个 *Boolean circuit* 可以被视为算法（软件）的硬件模拟。确实，它抽象了真实计算机内部的集成电路和许多物理控制设备。对电路的布尔输入进行计算是通过应用一系列布尔运算（称为 *gates*）来计算输出（s）。在这里，我们将考虑最常用的 *universal* 闸极集（有时称为 *de Morgan* 基础），$\{\wedge, \vee, \neg\}$：逻辑与（合取）、或（析取）和非（否定），分别。我们假设在这里 $\wedge, \vee$ 都应用于两个参数。请注意，尽管算法可以处理任何长度的输入，但电路只能处理一个输入长度（它拥有的输入“线”数量）。图11说明了在4位上计算奇偶函数的计算；计算从输入（在底部）到输出（在顶部）进行。

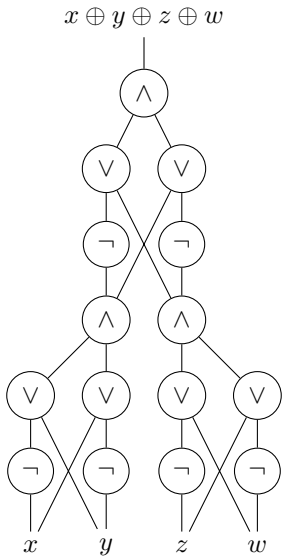


图11. 一个计算4位奇偶校验的电路。

一个电路通常表示为一个（有向、无环）图，其中门分配到其内部顶点。请注意，在布尔逻辑中常用的 *Boolean formula* 只是一个布尔电路，其图结构为树。

回忆一下，$\mathbf{I}$ 表示所有二进制序列的集合，而 $\mathbf{I}_k$ 是长度恰好为 k 的序列的集合。如果一个电路有 n 个输入和 m 个输出，那么它显然计算一个有限函数 $g\colon\ \mathbf{I}_n \to \mathbf{I}_m$。电路的效率通过其 *size* 来衡量，这是算法中时间的类比。

Definition 5.1 (电路大小)**.** 对于一个有限函数 g，用 $\mathsf{S}(g)$ 表示计算 g 的最小布尔电路的大小。

更一般地，对于 $f\colon\ \mathbf{I} \to \mathbf{I}$，在 f_n 的限制下，f 对大小为 n 的输入的约束，我们定义 $\mathsf{S}(f)$ 为从 n 到 $\mathsf{S}(f_n)$ 的映射。

我们关注渐近行为，因此将函数 $f\colon \mathbf{I} \to \mathbf{I}$ 视为一个有限函数序列 $f = \{f_n\}$，其中 f_n 是 n 输入位上的函数，即 f 对大小为 n 的输入的限制。我们将研究 $\mathsf{S}(f_n)$ 的复杂性，将其作为 n 的函数渐近地研究，并用 $\mathsf{S}(f)$ 表示。例如，设 PAR 为奇偶校验函数，计算二进制字符串中 1 的数量是偶数还是奇数。那么 PAR_n 是其 n 位输入的限制，并且不难验证 [2]。

[2]Namely, $PAR_n(x_1, x_2, \ldots, x_n) = x_1 \oplus x_2 \oplus \cdots \oplus x_n$.

电路族可以有效地模拟算法。证明一个算法（例如，图灵机）对于函数f在时间T内运行会产生一个大小为f_n的函数$O((T(n))^2)$的电路族是非常直接的。计算f_n的电路c_n模拟了给定图灵机在长度-n的输入上的运行。由于这个无限电路族$\{c_n\}$来自单个算法，因此它被称为*uniform*族。忽略均匀性，我们得到了算法类$\mathcal{P}$的电路模拟。

Definition 5.2 (该类 $\mathcal{P}$/poly)**.** 令 $\mathcal{P}$/poly 表示由一族多项式大小电路可计算的所有函数的集合，即所有函数 $f : \mathbf{I} \to \mathbf{I}$，使得 $\mathsf{S}(f)$ 与 n 的增长是多项式的。

上述模拟证明了以下定理。

Theorem 5.3. $\mathcal{P} \subseteq \mathcal{P}$/多*.*

作为这个简单模拟的结果，电路的下界意味着算法的下界，因此我们可以尝试通过电路（完全放弃均匀性）来攻击 $\mathcal{P}$ 与 $\mathcal{NP}$ 的问题。本节指导的猜想，比猜想3.6要强得多，如下所示。

Conjecture 5.4. $\mathcal{NP} \not\subseteq \mathcal{P}$/多边形。

这是一个合理的猜想吗？如上所述，$\mathcal{P} \subseteq \mathcal{P}$/多项式。然而，这个陈述的逆命题却完全失败！存在*undecidable*个函数f (，无论它们的运行时间)如何，都无法由图灵机计算，它们具有线性大小的电路.[3]这种小电路解决不可解问题的非凡能力来自于放弃了一致性假设；即不同输入长度的电路没有共享的描述。事实上，电路模型有时被称为“非均匀的。”

如果小型电路可以计算不可判定的函数，这似乎使得证明超多项式电路下界比证明 $\mathcal{P} \neq \mathcal{NP}$ 要困难得多。然而，有一种强烈的观点认为，非均匀性提供的额外能力对于 $\mathcal{NP}$ 中的问题是不相关的。这一观点得到了 Karp 和 Lipton [KL82] 的定理的支持，该定理证明了（非均匀）假设 $\mathcal{NP} \subseteq \mathcal{P}$/poly 导致了（均匀）复杂度类捕获 *quantified problems* (的一个令人惊讶的“崩溃”，这种崩溃类似于，但比 $\mathcal{NP} = \text{co}\mathcal{NP}$ 弱。

Theorem 5.5 [KL82]. *If* $\mathcal{NP} \subseteq \mathcal{P}$/多*, then* $\Pi_2 = \Sigma_2$ *(and hence,* $\mathcal{PH} = \Sigma_2$*).*

仍然，在寻求下界时，为什么用可能更强大的电路族来替换图灵机？希望是专注于一个 *finite* 模型将允许使用组合技术来分析有效算法的强大和局限性。这种希望已经在研究受限制的电路类（例如，参见第5.2.3节）中实现。

5.2.1 Basic results and questions

我已经提到了关于布尔电路的几个基本事实，特别是它们可以有效地模拟图灵机的事实。下一个基本事实是*most Boolean functions require exponential- size circuits*。

这是由于函数数量和小电路数量之间的差距。固定输入位数为n。在n位上可能的功能数量正好是2^{2^n}。然而，通过粗略估计该大小图的数量以及每个节点可能的选择门)，大小为s (的不同电路的上界大约是2^{s^2}。由于每个电路计算一个函数，我们必须有$s > 2^{n/3}$对于*most*个函数.[4]这被称为*counting argument*，最初归功于香农[Sha49b]。

Theorem 5.6 [Sha49b]. *For almost every function* $f : \mathbf{I}_n \to \{0, 1\}$*,* $\mathsf{S}(f) \geq 2^{n/3}$*.*

[3] An example is $f = \{f_n\}$, where f_n is a constant function, whose output is 1 or 0, depending on whether n encodes a solvable Diophantine equation or not, respectively.

[4] In many cases, I will deliberately give weaker bounds than the best possible when it allows simpler calculations and crude estimates.

因此，对于电路（以及图灵机）的困难函数很多。然而，由于上述困难是通过计数论证证明的，它并没有提供指定一个困难函数的方法。我们将在第6章回到这个问题的非构造性本质。到目前为止，我们无法证明任何*explicit*函数f (的这种困难，例如对于像SAT)这样的$\mathcal{NP}$-完备函数，尽管人们认为这是真的。这基本上意味着在解决SAT时，无法在蛮力穷举搜索上实现显著的时间节省。

Conjecture 5.7. $\mathsf{S}(SAT) = 2\Omega(n)$.

它并不令人惊讶，我们无法证明这个猜想，因为它比猜想3.6要强得多。[5]但我们无法建立下界的情况要糟糕得多——对于*any*显式函数，没有任何*nontrivial*下界是已知的。请注意，对于任何在n位（这取决于所有输入）上的函数f，我们只需读取输入，就 trivially 必须有 $\mathsf{S}(f) \geq n$。电路复杂性的主要未解决问题是超越这个平凡的界限，对于自然问题（例如，在$\mathcal{NP}$）——经过60多年的深入研究，我们仍然无法解决这个问题。

Open Problem 5.8. 找到一个显式函数 $f : \mathbf{I}_n \to \mathbf{I}_n$，使得 $\mathsf{S}(f) \neq O(n)$。

一个特别基本的特殊情况是这个问题，即加法是否比乘法更容易执行。分别用 ADD 和 $MULT$ 表示一对整数（以二进制表示）上的加法和乘法函数。对于加法，我们有一个最优的上界；即 $\mathsf{S}(ADD) = O(n)$。对于乘法，标准（小学）二次时间算法通过 Schönhage 和 Strassen [SS71]（通过离散傅里叶变换）得到了略微超线性的改进，得到 $\mathsf{S}(MULT) = O(n \log n \log\log n)$。已知最好的算法是 Fürer [Für09] 提出的，但它仍然比 $n \log n$ 慢。现在的问题是 *whether there exist linear-size circuits for multiplication.*。用符号表示，是 $\mathsf{S}(MULT) = O(n)$?

无法证明任何非平凡的下界，我们现在转向受限模型。在为自然受限电路类证明强下界的技术开发方面取得了一些显著的成果。我们详细讨论了两种这样的模型：首先是公式，然后是单调电路。

5.2.2 Boolean formulas

公式在数学中普遍存在，主要使用算术门如 $+$ 和 $\times$ (，算术计算在第 12 章讨论。我们在此关注布尔公式，这些在逻辑中是标准的，具有相同的德摩根连接词集 $\{\wedge, \vee, \neg\}$ (，例如，$(x \vee \neg y) \wedge z$)。一个公式可以被视为一个具有树结构的电路。一个计算 4 位奇偶函数的布尔公式的示例显示在图 12 中。我们用公式中变量的出现次数来表示公式的大小，即表示它的树中的叶子数（这，除以 2 的因子外，与门的数量相同）。让我们定义（必然是单比特输出）布尔函数的公式大小。

Definition 5.9 (公式大小)**.** 对于一个有限函数 $g : \mathbf{I}_n \to \{0, 1\}$，用 $\mathsf{L}(g)$ 表示计算 g 的最小布尔公式的尺寸。对于 $f : \mathbf{I} \to \{0, 1\}$，其中 f_n 是 f 对尺寸为 n 的输入的限制，我们定义（如上所述的电路）$\mathsf{L}(f)$ 为从 n 到 $\mathsf{L}(f_n)$ 的映射。

一个公式就像电路一样是一个通用的计算模型，因为每个布尔函数都可以通过布尔公式来计算。然而，正如我们马上将要看到的，公式是一个更弱的计算模型，所以我们可能希望为它证明更好的下界。事实上，这已经发生在本质上最简单的函数上，即上面讨论的 *parity* 函数 PAR。我们提到了 $\mathsf{S}(PAR) = O(n)$，并且很容易证明（请尝试）$\mathsf{L}(PAR) = O(n^2)$。电路复杂性的最早结果之一是奇偶性的下界。它们使用了非常不同、重要的下界技术，具有广泛的应用性。让我们依次讨论这两个。

Subbotovskaya [Sub61] 证明了 $\mathsf{L}(PAR) = \Omega(n^{1.5})$，开创了 *random restriction* 方法。这种证明技术是如何展示任何奇偶性公式都必须很大（并且更一般地，

[5]There is great value in making strong conjectures! We recommend finding out more about the related exponential time hypothesis (ETH) and its variants in the original papers [IPZ01, IP01], and in some of the recent applications (e.g. in the survey [LMS11]). This conjecture leads to a more "fine-grained" complexity theory than presented here, which in particular can distinguish different polynomial running times (see e.g. the survey [Wil18]).

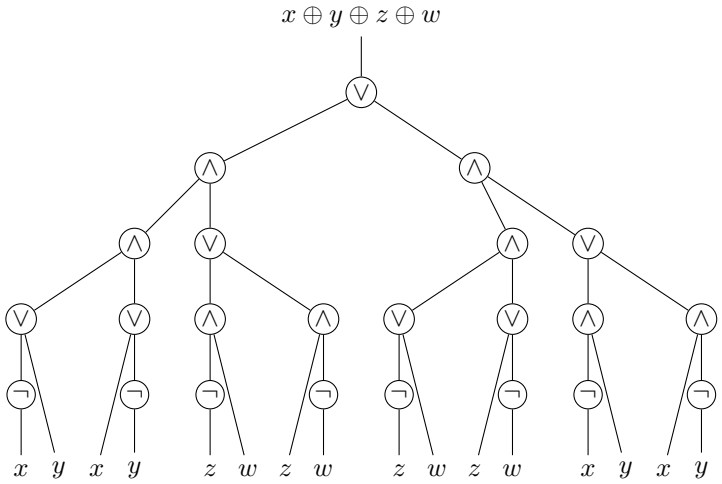


图12. 一个计算4位奇偶性的公式。

那任何一类计算 {v*} 的函数 f 必须很大吗？首先观察，将一些输入位固定为某些固定常量会导致在“硬编码”这些值后得到的计算 C'（更简单，这是在剩余变量上计算限制函数 f'。现在的想法是，如果计算模型“弱”且函数“复杂”，那么巧妙地选择要固定的变量及其值，将使 C' “平凡”而 f' “非平凡”，从而产生一个专利矛盾。[6] 当然，C 并未给出：我们必须排除任何小的计算，因此这种选择并不容易。Subbotovskaya 的想法是简单地随机选择这样的输入限制！在这里，这样的限制以比函数快得多的速度发生。因此，如果 C 太小，C' 就变成了一个常数函数，而 f' 就变成了剩余未固定变量的奇偶函数，导致下界。我们很快将回到这种通过随机限制进行缩小的想法。

十年后，Khrapchenko [Khr71] 将奇偶性下界改进为紧的 $\mathsf{L}(PAR) = \Omega(n^2)$。他使用对公式结构的自然归纳法给出了一个关于该函数 *sensitivity* 的一般下界。[8] Karchmer 和 Wigderson [KW90] 提供了该下界的另一种，*information-theoretic* 证明。他们的 *communication complexity* 方法表明，给定函数的公式可以被视为相关问题的“通信协议”，因此可以在这种信息论设置中证明下界（和上界）。这种技术有进一步的应用，我们将在下面再次遇到；它在第15章正式化通信复杂度模型后在15.2.3节中解释。

一个由安德烈夫[And87]提出的新想法，被哈斯泰德[Hås98]推向极限（使用对随机限制下公式收缩的紧密分析），给出了已知的最佳近似立方间隙。在计算复杂性不同子领域之间神秘联系的一个另类例子中，塔尔[Tal14]给出了一个不同的证明（具有略微更好的界限），令人惊讶地涉及到了一个*quantum argument*。更确切地说，它使用量子算法对任意布尔公式的一般模拟结果（我们在第11章讨论量子计算）。

Theorem 5.10 [Hås98, Tal14]. *There is a Boolean function f with $\mathsf{S}(f) = O(n)$ and $\mathsf{L}(f) = \Omega(n^{3-o(1)})$.*

实际差距被认为呈指数级，[9] 证明超多项式差距是一个重大挑战

[6] The words “trivial” and “nontrivial” have different meanings in different applications.

[7] Which was later rediscovered independently in [FSS84, Ajt83] in the context of *constant-depth* circuits, and then used throughout complexity theory. See [Bea94] for applications in circuit and proof complexity.

[8] Sensitivity is a key parameter (actually, a family of parameters) of Boolean functions, useful, for example, when viewed as voting schemes. See more in Section 13.7.

[9] Contrast this with the situation in *arithmetic* complexity to be discussed in Chapter 12, where formulas and circuits are much closer in power.

电路复杂性。

Conjecture 5.11. 存在一个布尔函数 f，具有 $\mathsf{S}(f) = O(n)$ 和 $\mathsf{L}(f) \neq n^{O(1)}$。

这个猜想（我们不会正式提出）的一个直观含义是断言存在具有快速顺序算法但没有快速并行算法的问题。

我们现在描述一种证明这个重要猜想的方法，该方法归功于Karchmer、Raz和Wigderson [KRW95]。它要求我们理解公式大小在函数*composition*的自然运算下的行为。

Definition 5.12. 两个布尔函数 $g \circ f : \{0,1\}^n \to \{0,1\}$ 和 $g : \{0,1\}^m \to \{0,1\}$ 的组成 mn 输入位，视为 m 向量 $x^i \in \{0,1\}^n$，并由 $g \circ f(x^1, x^2, \ldots, x^m) = g(f(x^1), f(x^2), \ldots, f(x^m))$ 定义。

最明显的计算这种组成的公式给出 $\mathsf{L}(g \circ f) \leq \mathsf{L}(g) \cdot \mathsf{L}(f)$。KRW 假设 [KRW95] 认为没有“显著更好”的方法来做这件事。[10]

Conjecture 5.13. 对于每个 f, g，$\mathsf{L}(g \circ f) \geq \alpha \cdot \mathsf{L}(g) \cdot \mathsf{L}(f)$ 对于某个绝对常数 $\alpha > 0$。

他们进一步展示了这一猜想（甚至更弱的猜想）如何蕴含猜想5.11，并且他们也概述了证明它的计划。关于该计划的现状和历史进展，请参阅[GMWW14,DM16]。有趣的是，三次下界，定理5.10，可以被视为对猜想5.13的一种非常受限形式的证明。实际上，[DM16]通过通信复杂度方法提供了定理5.10的一个替代证明。

让我们以通信复杂度方法提出的另一种方法来结束本节，以证明猜想5.11。我将非正式地陈述它，并再次将读者引荐到第15.2.3节。考虑以下任务。我在你耳边低声说一个n-位素数x，并在你朋友耳边低声说一个n-位合数y。你的目标是同意任何数字$z < 10n$，使得$x \neq y \pmod z$。证明这个目标不能通过使用$O(\log n)$位的通信来实现，你就已经证明了猜想5.11！

5.2.3 Monotone circuits and formulas

许多自然函数在自然意义上是 *monotone*。这里有一个例子，来自我们的 $\mathcal{NP}$-完备问题列表。设 *CLIQUE* 为一个函数，给定一个在 n 个顶点上的图（比如说，通过它的邻接矩阵），如果它包含一个大小为（比如说）$\sqrt{n}$ (的 *complete* 子图，则输出 1，即某些大小-$\sqrt{n}$ 子集中的所有顶点对通过边) 连接。这个函数是单调的，即在添加边不能破坏任何完全图的意义上。更一般地，一个布尔函数是单调的，如果“增加”输入（将输入位从 0 翻转到 1）不能“减少”函数值（使其从 1 翻转到 0）。

自然限制电路的方法是通过从门集合中去除否定，即只允许门 $\{\wedge, \vee\}$。由此产生的电路称为 *monotone circuits*，很容易看出它们可以计算每个单调函数。

一个类似于我们用于通用电路的计数论证表明，大多数单调函数需要指数级大小的单调电路。然而，证明一个明确单调函数的超多项式下界已经开放了40多年，直到Razborov发明了*approximation method*[Raz85a]。

Theorem 5.14 [Raz85a], [AB87]。*CLIQUE requires exponential-size monotone circuits.*

非常粗略地说，近似方法将（假定的）单调电路中的每个$\{\wedge, \vee\}$门替换为其他精心选择的（且难以描述的）*approximating*门，$\{\tilde{\wedge}, \tilde{\vee}\}$，分别。这种选择满足两个关键属性，这两个属性结合起来可以轻易排除*CLIQUE*的小电路：

[10] Indeed, the slack in this inequality can be super-constant, and it is interesting even if this holds for most functions f (or for most g).

(1) 通过用其近似器替换一个特定的门，只能影响电路在非常 *few* (在某些自然但非平凡的计数度量) 输入上的输出。因此，在一个只有几个门的简单电路中，即使将所有门都替换为它们的近似器，结果得到的电路在大多数输入上仍然表现得像原始电路。

(2) 然而，由近似门组成的 *every* 电路（无论大小）产生的函数在 *many* 输入上与 $CLIQUE$ 不一致.[11]

一种自然的方法是将近似方法描述为对电路上的 *progress measure* (或 *potential function*) 的描述，其中每个门只贡献很少。如果一个函数在这个度量上很昂贵，那么为它设计的任何电路都必须很大。自然地，我们可以根据它们在电路中的顺序，以 *dynamic* 的方式看待门对这种度量的微小总贡献。然而，有一种 *static* 的观点（被称为 *fusion method*），它实际上更好地捕捉了证明下界的方式，这在 [Wig93] 中被阐述.[12]

$CLIQUE$ 函数众所周知是 $\mathcal{NP}$-完备的，因此自然会想知道小单调电路是否足以在 $\mathcal{P}$ 中表示单调函数。然而，Razborov [Raz85b] 也使用了近似方法来证明计算 *perfect matching* 问题（该问题是单调的，并且位于 $\mathcal{P}$ 中）的单调电路的 $n^{\Omega(\log n)}$ 大小下界：给定一个图，能否将顶点配对，使得每对顶点都通过一条边相连（见图 3 中的插图）？

Theorem 5.15 [Raz85b]. *Perfect matching requires super-polynomial size monotone circuits.*

有趣的是，对于这个问题，没有已知关于单调电路的指数下界。Raz和Wigderson [RW92] 使用了通信复杂度技术（见下文）来证明完美匹配需要指数大小的单调公式。

Theorem 5.16 [RW92]. *Perfect matching requires exponential size monotone formulas.*

Tardos [Tar87] 最终证明了在计算（单调、阈值版本的）Lovász' *Theta function* Θ问题中，单调与非单调电路之间存在指数差距。此问题通过半定规划属于$\mathcal{P}$，并且下界使用的是定理5.14比所述的更强：即使只有输入图G要么是$k+1$-团（在这种情况下，$\Theta(G) \geq k+1$）要么是完全k-分割图（在这种情况下，$\Theta(G) \leq k$），它仍然成立。

电路与公式在单调情况下的相对功率也相当清楚。第一次分离是由Karchmer和Wigderson [KW90] 为图连通性证明的，这是一个具有简单单调、多项式大小电路的函数，但他们表明需要 $n^{\Omega(\log n)}$ 大小的单调公式，这是一个紧界.[13]

Theorem 5.17 [KW90]. *Undirected graph connectivity (which has monotone polynomial-size circuits) requires super-polynomial-size monotone formulas.*

定理5.16和5.17中的单调公式下界是使用上述提到的*communication complexity*方法（并在第15.2.3节中解释）证明的。

通信复杂度方法被Raz和McKenzie [RM99]推广，在单调电路和公式之间提供了更精细的分离。他们的方法（在文献中称为*lifting*或*pattern matrix*方法）在一系列最近的工作中得到了极大的增强和更好的理解，这使得它可以应用于其他单调模型，并以更简单的方式获得更强的结果——参见[RPRC16,PR17]及其历史讨论。

我们以一些简短的单调公式 *can* 来结束本节：计算多数函数 MAJ。即使构造一个非单调的多项式大小公式 MAJ 的任务也不完全简单——请尝试一下。然而，MAJ 是否有多项式大小的单调公式一直是个悬而未决的问题。

[11] Indeed, such circuits can only compute small, monotone DNFs (namely, disjunctive normal form formulas).

[12] This paper also discusses how this view extends to computation itself, and surprising consequences it generated beyond lower bounds.

[13] A completely different way of proving a (weaker) super-polynomial separation follows the KRW conjecture [KRW95] (Conjecture 5.13), which in the monotone case becomes a theorem.

几十年，直到20世纪80年代初以两种完全不同的方式解决，一种是由Ajtai、Komlós和Szemerédi [AKS83] 解决，另一种是由Valiant [Val84a] 解决。

Theorem 5.18 [AKS83, Val84a]. *Majority has polynomial-size monotone formulas.*

Ajtai、Komlós和Szemerédi的证明完全是构造性的，但极其复杂，它导出了一个指数在数百的多项式！与Valiant的证明极其简单、优雅，并导出一个小多项式界限形成对比，但它只是一个存在性证明。在概率方法力量的最令人震惊的例子之一（参见[AS00]），他表明，虽然构建小单调多数公式似乎很困难，但几乎所有这些都能计算多数。已知的大小上界是$O(n^{5.3})$。已知的下界（甚至在非单调公式中）是$\Omega(n^2)$。我们还没有工具来回答以下问题。

Open Problem 5.19. 存在一个单调公式，其大小为多数 $O(n3)$?

5.2.4 Natural Proofs, or, Why is it hard to prove circuit lower bounds?

20世纪80年代见证了证明自然、受限电路类电路下界的新技术的涌现。除了*approximation method*之外，还包括Furst、Saxe、Sipser [FSS84]和Ajtai [Ajt83]的*random restriction*方法（用于证明常深电路的下界，我们尚未讨论），Karchmer和Wigderson [KW90]的*communication complexity*方法（用于上述单调公式下界），以及其他方法。但它们以及所有后续结果都未能获得任何非平凡的一般电路下界，特别是证明$\mathcal{P} \neq \mathcal{NP}$。

这是否有根本原因导致这种失败？同样的问题也可以针对任何长期存在的数学问题（例如，黎曼猜想）提出。一个自然的（模糊的！）答案是，可能目前使用的工具和思想（这些工具和思想在攻击相关、更容易的问题上可能已经取得了成功）不足以应对。[16]

引人注目地，复杂性理论可以将这个模糊的陈述转化为定理。因此，我们为迄今为止的失败找到了一个正式的借口：我们可以将一组通用的思想和工具进行分类，这些工具负责了几乎所有已知的受限电路下界，但在证明一般性的下界时必然失败。这个由Razborov和Rudich [RR97] 开发的内省结果，提出了一种称为*Natural Proofs* 的框架。非常简短地说，如果一个下界证明 *natural* 应用于一个 *large, easily recognizable* 函数集，那么它就是一个下界证明。他们首先表明，这个框架包含了几乎所有已知的电路下界。然后他们表明，一般电路下界的自然证明在以下意义上是不可能的。任何一般电路下界的一个自然证明意外地暗示，作为一种副作用，一个子指数算法可以用于逆每个候选的单向函数。

具体来说，在这个形式意义上)的下界将意味着对于整数分解和离散对数等函数的亚指数算法，这些函数通常被认为具有指数级难度（以至于全球电子商务的安全性依赖于这样的假设）。这种联系强烈依赖于*pseudo-randomness*，这将在稍后讨论（在第8.4节末尾）。一个简单的具体推论（参见[RR97]以获取更一般的陈述）是，不存在自然证明整数分解需要大小至少为$2^{n^{1/100}}$ (的最佳当前上界是$2^{n^{1/3}}$)。

数学公理的选择（例如，在著名的 *continuum hypothesis* 情况中）是否导致了证明下界难度的出现？这个问题在 [Aar03] 中进行了综述。对自然证明框架工作的一个解释是作为逻辑证明系统的一个“独立性结果”。具体来说，Razborov [Raz95b] 启动了从某种自然片段的佩亚诺算术证明一般电路下界的独立性的形式研究（这与第6章末讨论的证明电路下界命题公式的难度相关）。这种独立性在公理化逻辑力量方面有多高？请注意，这是可能的

[14] Making essential use of *expander graphs*, which we will meet in Section 8.7.

[15] It solves a much more general problem: constructing a linear size, logarithmic depth sorting network, of which this result is a corollary.

[16] Such was the case, for example with Fermat's last theorem for centuries, during which tools developed to eventually allow Wiles and Taylor to prove it (and much more).

该 $\mathcal{P}$ 与 $\mathcal{NP}$ 问题独立于佩亚诺算术，甚至独立于ZFC集合论（正如连续统假设确实如此）。虽然 $\mathcal{P}$ 与 $\mathcal{NP}$ 的这种独立性是数学上的可能性，但今天很少有人相信这一点。

“自然证明”框架在多个方向上得到了扩展。这些扩展不仅产生了新的障碍和对其能力的障碍，还产生了新的计算模型（如*span programs*）和新的联系（例如，与*secret sharing*密码学问题的联系）。许多这些方面在[Wig93]中得到了概述。

Santhanam [San09] 和 Vindochandran [Vin04] 在5.1节中提到的下界是绕过自然证明障碍（以及相对化）的电路下界。然而，正如我们那里提到的，它们都进行了代数化；参见 Aaronson 和 Wigderson [AW09]。唯一一种避免所有已知障碍的电路下界技术起源于 Williams（参见 [Wil14, AW18]）。它使用了一种巧妙的对角化和模拟的组合，另一方面使用电路复杂性技术[17]。不幸的是，这一领域迄今为止为相对较弱的电路类提供了相对较少的下界。

为了证明超多项式 *formula* 下界，5.13 关于组合的猜想（如第 5.2.2 节所述）似乎也避免了所有已知的障碍，因此可以使用目前可用的技术来证明此类下界。但尚未取得成果。

给定上述讨论，人们当然会期望对于给定的布尔函数，最小电路大小问题（比如说）在计算上是困难的。这个计算问题的精确表述，称为最小电路大小问题，如下所示：给定一个布尔电路作为输入，判断是否存在一个更小的电路来计算相同的函数。我们甚至不知道这个问题是否是$\mathcal{NP}$-困难的，其计算复杂度是一个研究热点领域（参见[AH17]以获取最近的综述）。也许这个（某种程度上自我指涉的）对
*complexity of complexity*的研究将帮助我们证明下界?

Concluding, I view the mystery of the difficulty of proving (even the slightest non-trivial) computational difficulty of natural problems to be one of the greatest mysteries of contemporary mathematics.

[17] Especially one from [IKW02], a result surprisingly based on *pseudo-randomness*! We will discuss interactions of circuit complexity and pseudo-randomness in Section 7.3.

6 Proof complexity

读者将根据经验，在各种数学领域中看到各种数学定理及其证明。然而，本章所关注的证明以及在此处考虑的定理类型，很可能与您所习惯的不同。这种对证明的新视角极其丰富，揭示了一组美丽的数学问题和结果。关于这方面的广泛调查，请参阅[BP98, Seg07, RW00]，并在更广泛的背景下，参阅Kraj ˇcek的书籍[Kra95, Kra19]。

数学中 *proof* 的概念是区分数学研究与人类其他研究领域的主要因素。证明确立了数学 *theorems*，其有效性独立于物理现实。虽然证明通常在数学论文和书籍中以非正式的形式呈现，但我们对其结果，即定理的信心，源于（或信念）它们可以被完全严格化——也就是说，形式化为纯粹 *syntactic* 形式，其绝对正确性可以轻松验证。这种验证算法由一个 *proof system* 指定，它确定对于任何两个符号序列 x, y，y 是否是 x 的证明。在大多数数学证明系统中，y 是从一组“自明”公理推导出 x 的有效推理序列。这种形式化使得证明本身成为数学研究的对象，主要在证明理论和逻辑领域。

不用说，数学实践并非仅仅是句法游戏。虽然严谨至关重要，但数学家们关心他们所证明的内容以及证明方法 *meaning*。因此，定理 x 不会仅仅是抽象符号，而是通常描述某些自然数学结构的有意义属性。同样，证明系统包含各种推理类型，证明 y 通常会结合想法、技术和过去的定理，以“推理出”一个新的定理。经过几个世纪的实践，数学家们经常将诸如“美丽、深刻、原创、深邃”等形容词归因于证明，最值得注意的是“困难”，这正是本节关于证明复杂性的焦点。

是否有可能从数学上量化证明各种定理的难度？这正是*proof complexity*领域所承担的任务。它专注于*propositional*证明系统（我们很快将定义和讨论），这些系统从纯粹逻辑的角度来看是最简单的；这些系统旨在证明关于有限结构的陈述。证明复杂性试图根据证明它们的难度对命题定理（称为“重言式”）进行分类，就像电路复杂性试图根据计算它们的难度对函数进行分类一样。确实，证明复杂性与*proof theory*的关系类似于电路复杂性与*computability theory*的关系。在证明中，就像在计算中一样，将存在许多称为*proof systems*的模型，这些模型捕捉了验证者允许的推理能力（或结构）。

证明系统在数学的所有领域都大量存在（而不仅仅是逻辑），有时是隐含的。确实，一个给定的数学结构具有某种性质的陈述是数学家试图确立的典型例子。而数学中用于确立那些真实陈述——在这种环境下的定理——的 *framework* 通常（且自然地）可以被视为一个证明系统。让我们考虑一些熟悉的逻辑之外的证明系统（我们稍后会回到这一点）并讨论每个证明的 *length* 概念。特别是，这些揭示了证明长度是自然的，我们将在之后进一步讨论这一点。有些可能对你来说不熟悉——请随意跳过或自行了解更多。

1. 希尔伯特的零点定理可以看作是提供了一种（正确且完备）的证明系统，其中 *theorems* 是不一致的多项式方程组。[1] 一个 *proof* 将常数 1 表示为给定多项式的线性组合（具有多项式系数）。2. 每个有限呈现的群[2] 可以看作是一个证明系统，其中 *theorems* 是可以化简为恒等元素的词。*proof* 是将词转换为恒等元素的替换关系的序列。这样的证明有一个很好的几何表示，称为“德恩图”（或范坎彭图），其中证明长度由“面积”——图中区域数来表示。

[1] Such a system may be viewed as a statement: “No valuation to the variables simultaneously satisfies all equations in the system.”

[2] Namely, given by a finite set of generators and relations.

3. Reidemeister 移动是一种证明系统，其中 *theorems* 是平凡（无结）的结的平面图。一个 *proof* 是将给定的结的平面图简化为无交叉的图的移动序列。想象一下一条环形字符串在桌子上打乱，而移动它使其解开的过程是对每次一个交叉点周围的局部变化。

4. 冯·诺伊曼的最小-最大定理为每个零和博弈提供了一个证明系统。一个 *theorem* 是白方的最优策略，其 *proof* 是黑方的策略，保证相同的收益.[3]

在每个这些以及其他许多例子中，证明中的 *length* 都起着关键作用，证明系统的质量（或表达能力）通常与其提供的证明可以有多短有关。

1. 在零特征域的Nullstellensatz中，"系数"多项式的长度（通常通过它们的次数和高度来衡量）在交换代数软件（例如Gröbner基算法）的效率中起着重要作用。

2. 一般的词问题是不可解的。对于双曲群，Gromov关于证明长度的多项式上界有很多用途。其中之一是他自己构造了没有均匀嵌入到希尔伯特空间的有限生成群 [Gro03]。3. 在最近的一项进展中，[Lac15] 给出了关于无结点性Reidemeister证明长度的多项式上界。4. 在零和博弈中，幸运的是，所有证明都是线性的。

我们强调，渐近观点，即考虑如上所述的定理的 *families* 并将它们的证明长度作为已证明定理的描述长度的函数来衡量，在数学中是自然且普遍的。对于计算也是如此，在这里，这种渐近观点也揭示了底层数学对象的结构，并且证明长度的经济性（或效率）通常与更好地理解它们相关。虽然这种观点与大量数学工作相关，但它似乎不足以解释数学家面临的大多数挑战的难度，即证明 *single* 命题（其中不存在渐近性）的难度，例如黎曼猜想或 $\mathcal{P} \neq \mathcal{NP}$。

让我们更深入地探讨这个点。结果往往，这样的“单个”数学陈述可以近似地被观察和研究（当然，这种方法可能或可能不会更好地阐明它们）。例如，黎曼猜想有一个等价的表达形式，即一系列有限陈述（例如，关于Möbius函数的消去）直到每个有限界限n，我们将在第8.3节关于伪随机性中稍后看到。也许更有趣的是，在第8.4节中，我们将讨论将$\mathcal{P}$/多项式与$\mathcal{NP}$问题作为一系列有限陈述的表述，其证明复杂性最终与第5.2.4节中提到的自然证明范例密切相关，该节讨论了证明电路下界困难。这些都是一般现象的例子：一阶逻辑系统中的陈述（如Peano算术或其片段）可以转化为一系列有限命题陈述，其证明长度下界（在适当的命题系统中）可以意味着在一阶设置中的不可证明性。换句话说，我们在这里讨论的命题设置可以提供标准的不可证明性和独立性结果！这个想法最初由Paris和Wilkie [PW85] 在一个特定片段中提出，并由Kraj ček在书籍[Kra95]中更普遍地解释。

所有本章将涉及到的定理都是 *universal* 命题（例如，一组不一致的多项式方程是这样一个陈述：*every* 对变量的赋值无法满足所有方程）。一个普遍陈述的简短证明构成了该陈述的等价表述，即 *existential*——证明的存在本身证明了普遍属性（例如，希尔伯特零点定理中的“系数”多项式的存在，这表明给定多项式方程的不一致性）。这种关注的数学动机是明确的——能够以普遍和存在的方式描述一个属性构成了 *necessary* 和 *sufficient* 条件——数学理解的基础，这在第3.5节中讨论。在这里，我们将非常挑剔，并按照一个（计算）尺度来量化这种理解：存在证明的 *length*。

[3] 自然地，反向也成立，在这个系统中定理和证明之间存在二重性。

我们将限制自己到 *propositional* 重言式，即那些涉及布尔变量的（我们很快就会看到一个例子）。这些可以自然地涵盖关于离散结构的所有真命题，并最终提供一种非常广泛和深入的研究领域。这种关注实际上保证了任何考虑的定理的证明长度的指数（因此是已知、有限的）上界，使我们免于任何 Gödelian 不可证明的担忧。它将潜在证明长度的范围（与 $\mathcal{P}$ 与 $\mathcal{NP}$ 的情况中的时间一样）限制在多项式和指数之间。与计算时间类似，这里的指数证明长度将对应于平凡、“蛮力”证明以及找到巧妙简短证明的可能性（或不可能性）。

我们将处理的陈述、定理和证明的类型，以下例子最能说明。

6.1 The pigeonhole principle—a motivating example

考虑众所周知的“鸽巢原理”，该原理指出从有限集合到较小集合不存在单射映射。虽然这是显而易见的，但我们注意到这个原理对于证明指数级困难函数的 *existence*（定理 5.6）中的计数论证是至关重要的——这部分解释了我们对它的证明复杂性的兴趣。更普遍地说，这个原理体现了数学中的 *nonconstructive* 论证，例如闵可夫斯基定理，即具有足够体积的中心对称凸体必须包含一个格点，或者埃尔德什的关于存在小拉姆齐图的概率证明。在这些以及其他许多例子中，证明并没有提供关于所证明存在的对象的信息。对于其他捕捉拓扑定理本质的组合学恒真命题的证明也是如此（例如，布劳威尔不动点定理、博鲁斯克-乌拉姆定理和纳什均衡）——参见 Papadimitriou [Pap94] 以了解更多。

让我们阐述抽屉原理并讨论其证明的复杂性。首先，我们将它转化为一系列有限陈述。固定 $m > n$。让 PHP_n^m 表示陈述 *there is no* 1–1 *mapping of* m *pigeons to* n *holes*。为了数学地表述这一点，想象一个 $m \times n$ 矩阵布尔变量 x_{ij} 描述一个假设的映射（其解释为 $x_{ij} = 1$ 表示第 i 个鸽子被映射到第 j 个洞）。[5]

Definition 6.1 (The p igeonhole 原理)**.** 鸽巢原理 PHP_n^m 没有 w 状态表明

- 或者某些鸽子 $i \in [m]$ 没有映射到任何地方（即，对于固定的 i，所有 x_{ij} 都是零），或者
- 两只鸽子映射到同一个洞（即，对于某些不同的 $i, i' \in [m]$ 和某些 $j \in n$，我们有 $x_{ij} = x_{i'j} = 1$）。

这些条件可以很容易地用布尔门在变量 x_{ij} (中表示，称为 *propositional formula*)。让我们明确写出：

$$\left(\bigvee_{i\in[m]}\left(\bigwedge_{j\in[n]}\neg x_{ij}\right)\right)\vee\left(\bigvee_{i\neq i'\in[m]}\bigvee_{j\in[n]}(x_{ij}\wedge x_{i'j})\right)$$

鸽巢原理是这样一个公式的陈述，即它由变量*tautology* (的*every*真值赋值所满足，即)。

甚至更方便的是，这个重言式的否定（它是一个 *contradiction*，即一个没有任何赋值可以满足的公式）可以通过对这些布尔变量的相互矛盾的约束集合来捕捉。这些约束集合可以用不同的语言表达：

- **Algebraic:** 作为一个在$\mathbb{F}_2$ (或其他域)上的有界度多项式集。
- **Geometric:** 作为一个具有整数系数的线性不等式集（我们寻求一个 $\{0, 1\}$ 解）。

[4] A graph without large cliques and independent sets.

[5] Note that we do not rule out the possibility that some pigeon is mapped to more than one hole—this condition can be added, but the truth of the principle remains valid without it.

- **Logical:** 作为一个布尔析取集。

我们将看到在第6.3节中，每个设置自然地暗示（几个）推理工具，例如代数设置中Nullstellensatz的变体、逻辑设置中的Frege系统以及几何设置中的整数规划启发式算法。所有这些都可以形式化为可以证明这个（以及任何其他）重言式的证明系统。我们主要关注的是这些证明系统的效率以及它们在*proof length*中的相对能力。在转向这些特定系统之前，我们将全面讨论这个概念。

6.2 Propositional proof systems and $\mathcal{NP}$ vs. co$\mathcal{NP}$

本节中的大多数定义和结果来自Cook和Reckhow [CR79] 发起的这一研究方向的开创性论文。我们定义了证明系统和每个证明长度的复杂性度量，然后将其与我们已遇到的复杂性问题联系起来。

所有我们将考虑的定理都将是对命题重言式。以下是任何证明系统应具备的显著特征，我们期望[6]：

- **Completeness.** 每个真命题都有一个证明。
- **Soundness.** 没有错误陈述有证明。
- **Verification efficiency.** 给定一个数学陈述 T 和一个声称的证明 π，可以很容易地检查（在 $\mathcal{P}$）是否确实 π 在系统中证明了 T。请注意，这里的验证过程效率是指其运行时间，以 *total length of the alleged theorem and proof* 为单位。

Remark 6.2. 请注意，我在 $\mathcal{NP}$ 的定义中删除了所用的要求，将证明限制为简短（与断言长度多项式相关）。当然，原因是证明长度现在是我们衡量复杂度的标准。

所有这些条件都通过以下定义简洁地捕捉，对于命题陈述。

Definition 6.3 (证明系统 [CR79])**.** A (*propositional*) *proof system* 是一个多项式时间算法 M，具有以下性质：T 是一个重言式当且仅当存在一个 (*proof*) π 使得 $M(\pi, T) = 1$。

作为一个简单的例子，考虑以下 truth-table 证明系统 M_{TT}。基本上，这台机器将声明一个公式 T 为定理，如果它在每个可能的输入上评估都使 T 为真。更正式地说，对于任何在 n 变量上的公式 T，机器 M_{TT} 接受 (π, T) 如果 π 是一个长度为 n 的 *all* 二进制字符串的列表，并且对于每个这样的字符串 σ，$T(\sigma) = 1$。

注意，M_{TT} 确实是一个证明系统：它是可靠的、完备的，并且在输入长度上以多项式时间运行，这是公式和证明的长度之和。但在系统 M_{TT} 中，证明的长度与变量的数量呈指数级，因此通常也与给定公式的长度呈指数级。我们将关注这个长度。这引导我们定义一般命题证明系统 M 的效率（或复杂性）——每个重言式的最短证明有多短。

Definition 6.4 (证明长度 [CR79])**.** 对于每个重言式 T，令 $\mathsf{S}_M(T)$ 表示 T 在 M (中的最短证明的大小，即最短字符串 π 的长度，使得 M 接受 (π, T))。令 $\mathsf{S}_M(n)$ 表示所有长度为 n 的重言式 T 上 $\mathsf{S}_M(T)$ 的最大值。最后，如果对于所有 n，我们都有 $\mathsf{S}_M(n) = n^{O(1)}$，则称证明系统 M *polynomially* 为 *bounded*。

[6] Actually, even the first two requirements are too much to expect from strong proof systems, as Gödel famously proved in his Incompleteness theorem. However, for propositional statements that have finite proofs, there are such systems.

存在一个多项式界证明系统（即，对所有tau-逻辑都有多项式大小的证明）吗？以下定理提供了这个问题与计算复杂性和第3.5节的主要问题的基本联系。其证明直接从SAT的$\mathcal{NP}$-完备性、满足命题公式的难题、一个公式不可满足当且仅当其否定是重言式的事实，以及在任何命题证明系统中，一个简短证明（在$\mathcal{NP}$的意义上）证明了这种不可满足性的观察得出。

Theorem 6.5 [CR79]. *There exists a polynomially bounded proof system if and only if* $\mathcal{NP} = \text{co}\mathcal{NP}$.

在下一节中，我们关注自然限制证明系统。请注意，Cook和Reckhow [CR79] 引入了一种称为*polynomial simulation*的证明系统之间的归约概念，这使我们能够创建一些系统相对能力的偏序。这只是复杂性理论在$\mathcal{NP}$-完备性成功后发展起来的计算复杂性方法有用性的一个例子。

6.3 Concrete proof systems

几乎所有本节中的证明系统都是熟悉的类型，可以追溯到欧几里得在*The Elements*中为平面几何引入的演绎系统。每个证明都以公式列表（假定“正确”）开始，并通过连续使用简单的（并且是可靠的！）推导规则，推断出新的公式（每个公式在证明中被称为*line*）。通常，起始公式将是*axioms*，最终推导出的公式将是已证明的定理。

这里，以及在这个研究领域中，通常更方便关注与*contradiction*系统相关的概念，这些系统通过反驳其否定来证明一个定理。更确切地说，一个人从一个矛盾的公式集合开始，并推导出一个显然的矛盾（例如$\neg x \wedge x$，$1 = 0$，$1 < 0$），这取决于设置。我们强调了在代数、几何和逻辑系统中基本重言式证明长度的一些结果和开放问题。在每个这些部分中，我给出了一个证明系统在行动中的例子，反驳了图13中的小矛盾ϕ，它有五个（相互矛盾的）“公理”在四个变量x, y, z, w中。

$$\begin{aligned}
&\text{A1}\ \ \neg x \vee w\\
&\text{A2}\ \ \neg w \vee y\\
&\text{A3}\ \ \neg y\\
&\text{A4}\ \ x \vee y \vee z\\
&\text{A5}\ \ \neg z \vee x
\end{aligned}$$

图13. 矛盾 ϕ。

6.3.1 Algebraic proof systems

这里，布尔矛盾的自然表示是一组没有公共根的多项式。固定一个域$\mathbb{F}$ (，但并非所有结果在任意域)中都成立。为了确保我们只考虑具有布尔（0, 1）值的根，我们总是向这样的集合中添加每个变量x)的多项式$x^2 - x$ (。这些添加的“公理”确保所有可能的根都在$\mathbb{F}$本身中（无需代数闭包）。这也有效地使证明中的所有多项式都是多线性的，并极大地简化了在一般情况下出现的一些问题——例如，多项式的次数永远不会超过变量的数量。

注意，总可以将布尔公式编码为多项式。这里有一种表示上述鸽巢原理约束 PHP_n^m 的方法，定义为（定义6.1）一组矛盾（常数度）多项式。对于每个鸽子 i，添加多项式 $\sum_j x_{ij} - 1$，对于每两只鸽子 i, i'，以及每个洞 j，添加多项式 $x_{ij}x_{i'j}$。

现在，让我们讨论两个具体的代数证明系统：零点定理(NS)和多项式演算(PC)。

The Nullstellensatz (NS) proof system 论文[BKI+96]提出了一种基于希尔伯特零点定理的证明系统，该定理表明，如果多项式$f_1, f_2, \ldots, f_m$在n变量上没有公共根，则常数函数1包含在这些多项式生成的理想中。换句话说，必须存在多项式$g_1, g_2, \ldots, g_m$，使得

$$\sum_i f_i g_i \equiv 1.$$

注意，系数 g_i 确实构成一个证明：如果 f_i 有一个共同根，这样的恒等式就不能成立。这保证了正确性。完备性由希尔伯特著名的零点定理给出，在这个布尔设置中，证明要容易得多。

自然证明度量的度量 $length$ 是多项式作为所有系数列表的描述长度（这被称为 $dense$ 表示）。Nullstellensatz 证明的另一个重要复杂度参数是 $degree$：如果对于所有 i，多项式 $f_i g_i$ 的度数至多为 d，则证明具有 d 度。注意，如果 d 是任何证明的最小度数，那么在这个稠密表示中，任何证明的长度至少为 $\binom{n}{d}$，最多为 $m(\text{\^{}}2n)^d$。因此，度 d 实际上决定了这种表示中的证明长度，因此我们将关注度。最后，请注意，在这个系统中证明验证很容易；一个简单的多项式时间算法可以有效地测试给定的 g_i 是否满足 $\sum_i f_i g_i \equiv 1$.

The polynomial calculus (PC) proof system 一个相关的证明系统，直观上基于Gröbner基的计算，是polynomial calculus，简称PC，由Clegg、Edmonds和Impagliazzo [CEI96]引入。该系统中的$lines$是多项式（再次由所有系数显式表示），并且有两个$deduction\ rules$，捕捉到$ideal$的定义：两个理想元素的加法和理想元素与任何多项式的乘法。具体来说，对于任意两个多项式g、h和变量x_i，我们可以执行以下操作：

- **Addition:** 使用 g 和 h 推导 $g + h$。
- **Extension:** 使用 g 推导 gx_i (，或者更一般地，推导 gh)。

观察规则 $soundness$：如果规则输入的多项式在给定的变量赋值上评估为0，则输出多项式也是如此。任务，如在 Nullstellensatz 中，是从公理中推导出常数1。可以将此类证明视为构建矛盾恒等式（如在 Nullstellensatz 中），但以“小步骤”进行，这可能导致证明中出现的多项式的较小次数和描述长度。图14描述了图13中系统 PC 的矛盾 ϕ 的反驳，扩展为树状结构。请注意，公理 A1–A5 被编码为多项式，并且使用了两个推理规则。

Automatizability 可以证明，对于上述两种证明系统，如果存在某个重言式的大小为 s 的证明，那么这个证明可以在时间多项式于 s 的时间内被 $found$。实际上，由于我们是在多项式的密集表示中测量大小，因此在 Nullstellensatz 系统中找到证明简化为求解一个大小为 poly(s) 的线性系统，其变量是证明中出现的所有多项式的系数。对于多项式演算，[CEI96] 给出的证明查找算法是 Gröbner 基础算法的一个简单变体，在我们的命题设置中需要多项式时间，因为证明大小也是如此。

一个具有此属性（即，如果存在，则短证明可以高效地找到）的证明系统被称为$automatizable$，因为可以高效地自动化证明搜索。回忆我们上面关于$\mathcal{P}$与$\mathcal{NP}$与co$\mathcal{NP}$的讨论，我们并不期望真正强大的命题证明系统是可自动化的。

两个系统已经说明，某些类型的推理可能比其他类型的推理更有效率。PC系统已知比Nullstellensatz指数级强大。更确切地说，[CEI96]证明存在需要指数长度Nullstellensatz证明的重言式，但只需要多项式长度的PC证明。然而，对于PC系统，也已知有强的大小下界（如上所述，从度数下界获得）。事实上，鸽巢原理对这个系统来说是困难的。对于其自然的

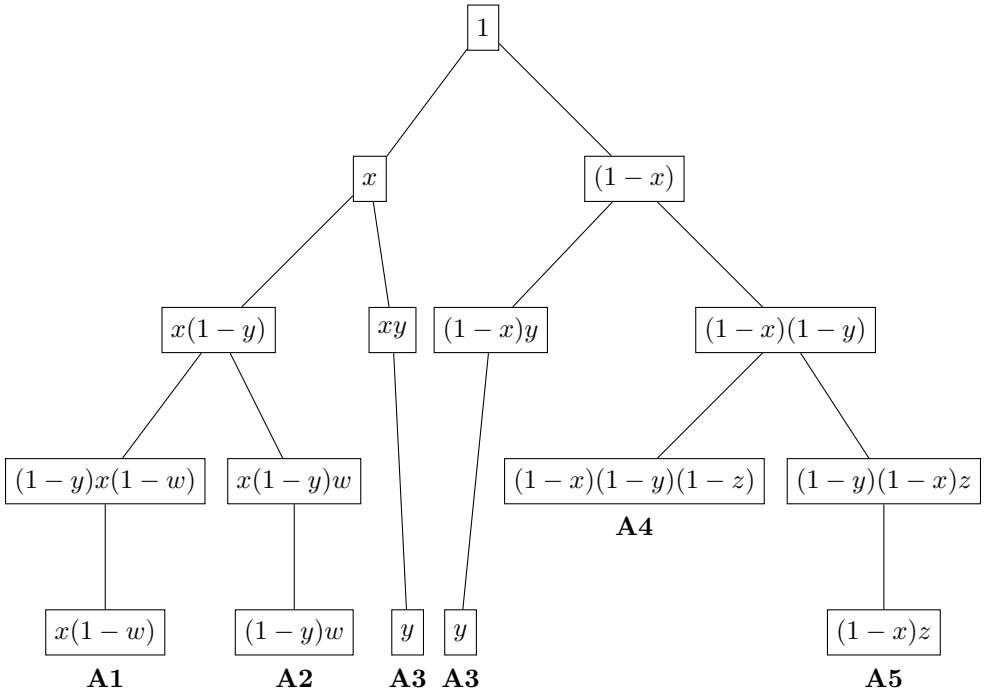


图14. 关于ϕ的树形多项式演算反驳。

编码 *PHP* 为与上述类似的矛盾二次多项式集，Razborov [Raz98b] 证明了以下内容。

Theorem 6.6 [Raz98b]. *For every n and every $m > n$, $\mathsf{S}_{\mathrm{PC}}(PHP_n^m) \geq 2^{n/2}$ over every field.*

6.3.2 Geometric proof systems

另一种在实空间中表示布尔矛盾的自然方法是使用一组不包含整数点的区域。有趣的矛盾丰富的来源是组合优化的整数规划。这里约束是（仿射）线性不等式，具有整数系数（因此区域是布尔立方体由半空间雕刻出的子集）。更普遍地，某些整数规划松弛可以通过多项式不等式系统来表示。一个证明系统以不消除整数点的方式从旧不等式中推断出新不等式。

我们再次用两个具体系统展示这种几何直觉：切割平面(CP)和平方和(SOS)。

Cutting planes (CP) proofs 最几何证明系统称为 cutting planes (CP)，由 Chvátal [Chv73] 提出。它的 *lines* 是具有整数系数的线性不等式。它的 *deduction rules* 是加法和整数除法。具体来说，假设 ℓ_i, m_i, a, b, c 是整数。

- **Addition:** 使用 $\sum \ell_i x_i \geq a$ 和 $\sum m_i x_i \geq b$ 推断 $\sum(\ell_i + m_i)x_i \geq \quad a + b$.
- **Integer division:** 如果 c 整除所有 m_i，则使用 $\sum m_i x_i \geq b$ 推断 $\sum(m_i/c)x_i \geq \lceil b/c \rceil$。

一个反驳从公理中推导出一个基本矛盾（例如，$0 \geq 1$）。图15描述了在系统CP中，图13中矛盾ϕ的反驳，扩展为一个树状结构。请注意，公理A1–A5被编码为线性不等式，并且使用了两个推理规则。

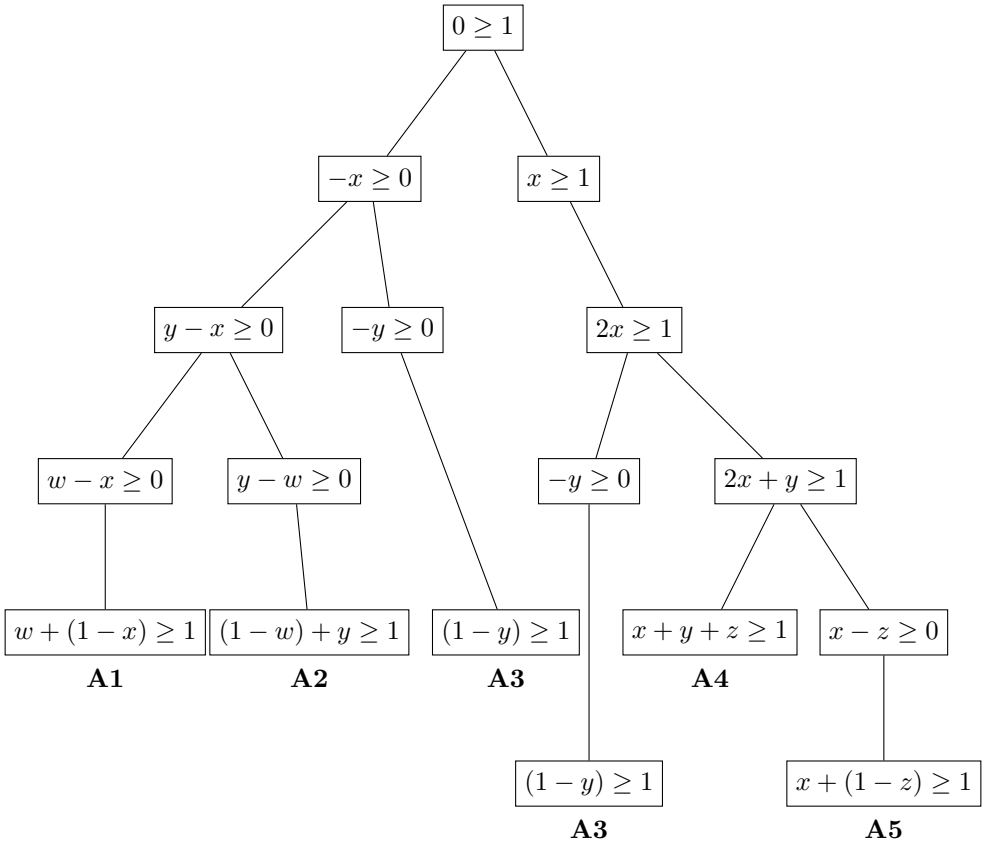


图15. 对 ϕ 的树形 cutting planes 反驳。

可以看出，如果原始布尔公理是析取（如矛盾 ϕ 中所示），那么当我们将它们翻译成上述线性不等式时，当它们有一个满足的整数赋值时，它们也有一个布尔赋值。换句话说，cutting planes 是一个有效且完备的命题证明系统。

再次考虑抽屉原理 PHP_n^m。首先，让我们将其表达为一系列矛盾的线性不等式：对于每只鸽子，其变量的总和应为 *at least* 1。对于每个洞，其变量的总和应为 *at most* 1。因此，以这两种方式加总所有变量意味着 $m \leq n$，这是一个矛盾。因此，抽屉原理有多项式大小的 CP 证明。

虽然 PHP_n^m 在这个系统中很容易，但已经为其他重言式证明了指数下界，接下来我将解释如何做到这一点。考虑重言式 $CLIQUE_n^k$：没有图在 n 个节点上可以同时具有一个 k-团和一个合法的 $(k-1)$-着色。将这个重言式表述为一个命题公式很容易。注意，它以某种方式编码了许多抽屉原理的实例，每个实例对应于顶点的每个 k-子集。**Theorem 6.7** [Pud97]。$\mathsf{S_{CP}}\left(CLIQUE_n^{\sqrt{n}}\right) \geq 2^{n^{1/10}}$.

这个定理的证明由Pudlak [Pud97] 提出，相当引人注目。它将 *reduces* 这个证明复杂性下界转化为电路复杂性下界。换句话说，Pudlak表明，任何关于特定结构的重言式短 CP 证明都产生一个计算相关布尔函数的小电路（这是一个通用方法，在第6.4节中讨论）。你可能已经猜到，对于手头的重言式，要使用的困难函数确实是之前引入的 $CLIQUE$ 函数。因此，如果得到的电路是一个 *monotone* 布尔电路（如第5.2.3节所述），我们就可以通过定理5.14来完成。实际上，Pudlak得到的电路是 *monotone*，但它们更强，因为它们可以使用实数而不是布尔值。更确切地说，这些电路不仅可以使用 $\wedge, \vee$ 作为基本门，还可以使用 *any* 单调二进制运算作为门：它们的输入和输出必须是

布尔，但中间值可以是任意实数。这样的电路确实非常强大——它们可以以线性大小解决 $some$ $\mathcal{NP}$-完全问题 [Ros97]！尽管如此，Pudlak 继续推广 Razborov 的近似方法（第 5.2.3 节）用于此类电路，并证明即使它们也需要指数大小来计算 $CLIQUE$。Haken 和 Cook 独立获得的另一种证明出现在 [HC99] 中。在 15.2.4 节中描述了通过不同方法获得的树形 CP-证明大小的较早下界。

Sum-of-Squares (SOS) proofs 一个更强大的几何证明系统（对于实数上的多项式）最近在优化、机器学习和复杂性中变得重要，被称为sum-of-squares (SOS)系统.[7]它在几篇论文[Sho88, Nes00, Par00, GV01, Las01]中提出，其动机来自优化、统计学（矩问题）和证明复杂性。有趣的是，SOS的起源可以追溯到希尔伯特的第十七个问题，它以同样的方式启发了这个证明系统，就像他的零点定理在上一节中启发了证明系统一样。回想一下，希尔伯特的第十七个问题涉及实数上的多元多项式。它从观察每个多项式的平方和总是非负的出发，并询问逆命题是否成立，即一个在每处非负的多项式是否可以写成有理函数的平方和（$every$）。希尔伯特的第十七个问题被Artin[Art27]肯定地解决了。这些想法的进一步发展导致了Krivine[Kri64]、Stengle[Ste74]、Putinar[Put93]等人的正则化定理，这是实代数几何的基石，它给出了一个多项式方程组和不等式组何时有no公共解的描述。

SOS系统（我们仅为了简化解释，仅用于反驳方程组系统），利用这一特征。它证明了具有任意数量变量)的实多项式集$f_1, f_2, \ldots, f_n$ (没有公共根，通过展示满足以下条件的多项式$g_1, g_2, \ldots, g_n$和$h_1, h_2, \ldots, h_k$:

$$\sum_i f_i g_i \equiv 1 + \sum_j h_j^2.$$

这样的证明被称为具有 d 度，如果所有 $f_i g_i$ 和 h_j^2 的度数都不超过 d。显然，SOS 系统至少与上一节中讨论的 Nullstellensatz 系统一样强大，其中不能使用平方。此外，Grigoriev [Gri01a] 给出了例子，表明 SOS 系统可以指数级地更强（即重言式），[8] 对于 Nullstellensatz 或 PC 反证需要线性度数，但 SOS 以常数度数证明它们（因此以多项式大小）。结果发现 SOS 也比切割平面系统 CP 更强大，以及像 [SA90, LS91] 中的那些基于线性半定证明系统等几个重要的系统。

另一个与Nulstellensatz和PC共享的属性正在$automatizable$。[9]也就是说，如果一个多项式方程组（和不等式）有一个度-d SOS证明，那么这个证明（其大小最多为n^d，所需系数的数量）实际上可以在时间$n^{O(d)}$ (内使用半定规划)进行。对于常数d，这样的证明（如果存在）可以在多项式时间内找到。这个属性对于获得各种（非凸）问题的有效近似算法很重要，这些问题可以表述如下：在$\mathbb{R}^n$的半代数子集中找到给定多变量多项式的最大值，该子集由多项式方程和不等式定义（例如，所有都是常数度）。SOS提供了一系列近似算法，这些算法按用于证明此近似的多项式度d进行参数化。随着d的增加，近似质量通常提高，而运行时间增加。已知该算法有众多应用（例如，参见Lasserre [Las09]）。Barak和Steurer [BS14]概述了最近的应用和与复杂性理论、机器学习、量子信息等领域的联系。许多基本不等式（柯西-施瓦茨不等式、Hölder不等式、多项式的超合同不等式、各种范数的三角不等式等）都有$constant\text{-}degree$ SOS-证明。[10]

[7] It is also referred to in the literature as Positivestellensatz or Lasserre.

[8] For example, that $x_1 + x_2 + \cdots + x_n = \frac{1}{2}$ has no 0/1-solution.

[9] Although one should consider the caveats and precise statement in [O'D17].

[10] For example, here is a degree-4 proof of the Cauchy-Schwarz inequality: $(\sum_i^n x_i^2)(\sum_i^n y_i^2) - (\sum_i^n x_i y_i)^2 = \frac{1}{2}(\sum_{i \neq j}(x_i y_j - x_j y_i)^2) \geq 0$.

哪些重言式对这个 SOS 系统来说难以处理？像往常一样，我们在这里主要关注离散问题（即布尔变量的多项式方程）如前节所述。它们的编码作为实多项式很容易实现，通过添加多项式 $x_i^2 - x_i$ 作为公理。在这种设置下（由于多项式是齐次的，不失一般性），不难看出证明永远不会需要大于 n 的次数。我们拥有的最强结果之一是对于几乎所有的线性方程组 $\mathbb{F}_2$ 的不一致系统，存在一个线性次数的下界。

Theorem 6.8 [Gri01b,Sch08]. *For every n, let $f_1, f_2, \ldots, f_{10n}$ be randomly and independently chosen linear equations over n variables of the form $x_i + x_j + x_k = b$ (where i, j, k are uniformly random in $[n]$ and b is random in $\{0, 1\}$). Then with probability $1 - o(1)$, the encoded system of real polynomials has no common root, and every* SOS *refutation requires degree $\Omega(n)$.*

度与{v*}证明的大小之间的关系（即，可以通过半定规划在时间$n^{O(d)}$内找到度d证明，因此大小最多为$n^{O(d)}$）被证明是紧的，例如，在[LN15]中对于4 − SAT。在Lee，Raghavendra和Steurer [LRS14]的突破性工作中，建立了半定程序的大小与SOS证明的度之间的紧关系，这暗示了指数级SDP下界。

6.3.3 Logical proof systems

本节中的证明系统都将具有 *lines* 为布尔公式，这些系统之间的差异将在于对这些公式施加的结构限制。我们介绍了最重要的几个：Frege，捕捉“多项式时间推理”，以及 Resolution，在自动定理证明中最有用的系统。

The Frege proof system 最基本的形式化证明系统，称为“Frege系统”，不对被证明操作的公式施加任何限制。作为一个反驳系统，它有一个非平凡的性质*derivation rule*，称为*cut rule* (或Modus Ponens)：

- **Cut rule:** 使用公式 $A \vee C$, $B \vee \neg C$ 推导公式 $A \vee B$。

其他推导规则允许，例如，取两个先前推导公式的合取，以及取一个先前推导公式与任意一个公式的析取。通常，反驳应该从给定的公理推导出一个矛盾（例如，空子句，或 $x \wedge \neg x$）。一个 Frege 证明的 *size* 简单地是其中出现的所有公式的总大小。

每个逻辑基础书籍都有一种略微不同的方式来描述 Frege 系统。计算方法的一个便利结果是，特别是证明系统之间有效简化的概念，Cook 和 Reckhow [CR79] 提出了一个证明，即它们在以下意义上是 *all* 等价的：最短证明（直到多项式因子）与您选择的变体无关。

Frege系统可以多项式模拟*both* Polynomial Calculus[11]和Cutting Planes系统。特别是，上述针对鸽巢原理描述的“计数”CP证明可以在Frege系统中高效进行（并非完全平凡！），得出以下定理。

Theorem 6.9 [Bus87]. (PHP_n^{n+1}) *has* Frege *proofs of size $n^{O(1)}$.*

Frege 系统在逻辑中是最基本的，因为它们最常见，在这些系统中，多项式长度的证明自然对应于关于可行对象的“多项式时间推理”。在某种意义上，Frege是计算类$\mathcal{P}$的证明复杂度类似物。证明复杂性的主要未解决问题是找到任何在Frege系统中没有多项式大小证明的重言式（通常，我的意思是重言式族）。

Open Problem 6.10. 证明 Frege 的超多项式下界 系统。

[11] This is simple over the binary field, and with appropriate representation applies to other fields as well.

[12] A variant of Frege, called Extended-Frege, operates with circuits instead of formulas as lines in the proof (with similar derivation rules) and may perhaps better capture polynomial time reasoning.

The resolution proof system 由于 Frege 的下界难以确定，我们转向 Frege 的子系统，这些子系统既有趣又自然。最广泛研究的是 Resolution。其重要性源于它被大多数命题（以及一阶）*automated theorem provers* 使用，通常称为“戴维斯-普特南”或“DLL”过程 [DLL62]。这个算法族旨在找到在各种计算机科学应用中出现的布尔重言式的证明，从软件和硬件设计的验证和通信协议到基本数论和组合定理的自动证明。

lines 在 Resolution 反驳中是 *clauses*，即文字的析取（如 $x_1 \vee x_2 \vee \neg x_3$）。*cut rule* 在 Frege 中简化为 *resolution rule*：

- **Resolution rule:** 使用子句 $A \vee x$ 和 $B \vee \neg x$ 推导子句 $A \vee B$。

一个 Resolution 反驳从一组相互矛盾的命题（公理）开始，通过重复应用上述的归结规则，推导出空子句（矛盾）。一个 *size* 的 Resolution 证明可以简单地视为证明中的命题数量（因为没有析取的大小超过 n）。注意，Resolution 是 Frege 的限制，其中只允许使用最简单的公式类型——即命题——作为证明中的行。

图16描述了图13中矛盾ϕ的驳斥，该驳斥在Resolution证明系统中扩展为树状结构。

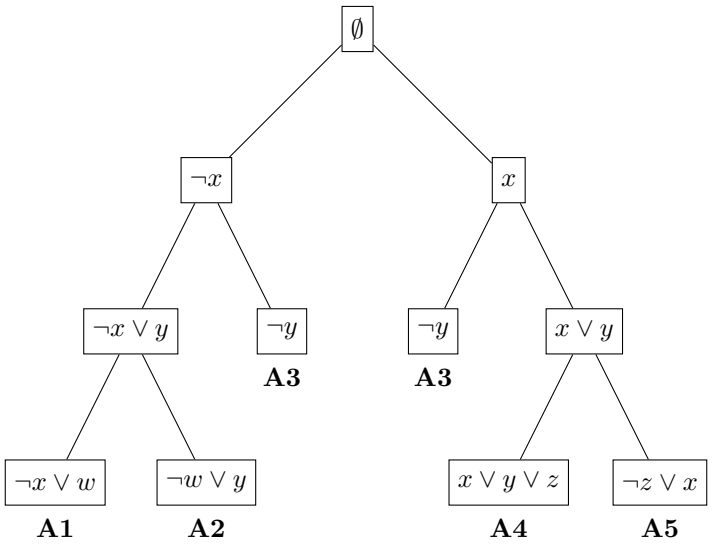


图16. 对 ϕ 的树状 Resolution 反驳。

历史上，证明复杂性第一个主要结果是Haken的[13][Hak85]关于Resolution证明的鸽巢原理的指数下界。

Theorem 6.11 [Hak85]. (PHP_n^{n+1}) *requires* Resolution *proofs of size* $2^{\Omega(n)}$.

为了证明这个定理，Haken 开发了 *bottleneck method*，它与电路复杂性第 5.2 节中提到的随机限制方法和近似方法都有关。这个下界被 Chvátal 和 Szemerédi 在 [CS88] 中扩展到 *random tautologies*。更精确地说，他们证明了随机选择足够多的子句不仅以高概率使它们相互不满足，而且证明这种不满足性几乎肯定需要指数级大的 Resolution 证明。Ben-Sasson 和 Wigderson 在 [BSW99] 中开发的 *width method* 统一并提供了对这些和其他结果以及更简单证明。此外，它揭示了在 8.7) 节中讨论的图 *expansion* (在许多证明复杂性下界中的作用。

[13]Armin Haken, the son of Wolfgang Haken, cited earlier for his work on knots and the 4-color theorem.

寻找存在时短 Resolution 证明的问题（换句话说，这个系统可自动化的程度如何？）由于其广泛用于自动定理证明而极具兴趣。已知的最优界限（分别在 [BKPS02, AR08] 中）是，在 n 变量上的重言式证明的大小 s Resolution 可以在时间 $\exp(\sqrt{n \log s})$ 内找到，但在一个自然的复杂性假设下，至少需要花费时间 $s^{\log n}$。它们中的任何一个可以被改进以缩小这个巨大的差距吗?

6.4 Proof complexity vs. circuit complexity

这两个领域看起来非常不同，尽管计算和证明的局部演化在句法上相似。首先，它们关心的对象数量差异很大。有双指数多的函数（在n位上），但只有指数多的长度为n的重言式。因此，计数论证表明某些函数（尽管不是显式的）需要指数级电路下界（定理5.6），但不存在类似的论证来表明某些重言式需要指数级大小的证明。因此，尽管我们更喜欢自然、显式的重言式的证明长度下界，但在这种设置中，即使是强证明系统中困难重言式的非构造性*existence*结果也是有趣的。

尽管这两个领域的性质不同，但它们之间存在着深刻的联系。在电路复杂性中使用的许多技术，尤其是 *random restrictions*，对证明复杂性也很有用。Pudlak对Cutting Planes的下界，我们在定理6.7中看到，以极其引人入胜的方式使用了电路下界：单调电路下界直接意味着（非单调）证明系统下界。这种特定类型的归约被称为“可行插值方法”（它可以被视为一阶逻辑中*Craig's interpolation*的定量版本），我们将在下面定义它。

Feasible interpolation 一个证明系统具有 *feasible interpolation*，如果它在大小为 s 的证明中证明形式为 $F(x,y) \vee G(x,z)$ (的命题在变量集合 x,y,z) 的不相交集合中是重言式，那么在输入 x 上存在一个大小为 poly(s) 的布尔电路，该电路可以识别当这些 x 变量如此固定时，F 或 G 是否是重言式（当然，其中至少有一个必须是，并且可能两个都是，在这种情况下，任何输出都是好的）。

可行的插值方法由Kraj ́cek [Kra94] 和（更隐晦地）由Razborov [Ra引入 z95b] 为了证明 Resolution 的下界。他们注意到，对于适当的重言式，可行的插值保证的小电路可以被制成 *monotone*。因此，可以用电路的单调下界来证明 Resolution 的下界。这种方法最初由 Bonet、Pitassi 和 Raz [BPR97] 用于 Cutting Planes，并且对于其他相对较弱的证明系统也是已知的。我们注意到，可行的插值是比上面讨论的 *automatizability* 更弱的一个性质，因此代数系统 NS 和 PC 也具有这个性质。然而，读者应该检查，如果可行的插值对于 *every* 命题证明系统成立，那么 $\mathcal{NP} \cap \text{co}\mathcal{NP} \subseteq \mathcal{P}/\text{poly}$（因此我们不会期望强证明系统具有它）。实际上，Kraj ́cek 和 Pudlak [KP89] 证明，如果可行的插值对于标准 Frege 系统成立，那么整数分解是容易的（并且更一般地，单向函数（见第 4.5 节）不存在）。

这个连接引发了这样的问题：是否可以使用类似性质的下推来从（尚未证明的）电路下界中获得强系统（如Frege）的下界。

Open Problem 6.12. $\mathcal{NP} \not\subseteq \mathcal{P}/$多项式是否意味着超多项式Frege下界?

为什么 Frege 的下界难以确定？事实上，我们并不知道。Frege 系统（及其相关，Extended Frege），捕捉 *polynomial-time reasoning*，因为证明中出现的基本对象是多项式时间内可计算的。因此，这些系统的超多项式下界是证明复杂度在电路复杂度中证明超多项式下界的类比。正如我们在电路中看到的那样，我们至少在一定程度上通过自然证明理解了现有技术的局限性。然而，对于证明复杂性，没有已知此框架的类似物。

我以一个自相矛盾的命题作为结论，捕捉了 $\mathcal{P}/$ 多项式与 $\mathcal{NP}$ 的问题。这个命题的证明复杂性可能进一步阐明为什么证明电路下界是困难的。

这个重言式，由Razborov [Raz95a, Raz96] 提出，简单地命题性地编码了陈述 $\mathcal{NP} \not\subseteq \mathcal{P}/\text{poly}$，即 SAT 没有小型电路。更精确地说，固定 n，即 SAT 的输入大小，

并且 s，我们试图证明的电路大小下界.[14] 我们“下界”公式的变量 LB_n^s 编码了一个大小为 s 的电路 C。公式 LB_n^s 简单地检查 C 计算的函数是否至少在一个长度为 n 的实例 ϕ 上与 SAT 不一致。也就是说，要么 $\phi \in SAT$ 和 $C(\phi) =$ 为 0 或 $\phi \notin SAT$ 和 $C(\phi) =$ 为 1。请注意，这个重言式 LB_n^s 的描述大小为 $N = 2^{O(n)}$，因此我们寻求一个关于其证明长度的超多项式下界 N.[15]

证明 LB_n^s 对 Frege 来说是困难的，在某种意义上将给出证明电路下界困难性的另一种解释。这样的结果将与自然证明提供的结果类似，只是不依赖于 *one-way* 函数的存在。但具有讽刺意味的是，证明电路下界的这种无能似乎也阻止了我们证明这个证明复杂性下界！

证明 LB_n^s 对 Resolution 来说是困难的已经极其困难。这涉及到证明一个 *weak* 鸽巢原理的难度——一个比洞更多的鸽子。在几个部分结果之后，这通过Raz的强力方法 [Raz04a] 实现；Razborov [Raz04b] 的进一步加强（对于所谓的函数，到鸽巢原理）最终意味着 LB_n^s 对 Resolution 的难度。

[14] For example, we may choose $s = n^{\log\log n}$ for a super-polynomial bound, or $s = 2^{n/1000}$ for an exponential one.

[15] Of course, if $\mathcal{NP} \subseteq \mathcal{P}/\text{poly}$, then this formula is *not* a tautology, and there is no proof at all.

[16] This explains the connection mentioned between the pigeonhole principle and the counting argument proving existence of hard functions.

7 Randomness in computation

随机性和计算的婚姻一直是计算机科学中最肥沃的思想之一，涵盖了从密码学到计算学习理论再到分布式计算的各种模型。它使我们对基本概念有了新的理解，如知识、保密、学习、证明，以及确实的随机性本身。在本章和下一章中，我们将仅触及冰山一角——与高效计算和证明问题最密切相关的事物。第7.1节和第7.2节讲述了算法随机性的力量和弱点的（看似）矛盾故事。第7.3节解释了伪随机性的概念和伪随机生成器，解释了为什么我们倾向于相信算法中随机性的力量并不像看起来那么强大。更多信息的良好来源是[MR95]、[Gol99]和[Vad11]。

7.1 The power of randomness in algorithms

让我们从一个例子开始，这个例子说明了试图证明恒等式的数学家可能会遇到的潜在困境。假设我们在有理数域 $\mathbb{Q}$ 上工作。$n \times n$ 变量 n 的 $V(x_1, \ldots, x_n)$ Vandemonde 矩阵在 x_i 位置有 $(^{j-1})i, j$。Vandemonde 恒等式如下。

Proposition 7.1. $\det V(x_1, \ldots, x_n) \equiv \prod_{i<j}(x_i - x_j)$.

虽然这个特定的恒等式证明很简单，但许多类似的恒等式却要困难得多。假设你猜想了一个恒等式 $q(x_1, \ldots, x_n) \equiv 0$，用简短的公式（如上所述）简洁地表达，并想知道在投入大量精力证明它之前它是否为真。[1] 当然，如果多项式 q 的变量数 n 和次数 d 很大（如示例中所示），将公式展开以检查所有系数是否为零将需要指数级时间，因此不可行。事实上，对于这个问题，还没有已知的亚指数级时间算法！有没有一种快速简便的方法来找出答案?

一个自然的想法浮现：假设 q 与 *not* 完全相同为零，那么它定义的代数簇（q 消失的点）的测度为零。因此，如果我们为变量选择 *at random* 个值，我们可能会错过这个簇，而得到一个非零的 q_0。然而，如果 q 完全相同为零，那么我们选择的 *every* 赋值将评估为零。实际上，随机选择可以被限制在一个有限域中，以下可以通过对 n 的归纳简单证明。

Proposition 7.2 [DL78, Zip79, Sch80]. *Let q be a非零 polynomial of degree at most d in n variables. Let r_i be uniformly and independently chosen from*[2] $\{1, 2, \ldots, 3d\}$. *Then* $\Pr[q(r_1, \ldots, r_n) = 0] \leq 1/3$.

注意，由于在给定 f 的公式的情况下，在任意给定点评估多项式 q 是容易的，因此上述构成了一个高效的验证多项式恒等式的 *probabilistic* 算法。概率算法与迄今为止我们所看到的算法在两个方面有所不同。首先，它们能够抛掷独立、无偏的硬币，并在计算中使用结果。因此，概率算法的输出是一个随机变量。其次，概率算法会出错。美的是，如果我们愿意接受完美随机性的可用性作为额外输入，以及输出中存在的小错误，我们似乎得到了针对看似困难问题的更高效的算法。

自然中是否存在随机性[3]的深层问题从未阻止人类无论如何假设它，因为赌博、平局决定、民意调查等等。也许自然提供了某种随机性（例如太阳黑子、放射性衰变、天气、股市波动或互联网流量），但这些不可预测事件的实际物理测量并不产生完全独立和无偏见的硬币抛掷。关于这种*weak sources*随机性是否可用于概率算法的问题，以及

[1] This problem turns out to be even more fundamental than it may seem here. It is called the *Polynomial Identity Testing* problem (or PIT for short) and is deeply related to arithmetic complexity theory, the subject of Chapter 12. This problem is discussed at length, for example, in section 4 of [SY10].

[2] A general principle used here and throughout is that the access to random independent *bits* gives easy access to (essentially) uniform random samples from any finite range.

[3] What quantum mechanics says about it will be discussed in Chapter 11.

理论为此而发展，将在第9章中讨论。*Here we postulate access of our algorithms to perfect coin flips, and develop the theory from this assumption.*[4]

概率算法中存在错误似乎是一个严重问题——毕竟，我们计算是为了发现一个*fact*，而不是“可能。”然而，我们在现实生活中确实容忍不确定性（更不用说计算机硬件和软件错误了），因此允许算法中存在不确定性也是有意义的。此外，请注意，概率算法的错误比其他情况下的错误更容易控制——在这里，它可以任意减少，只需在效率上付出小小的代价。假设我们的算法在任何输入上的错误最多为1/3（如上述命题所示）。对于具有这种特性的任何算法，运行它k次（每次都进行独立的随机选择）并对答案进行多数投票，将使每个输入上的错误减少到exp($-k$)。[5]

因此，我们修正了我们对有效计算的概念，允许具有小误差的概率算法，并定义了具有有界误差、概率、多项式时间的类 $\mathcal{BPP}$ (的概率类似物)。我们注意到，可以（并且确实）定义其他确定性复杂类的概率类似物，并且可以研究在这些设置中随机性的能力。

Definition 7.3 (该类 $\mathcal{BPP}$ [Gil77]). 函数 $f : \mathbf{I} \to \mathbf{I}$ 在 $\mathcal{BPP}$ 中，如果存在一个概率多项式时间算法 A，使得对于每个输入 x，$\Pr[A(x) \neq f(x)] \leq 1/3$。

在这个定义中，$A(x)$表示概率算法A的输出随机变量。有时在记法上更方便明确提及算法中使用的随机位，即考虑$A(x)$为$A'(x, r)$。在这里，A'是一个确定性算法，它（除了实际输入x之外）接收一个适当长度的辅助输入r (，该输入被假定为随机位的均匀分布序列。在这个记法中，定义中的要求可以写成对于在$|x|$中多项式时间内运行的确定性算法A'，$\Pr_r[A'(x, r) \neq f(x)] \leq 1/3$。

我再次强调，本定义中的概率界限是在算法的内部硬币投掷 r 上，并且必须对 *every* 输入成立。$\mathcal{BPP}$ 的这个定义对误差概率界限的变化极为稳健：将 $1/3$ 替换为较低的 $1/10^{10}$，甚至 exp($-|x|$)，以及较高的 .4999 9 或甚至 $1/2 - 1/\text{poly}(|x|)$，都不会改变定义（这可以通过上述通过多数投票思想实现的误差减少来得出）。

Remark 7.4. $\mathcal{BPP}$ 这是一个 *decision* 问题类别。在本章中，我们还将考虑关系的概率算法、近似问题、计数问题以及其他问题。我们不会正式定义它们的复杂度类别；直观上，我们寻求一个概率算法，该算法将以高概率高效地为每个输入提供所需输出。关于 $\mathcal{BPP}$ 所述的一切也适用于这些类别。

Adleman [Adl78] 观察到在具有如此微小错误的概率算法中，一些（实际上，大多数）随机字符串同时适用于给定长度的每个输入。允许非均匀性，其中任何一个都可以被硬编码到一个 *circuit* 中，该 *circuit* 将在所有输入上正确计算。因此，对于 $\mathcal{BPP}$ 中的任何问题，都存在 *exist* 小电路.[6]

Theorem 7.5 [Adl78]. $\mathcal{BPP} \subseteq \mathcal{P}/多$.

概率算法在计算机科学存在之前就被用于统计学（用于抽样）和物理学（蒙特卡洛方法——见第20.1.5节）。然而，从Berlekamp的[Ber67]概率多项式分解算法和Solovay与Strassen[SS77]以及Rabin的[Rab80]概率素性检验开始，它们被引入计算机科学后，随之而来的是一系列结果，极大地增加了此类攻击所能解决的问题的多样性和复杂性——这是这一领域的一个缩影

[4] In practical implementations of probabilistic algorithms, these bits are usually generated by a variety of ad hoc “pseudo-random generators.” It is a remarkable empirical fact that almost universally, these ad hoc alternatives to random bits seem to work pretty well.

[5] The notation exp($-k$) means c^{-k} for some $c > 1$. This bound on the error follows from standard concentration bounds on the binomial distribution (e.g., the Bernstein/Chernoff bound). Specifically, the probability that k tosses of a biased coin, whose probability of heads is at most $1/3$, would produce more than $k/2$ heads, is exponentially small in k.

[6] Recall that in Section 5.2, we denoted the class of polynomial-size circuits by $\mathcal{P}$/poly.

可以从中获得Motwani和Raghavan的教科书[MR95]。我们在这里仅限于那些可以保存*time*的，并注意随机性似乎也有助于保存其他资源。

我这里列出了来自不同领域的大量问题，这些问题具有概率多项式时间算法，[7] 但对于其中已知最佳确定性算法需要指数时间。这些是本研究领域的重大成就之一。

- **Generating primes.** 给定一个整数 x (以二进制) 表示，生成区间 $[x, 2x]$ 内的素数（注意，该区间在输入长度 $|x|$ 中呈指数增长）。素数定理保证该区间内的随机数是素数的概率约为 $1/|x|$ (，因此以高概率在 $|x|$ 内以多项式时间出现，我们可以高效地检查其素性)。

- **Polynomial factoring** (Kaltofen [Kal83])。给定一个描述多变量多项式（在大型有限域上）的算术公式或电路[8]，找到它的不可约因子[9]。

- **Permanent approximation** (Jerrum, Sinclair, 和 Vigoda [JSV04])。给定一个非负实矩阵，将其永真值（在第12章定义）近似到（比如说）2的因子内。请注意，与它的相对值行列式不同，行列式可以通过高斯消元法轻松高效地计算，而永真值已知是 $\#\mathcal{P}$-完全（这意味着 $\mathcal{NP}$-难）计算精确值。

- **Volume approximation** (Dyer, Frieze 和 Kannan [DFK91])。给定一个高维凸体（例如，由其边界超平面给出的多面体），将其体积近似到（比如说）2倍以内。再次，精确计算体积是 $\#\mathcal{P}$-完全的。

关于这种新的概率计算计算范式，最基本的问题就是它是否真的为确定性计算增加了任何能力。

Open Problem 7.6. 是 $\mathcal{BPP} = \mathcal{P}$?

答案似乎是否定的：我们根本不知道如何通过一个在亚指数时间内运行甚至是在多项式时间内运行的确定性算法来解决上述问题，以及其他许多问题。然而，下一节应该彻底改变这种观点，通过基本概念 *computational pseudo-randomness*。

7.2 The weakness of randomness in algorithms

让我们从底线开始：如果我们上面看到的众多 $\mathcal{NP}$-完备问题中任何一个都是 *hard*，那么随机性就是 *weak*。*hard* 和 *weak* 这两个词在形式上意味着有权衡。具体来说，我们给出也许是这种结果中最戏剧性的一项，归功于 Impagliazzo 和 Wigderson [IW97]。

Theorem 7.7 [IW97]. *If SAT cannot be solved by circuits of size* $2^{o(n)}$*, then* $\mathcal{BPP} = \mathcal{P}$*. Moreover, the conclusion holds if SAT is replaced in this statement by any problem that cannot be solved by circuits of size* $2^{o(n)}$ *but has* $2^{O(n)}$*-time algorithm.*[10]

重述，对几乎所有感兴趣问题的指数电路下界意味着可以从算法中消除随机性 *always* 而不牺牲效率（至多多项式）。存在许多这种结果变体，通常称为“*de-randomization*”。一个变体给出了硬问题与随机性之间的权衡：削弱对硬问题的假设下界只是削弱了随机性的确定性模拟，但仍然非常非平凡。（换句话说，替代概率算法的确定性算法在枚举值时比暴力搜索要高效得多。）

[7] Note that these problems are not strictly in $\mathcal{BPP}$, because they compute relations or approximation problems.

[8] See precise definitions in Chapter 12.

[9] It is not even clear that the output has a representation of polynomial length—but it does! A structural corollary of this result is that the factors have small arithmetic circuits as well.

[10] The class with such algorithms includes most $\mathcal{NP}$-complete problems but also presumably far more complex ones (e.g., determining optimal strategies of games, which are $\mathcal{PSPACE}$-complete, and beyond).

随机比特）。例如，如果 $\mathcal{NP} \not\subseteq \mathcal{P}/$ 多项式，那么 $\mathcal{BPP}$ 对于每个 $\varepsilon > 0$ 有具有亚指数运行时间的确定性算法 $\exp(n^{\varepsilon})$。

另一个重要扩展是用均匀难度假设（类型为$\mathcal{BPP} \neq \mathcal{NP}$）替换非均匀电路下界。这导致了一种“平均情况”去随机化，正如在Impagliazzo和Wigderson [IW98]中定义和证明的那样。

注意这样一个显著且反直觉的特征：它们断言 *if one computational task is hard, then another is easy!*

鉴于定理7.6，我们现在面临决定放弃两个极具吸引力的信念中的哪一个（因为它们是相互矛盾的）。第一个是某些自然问题（例如，$\mathcal{NP}$-完全问题）不能被有效地解决。第二个是随机性是一个非常强大的算法资源。经验、直觉和最先进的知识似乎都支持这两个信念，但它们似乎在支持第一个信念方面要强得多。因此，大多数专家不情愿地放弃了第二个信念，现在认为随机性不能显著加快算法的速度。也就是说，对于同一任务，可以使用确定性算法代替概率算法，而这些算法的成本并不会高很多——这通常被称为 *de-randomization*。我们将其应用于分类问题（对于搜索和近似问题也有类似定理，但我们将不在此处讨论它们）。

Conjecture 7.8. $\mathcal{BPP} = \mathcal{P}$.

我们现在构建一个关于导致这一令人惊讶的结果集的高级描述，这些结果通常在标题 *hardness vs. randomness*.[11] 下为人所知。使这一 *de- randomization* 成为可能的核心概念是 *computational pseudo-randomness* 和 *pseudo-random generator*。我在这里解释这两个概念，我们将看到它们如何产生去随机化。在第7.3节中，我们将描述它们的历史和重要性，以及去随机化之外的内容，并讨论不同的伪随机生成器。我们建议读者参考Goldreich的书籍第8章 [Gol08] 和他的专著 [Gol99] 以获取更多细节。

我们寻求一种消除任何（有效）概率算法使用的随机性的通用方法。固定任何这样的算法 A。它有两种输入。一种是“真实”输入 x，另一种是“随机性” y，假设其长度为 n 位。误差保证是，对于每个 x，如果 y 是在所有长度为 n 的二进制序列上的均匀分布 U_n 上分布的，那么 $A(x, y)$ 将以最多 $1/3$ 的概率出错。想法是“欺骗” A，将分布 U_n 替换为另一个“看起来像” U_n 的分布 D。换句话说，A 将无法从 U_n 中区分任何输入 x。更精确地说，无论 y 是按照 U_n 还是 D 分布的，在每次 x 中，$A(x, y)$ 接受的概率几乎相同。让我们首先定义可以“欺骗” *all* 有效算法的分布。我们稍后将会看到，实际上使用这样的分布，我们需要使它们 *useful*，并讨论其意义。

7.2.1 Computational pseudo-randomness

第一个关键概念是我们定义的Goldwasser-Micali [GM84]和Yao [Yao82b]的{v*}。为了简单起见，我们将在这里根据电路来定义它。对于一个有n个输入和D在n-位序列上的分布的电路C，用$C(D)$表示从D中抽取y时$C(y)$ =为1的概率。像往常一样，所有定义都应该从渐近的角度来理解。特别是，当我们讨论一组电路$\mathcal{C}$时，我们将隐含地意味着一个由输入长度n参数化的电路序列$\{\mathcal{C}_n\}$，其中$\mathcal{C}_n$通常作为n的函数被限制在大小上。正如本书的其他地方一样，我将滥用符号并省略n (，例如，当参数n隐含时，将$\mathcal{C}$与$\mathcal{C}_n$识别为相同)。图17显示了示意图。

Definition 7.9 (伪随机性). 设 $\mathcal{C}$ 是一个电路族（在 n 位上），且 $\epsilon > 0$。一个在 n 位上的分布 D (被称为 ($\mathcal{C}, \epsilon$)-*pseudo-random*，如果对于每个 $C \in \mathcal{C}$ 我们都有 $|C(D) - C(U_n)| \leq \epsilon$。

[11] The title of Silvio Micali's PhD thesis. Micali, along with his advisor Manuel Blum, constructed the first hardness-based pseudo-random bit generator.

[12] I often omit the "computational" and call it only "pseudo-randomness" in this chapter.

[13] Indeed, I may even change the number of inputs to be some polynomial in n without warning, when this does not affect the argument.

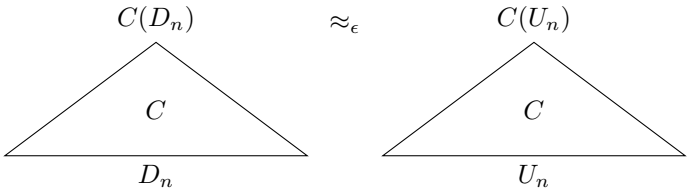


图17. 模拟随机分布 D_n ϵ 欺骗电路 C 的示意图。

换句话说，如果 D 在 $\mathcal{C}$ 中没有任何电路能够“区分”它和均匀分布，并且具有不可忽视的优势 ϵ，则 D 是伪随机的。等价地，D ϵ 欺骗 $\mathcal{C}$。

参数 ϵ 可以是本节目的一个小常数（例如，.01），但通常它可能依赖于 n，随着 n，$\epsilon = \epsilon(n)$ 趋于零，通常像 1/poly(n)。电路类可以是任意的，但本节中，自然将其取为 $\mathcal{P}$/poly，或者更好，某些固定多项式大小的电路族，例如，n^4。原因是对于我们要尝试去随机化的每个有效算法，如果它在时间（例如）n^2 内运行，那么一个电路可以在时间 n^4（内模拟它在每个 n-位输入上的作用，正如我们在第 5.2 节中看到的。

当我们寻求对概率算法的确定性模拟时，我们必须找到一个伪随机分布 D，它可以由0个随机比特有效地生成。但即使我们移除效率要求，这显然是不可能的，因为随机性不能被确定性生成.[14] 幸运的是，我们有一些优势。让我们尝试从*fewer*随机比特中生成这样的伪随机D，例如m比特。这使我们到达了伪随机生成器的下一个重要概念，将一些真正的随机比特拉伸成许多伪随机比特，在Yao [Yao82b]的同一篇论文中定义。

7.2.2 Pseudo-random generators

Definition 7.10 (伪随机生成器). 设 $\mathcal{C}$ 为一组电路。一个函数 $G : \{0,1\}^m \to \{0,1\}^n$ 被称为 $(\mathcal{C}, \epsilon)$-*pseudo-random generator*，如果在其均匀随机输入下，其输出分布 $G(U_m)$ 是 $(\mathcal{C}, \epsilon)$-伪随机的。

图中18显示了示意图。在这个定义中，我们再次从渐近角度思考，将所有参数化由n表示，即我们想要欺骗的电路/算法中随机输入的长度（这同时也是生成器的输出长度）。因此，G应该是一个函数族$\{G_n : \{0,1\}^{m(n)} \to \{0,1\}^n\}$，它能够欺骗具有错误率$\epsilon(n)$的$\mathcal{C}_n$电路。

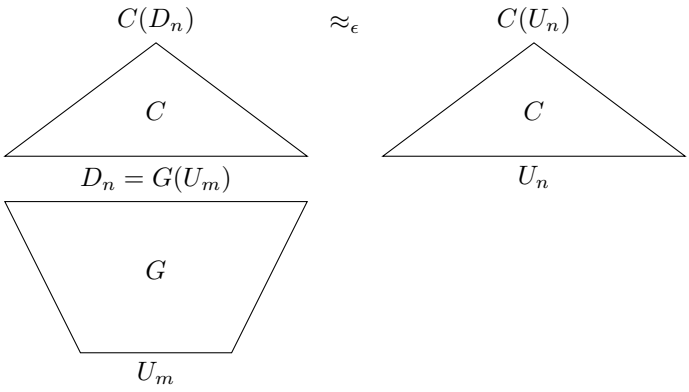


图18. 模拟随机生成器 G ϵ 欺骗电路 C 的示意图。

14确定过程在固定输入上的输出，嗯，是完全确定的。

伪随机生成器可以用来欺骗均匀概率算法，因为它们的每个输入执行都可以通过电路来模拟（参见第5.2节）。例如，如果我们有一个这样的伪随机生成器G，其中$\mathcal{C}$是大小为n^4的电路类，并且有$\epsilon = \frac{1}{20}$ (say)，那么我们可以使用$D = G(U_m)$来欺骗任何概率n^2-时间算法A在每一个输入x上。因为根据定义，在此分布下算法的错误最多为$\frac{1}{3} + \frac{1}{20} < \frac{2}{5}$，这至关重要（正如我们马上将看到的）与$\frac{1}{2}$有界。

但是，我们没有 m 随机位来生成 D，因为我们旨在实现一个确定性算法。一种简单、蛮力[15]的方法是遍历所有可能的序列 $z \in \{0,1\}^m$，对每个序列计算 $y = G(z)$，并使用具有随机性 y (的算法 A，即对每个这样的 y) 计算 $A(x, y)$。在得到所有这些 2^m 个结果后，我们进行多数投票，并宣布这是我们对 x 的答案。请注意，这个确定性过程将 *always* 对每个输入 x (给出正确答案，再次强调，这是真的，因为小于 $\frac{1}{2}$ 在支持 D 时引发错误，我们对整个支持) 进行多数投票。

运行时间取决于两个因素：2^m 和 G (的时间复杂度，实际上，取决于它们的乘积)。让我们逐一处理它们。首先，一个相当标准的计数论据表明，存在一个函数 G，其 $m = O(\log n)$ 是一个 $(\mathcal{C}, \epsilon)$-伪随机生成器。确实，一个随机的 G 几乎肯定具有这个属性。因此，2^m 可以是 n 的多项式，这解决了第一个因素。然而，一个随机的 G 通常计算起来是指数级的困难。我们寻找的是一个可以在多项式时间内计算的生成器 G。这将使第二个因素也是多项式的，并产生 $\mathcal{BPP}$ 的多项式时间模拟。是否存在这样的 *efficient* 伪随机生成器？这正是定理 7.7 提供的，*assuming* 即存在一个足够困难的函数.[16] 这种将困难转化为伪随机性的方法也在下一节中讨论。

7.3 Computational pseudo-randomness and pseudo-random generators

我们现在探讨这两个核心概念的历史和意义。第7.2节将它们描述为去随机化概率算法的极其自然的选择。确实如此。但它们并非天生如此，而是源自密码学和电路复杂性！这个故事再次展示了计算复杂性中思想的奇妙流动。

7.3.1 Computational indistinguishability and cryptography

计算伪随机性的概念实际上是一个更广泛概念（重要）的特殊情况，即*computational indistinguishability*，由Goldwasser和Micali在早期定义[GM84]（本章第18节将详细讨论此论文）。它认为，如果没有任何有效程序能够区分它们，则认为两个概率分布是计算上不可区分的。

Definition 7.11 (计算不可区分性)**.** 设 $\mathcal{C}$ 是一组电路。在 n 位上的两个分布 D 和 D' 被称为 $(\mathcal{C}, \epsilon)$-*indistinguishable*，如果对于每个 $C \in \mathcal{C}$，我们有 $|C(D) - C(D')| \leq \epsilon$。

观察发现，如果 D 和 U_n 不可区分，则 D 是伪随机的。这个定义的动机是密码学（欲了解更多，请参阅 Goldreich [Gol04] 的全面文本），我们简要描述一下。在他们的开创性论文中，Goldwasser 和 Micali [GM84] 为该领域奠定了形式数学基础。为了说明计算不可区分性的实用性，考虑密码学中最基本的概念：一个 *secret*。但是，秘密的正确定义是什么，如何才能实现它？Goldwasser 和 Micali 提出了基本概念 *probabilistic encryption*，它甚至可以隐藏一个比特，如下所示。他们的神奇加密方案将概率分布 D_0 和 D_1 分别分配给每个比特 0 和 1，在 n 比特上具有以下两个看似矛盾的性质。首先，它们具有不相交的支持集，因此从信息论的角度来看，它们显然是完全可区分的。其次，它们是

[15] But useful, if m is very small.

[16] It is a good exercise to convince yourself that a hardness assumption is needed. Hint: Consider the complexity of determining if a given n-bit sequence is in the image of G, and argue that if G is a pseudo-random generator, then this function must be hard.

计算上不可区分，因此它们对高效观察者来说看起来是相同的。Goldwasser和Micali [GM84]表明，不仅存在这样的分布，而且还展示了如何高效地生成它们。美妙之处在于，*by definition*，没有任何高效的过程甚至*guess*以比$\frac{1}{2}+\epsilon$更好的概率来*guess*秘密位。更普遍地，通过重复，这允许以确保不泄露*any*部分信息的方式加密更长的消息。计算上不可区分性允许加密方案的设计者完全忽略有关攻击系统特定类型敌手的各种细节，而只需考虑他们的计算能力限制。

一些人在这个时候可能会想知道，如果*no one*能说出秘密，这如何对加密有用。好吧，它们的构造也关键地提供了一个陷阱门，允许“适当的当事人”（拥有额外知识）能够有效地区分这两个分布，并解密加密信息。所有这些魔法都依赖于一个“密码学硬度假设”，就像第4.5节中提到的一次函数或陷阱门函数，在论文[GM84]中，这些恰好是整数分解硬度的变体。更精确地说，并且与本书的一个主要主题相关，[GM84]提供了一种新的*reduction*类型。Goldwasser和Micali证明，破解这个加密方案（即，从其加密中至少以$\frac{1}{2}+\epsilon$的概率猜测秘密）将意味着比硬度假设允许的更快算法（例如，将提供一个比目前所知的更快分解算法）。

加密协议的对手被视为计算上有限实体，并使用计算不可区分性来证明它们无能为力，这种观点的力量才刚刚在这个例子中显现出来。论文继续发展基于这些原则的加密协议安全性的形式定义，这些定义支撑着实际上成千上万的定义和证明加密原语和协议性质的密码学论文。其中一个例子*zero-knowledge proofs*在第10.2节中进行了非正式解释，并正式依赖于计算不可区分性。不用说，在分类对手时关注计算复杂性，与这些众多原语和性质之间的归约网络完美契合，并允许它们基于数学上干净且经过更好测试的难度假设。

观察计算不可区分定义的灵活性。它允许任何一类敌手（也称为 *observers*、*distinguishers* 或 *tests*）$\mathcal{C}$，并且确实不同的设置邀请不同的类别。一个特别有趣的类别是 *all* 电路，或者说所有布尔函数。在这种情况下，D 和 D' 的计算不可区分意味着什么？简单地说，这意味着对于每个布尔函数 f，我们都有 $|f(C) - f(D')| \leq \epsilon$，它限制了这两个分布的 *statistical distance*（即它们差异的一半的 L_1 范数）：$\frac{1}{2}|D - D'|_1 \leq \epsilon$。这表明限制观察者的类别给 L_1 距离度量带来了一定的 *coarsening*。它允许在新的度量下，两个支持不相交的分布非常接近，并且在第 7.2 节中，我们看到它允许具有不同熵的分布在这个度量下非常接近（尽管在这两种情况下，统计距离是最大的，本质上为 1）。这种信息论和计算复杂性设置之间的二分法是现代密码学的核心，也将在第 18 章中再次强调。

让我们再考虑一个关键点。随机性，本节的英雄，被人类使用了数千年，并且被数学家和科学家们研究了数百年。在概率论、统计学、统计物理、信息论、遍历理论、混沌理论、 Kolmogorov复杂度，甚至哲学中，都有关于随机性的各种方法、观点和理论。这种新的计算随机性理论在根本方面与它们都不同。在所有过去的方法中，无论是定性的还是定量的，现象的随机性（无论是抛硬币还是股市波动或DNA序列），都是现象本身的*objective*属性。在计算伪随机性中，它是*sub-jective*，观察者的属性！完全相同的（客观）现象（例如，相同的伪随机分布，甚至是一次单独的抛硬币）可能被计算有限的观察者认为是随机的，而不会被没有计算限制的观察者认为是随机的。

7.3.2 Pseudo-random generators from hard problems

对于本节，读者可能需要回顾第4.5节关于单向和陷阱门函数的内容，以及第5.2节关于电路复杂性的内容。这两个非常不同的困难函数来源将引导我们走向两种构建伪随机生成器的类型或范式，这两种范式在构建上有所不同。

仅在硬度源上使用，但在其他重要特征和一些应用中也是如此。两者有时被称为“BMY-发生器”和“NW-发生器”，将依次进行讨论。

伪随机生成器（简称PRG）在这些问题出现之前就已经使用了数十年。它被用来描述任何临时方法（尤其是那些在需要随机位的各种系统和算法中实际使用的方法），用于将短序列确定性地扩展为长序列。对PRG输出分布的一般性质进行理论研究的兴趣可能始于冯·诺伊曼[vN51]及其早期的计算机（该计算机需要随机位进行蒙特卡洛模拟和天气预报）。冯·诺伊曼关于这个主题的著名引言是

*“Anyone who considers arithmetical methods of producing random digits is, of course, in a state of sin.”*众所周知，冯·诺伊曼没有遵循自己的建议，我们也是如此.[17]

从难以计算的难题构建伪随机生成器的想法是在一系列作品的演变中逐渐形成的。我们现在回顾这些作品的一些主要思想、不同的动机和后果。这个高级描述必然是粗略的；更多细节，请参阅[Gol99, Gol04, Vad 11]。在这个故事中，计算复杂性方法、计算抽象和有效归约的计算复杂性领域的互联性都得到了强有力的体现。

BMY-generators Shamir首次给出了伪随机生成器的基于复杂度的定义[沙83]。再次，他的动机是密码学。他认为，如果一个用户生成一个序列，另一个用户可以 *predict*，那么安全和隐私可能会被侵犯。他将输出分布不可预测的生成器称为 *cryptographically strong*，并建议基于单向函数设计这一属性。Shamir 的计划由 Blum 和 Micali [BM84] 完全执行。他们正式定义了 *unpredictable gen- erators*，其中没有后续输出位可以被非平凡地猜测（概率为 $> \frac{1}{2} + \epsilon$），给定其前驱，由 *any* 高效观察者（在某些类 $\mathcal{C}$ 中）。让我们更精确地陈述（从左到右）不可预测分布和生成器的定义。对于一个分布 D，用 D_k 表示其在 k 位上的投影，用 $D_{[k]}$ 表示其在前 k 位上的投影。

Definition 7.12 (伪随机生成器{v*}设{v*}为一个电路族。

- 一个在 n 位上的分布 D 被称为 $(\mathcal{C}, \epsilon)$-*unpredictable* (从左到右顺序)，如果对于每个 $i \in [n]$ 和每个 $C \in \mathcal{C}_{i-1}$，我们有 $\Pr[C(D_{[i-1]}) = D_i] \leq \frac{1}{2} + \epsilon$。
- 一个函数 $G : \{0,1\}^m \to \{0,1\}^n$ 被称为 $(\mathcal{C}, \epsilon)$-*unpredictable generator*，如果在其均匀随机输入下，其输出分布 $G(U_m)$ 是 $(\mathcal{C}, \epsilon)$-不可预测的。

Blum和Micali展示了如何构建这样的高效不可预测生成器 G，基于离散对数函数的困难性（在第4.5节中作为单向函数候选进行讨论）。这里的理念信息是，某些困难假设可以保证在对抗任何高效观察者的情况下，高效生成具有强大伪随机属性的分发，*unpredictability*。

这让我们更接近伪随机生成器的目标吗？不可预测的分布与伪随机分布之间有什么关系？例如，如果一个分布从左到右不可预测（如上所示），那么它是否在其他方向上也不可预测？在继续阅读之前，请思考一下。

答案是“是”！序列中的下一篇论文，由姚 [Yao82b] 提出，证明了 *every unpredictable distribution is pseudo-random*。更确切地说，如果 D 是 $(\mathcal{C}, \epsilon)$-不可预测的，那么它也是 $(\mathcal{C}', \epsilon')$-伪随机的，其电路在 $\mathcal{C}'$ 中略小于 $\mathcal{C}$，并且 ϵ' 略大于 ϵ。这立即意味着 Blum-Micali 生成器是一个基于离散对数难度的伪随机生成器。在同一篇论文中，姚将这一结果扩展到一类广泛的单向 *permutations*.[18] 最后，姚建立了伪随机生成器与去随机化之间的联系，在自然难度假设下给出了 $\mathcal{BPP}$ 的第一个（高度）非平凡去随机化。

[17] Another famous early quote on the subject is the title of Coveyou's paper [Cov69]: “Random number generation is too important to be left to chance.”

[18] This is simply a one-way function which happens to be a permutation. Both examples from Section 4.5, modular exponentiation and modular powering, are indeed permutations.

Theorem 7.13 [姚82b]. *If one-way permutations exist, then for every $\epsilon > 0$, every problem in $\mathcal{BPP}$ can be solved deterministically in deterministic time exponential in n^{ϵ}.*

伪随机生成器基于单向函数通常被称为*cryptographic*生成器，有时也称为“BMY-生成器”，以纪念其发明者。这些生成器非常高效（在一种我们很快会讨论到的意义上），对于去随机化[19]来说绰绰有余，但对于密码学应用至关重要。回想一下，单向函数易于计算但难以逆转（例如，模幂运算），其逆函数，离散对数，被认为是难以计算的。这种二分法被如下使用。生成器的效率取决于函数容易方向的复杂性。其输出是伪随机的观察者类别取决于函数困难方向的复杂性。因此，这个生成器可以在多项式时间内运行，并且可能（取决于假设的单向函数强度）能够抵御超多项式或甚至亚指数时间攻击。这对于密码学至关重要，在密码学中，典型用户（如携带笔记本电脑的用户）远不如潜在的对手（如拥有大量计算资源的公司、政府或罪犯）强大。但密码学PRG对抗更强大对手的能力也导致更多概念上的影响，我们现在将讨论这些影响。

首先，很容易看出加密PRG隐含了一阶函数的存在，几乎可以说是定义上的（在足够长的输出部分上逆转它将允许完美预测其余部分）。Yao证明了存在一阶*permutations*隐含这样的PRG，而一系列工作最终在Håstad、Impagliazzo、Levin和Luby [HILL99]的一篇论文中填补了这一空白。他们证明了这两个概念是等价的：一阶函数存在当且仅当加密PRG存在！因此，在加密设置中，最自然的困难假设和最自然的伪随机性概念是一致的。

Theorem 7.14 [HILL99]. *The following are equivalent:*

- *There exist one-way functions.*
- *There exists a 多(n)-time computable $(\mathcal{P}/poly, 1/$多$(n))$-generator $G : \{0,1\}^m \to \{0,1\}^n$ for any $n =$ 多(m).*

注意，一个直接的推论是在存在单向函数的（看似）较弱假设下，定理7.13的逆随机化后果相同。关于单向函数存在性的进一步重要后果来自Goldreich、Goldwasser和Micali [GGM86]的工作，他们构建了伪随机*functions*。这是一个所有函数都易于计算的函数族，但任何随机的一个函数都无法被任何能够查询其任何（自适应选择的）输入序列的（有效）观察者与真正随机的函数区分开来.[20]伪随机函数是一种极其强大的密码学原语。除了在各种密码学设置中的明显效用外，我还提到两种其他应用。这种构造的一个后果在第5.2.4节中提到——电路下界*natural proofs*的结果对于所有单向函数的相当有效的逆运算至关重要（例如，比目前所知的分解算法要好得多）。这可能部分解释了我们无法证明电路下界的原因。第二个后果与计算学习理论相关。那里的一个中心问题是：哪些函数类可以从自适应观察中有效地学习？（这种学习概念捕捉了，例如，儿童概念的学习，以及通过进化发展的视觉系统）。根据定义，伪随机函数不能以这种方式学习，因为它们在任何新输入处的值对（有效的）学习者来说看起来是完全随机的。这个观察导致以下远非显而易见的事实：*Some functions are easy to compute but hard to learn*。

NW-generators 随机算法通过假设困难性进行去随机化的能力为计算复杂性提供了一个新玩具（实际上变得很受欢迎）：对于弱计算模型，我们 *do* 有下界，这些下界可能被转化为针对它们的无条件伪随机生成器，并且可以无条件地去随机化相应的概率类别。一个自然的候选者是 *constant- depth circuits*，其中已知指数下界，由 Håstad [Has86] 提出（例如，对于奇偶函数）。

[19] We will soon contrast this with another construction designed specifically for de-randomization.

[20] This may be viewed as a pseudo-random generator with *exponentially* long output (namely, the truth table of the function), every bit of which is efficiently computable, and which the adversary can query in arbitrary locations.

第一个直接使用这个困难函数来对这类电路进行伪随机性攻击的是Ajtai和Wigderson在[AW85]中。然而，他们将困难性转化为伪随机性的转换相当复杂，参数也相当弱.[21]几年后，Nisan [Nis91b] 提出了一种更加优雅和直接的设计（如下所述），这是一种基于奇偶性和其已知困难性的无条件伪随机生成器，同样适用于同一类弱电路。

这个新设计反过来激发了Nisan和Wigderson [NW94] 开发一个新的 *conditional* 生成器，有时称为NW-生成器。他们主要的动机是削弱去随机化如 $\mathcal{BPP}$ 类所需的硬度假设。如上所述，BMY-生成器假设存在单射函数来实现这一点。如果没有单射函数，这对密码学是一个致命打击，但它对复杂性世界的影响不大；例如，它保留了 $\mathcal{NP}$-完备问题（并且困难）。事实上，NW-生成器 *can* 利用这种（不那么结构化）的硬度。[NW94] 展示了即使在假设（平均情况）硬度的情况下，例如 $\mathcal{NP}$-完备问题的硬度，如何去随机化 $\mathcal{BPP}$。实际上，[NW94] 证明了更强烈的陈述。一个小的电路无法逼近的 *Any* 函数，具有 *exponential time* 算法，产生一个伪随机生成器。此外，从（困难）函数构造（伪随机）生成器是一种通用的、*black-box* 构造.[22] 如何从给定的函数 f 构造 NW_f 生成器的示意图如图19所示。

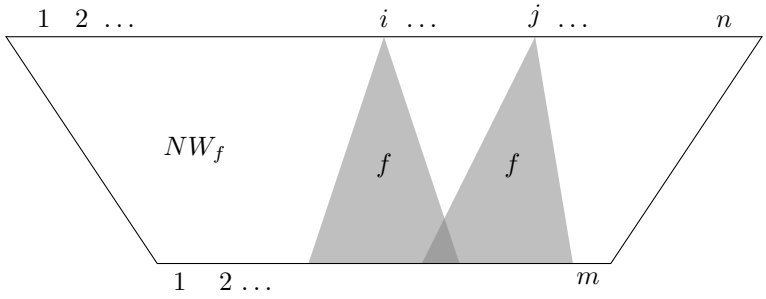


图19. NW_f的示意图。本质上，n输出是通过将f应用于n不同子序列（具有小的成对重叠）得到的，这些子序列是m位长输入序列。

这个一般结果在函数假设的难度和结果生成器的质量之间提供了一个权衡。在这里，我只陈述一个极端的参数选择，以与定理7.14进行对比。请注意，在硬度假设上的巨大放松是以生成器的运行时间为代价的（其拉伸和质量不变）。正如我们马上将看到的，这不会对去随机化产生影响。

Theorem 7.15 [NW94]. *The following are equivalent statements:*

- *There are exponential-time computable functions that cannot be approximated by polynomial-size circuits.*
- *There exists an* $\exp(m)$*-time computable* $(\mathcal{P}/poly, 1/\text{poly}(n))$*-generator* $G : \{0,1\}^m \to \{0,1\}^n$ *for any* $n = \text{poly}(m)$.

此外，使用无法由指数规模电路近似的函数实例化 NW-生成器，会产生一个指数扩展的生成器。事实证明，尽管生成器的运行时间呈指数增长，但该生成器仍能对 $\mathcal{BPP}$ 进行非平凡的去随机化。允许这一观察结果的关键是，在使用PRGs进行去随机化时，无论如何都会枚举所有 2^m 个可能的“种子”。

[21] Despite this, that type of transformation was resurrected and strengthened 30 years later, together with many new ideas, to obtain much improved parameters and new de-randomization applications. Some of this evolution is in the sequence of papers [GMR+12, CHHL18, RT18], with the last paper demonstrating the power of pseudo-randomness in completely different areas; in this case, quantum computing, the subject of Chapter 11.

[22] This notion means that the construction accesses the function simply by requesting its values on different inputs and is completely independent of the way it might be computed.

生成器。所以我们不妨“免费”允许，硬函数可在2^m-时间内可计算（而不是，例如，要求它在$\mathcal{P}$内），而不减慢整体确定性模拟时间。

与BMY生成器相比，NW生成器的主要弱点是它需要的计算时间比测试其输出伪随机性的对手多*more*。因此，对于大多数加密应用来说，它是无用的。然而，出于去随机化的目的，我们可以承受它，因为生成器只比对手多项式慢，例如，对于$\mathcal{BPP}$，仍然会在多项式时间内运行。与BMY生成器相比，NW生成器有几个优点。其中最主要的是，它允许看似更可信的硬度假设（例如，对于$\mathcal{NP}$的函数完全性和甚至更高的复杂性类）。另一个优点是，在生成器中使用困难函数的通用方式——生成器的输出位仅仅是困难函数在许多不同、精心选择的输入位子集上的评估。这种通用构造立即使我们能够去随机化几乎*any*合理的概率算法类（在其他计算模型和其他资源限制下），假设对于它们对应的电路类有一个困难函数。

西北发电机也发现了远不那么明显的应用和扩展，例如在 *randomness extraction* [Tre99]（第9章中提及），算术复杂性 [KI04]，概率可验证证明（PCPs）[HW03] 和学习理论 [CIKK16]。

“最优”硬度与随机性权衡，Impagliazzo和Wigderson [IW97] 第7.7定理，基于NW生成器，但仍需要相当多的工作。我只提到它的一方面。该定理的假设是一个标准的、最坏情况下的复杂性假设，而原始的BMY-和NW生成器都使用了平均情况假设。这两种假设之间的转换，称为*hardness amplification*，将任何最坏情况下的困难函数转换为平均困难函数，然后进一步将其转换为另一个，其输出在随机输入上基本上是不可预测的。硬度放大的主要工具是*arithmetization*[23]和*XOR lemma* (（参见历史、几个证明和调查[GNW11]））。获得这种转换的最优参数需要对这种硬度放大结果进行“去随机化”。这样做意味着为完全不同的目的构建新的伪随机生成器，而不是欺骗算法。这说明了像伪随机性这样的概念是如何演变并获得自己生命的，这是一个具有许多化身的现象。这种伪随机生成器在[IW97]中被设计出来，并导致了第7.7定理。

7.3.3 Final high-level words on de-randomization

Black-box vs. white-box de-randomization 截至目前，我们已描述了对去随机化问题的“黑盒”攻击。即，一个伪随机生成器应同时去随机化一类中的 *all* 概率算法，甚至不需要查看它们，只需使用它们的输入输出行为即可。正如人们所怀疑的，这种非常一般的伪随机性攻击路线是一种受益于专业化的范例。我们看到，为了去随机化一个概率算法，我们只需要一种方法来有效地生成一个低熵分布，当然是在每个输入上。对于一个 *given* 算法，这可能比同时欺骗 *all* 算法要容易得多。事实上，这种“白盒”方法，即仔细分析一些重要的概率算法及其使用随机性的方式，已经使得通过定制生成器使它们确定性，*without* 任何未经证明的困难性假设。这些成功故事（其中最引人注目的是Agrawal、Kayal和Saxena [AKS04]最近的不确定性素性测试和Reingold [Rei08]的无向图连通性的对数空间算法）实际上表明，概率算法随后去随机化作为 *deterministic* 算法设计的一个范例。在Motwani和Raghavan的教科书中 [MR95] 可以找到更多此类特定算法去随机化的基本示例。然而，请注意，对于某些问题，我们甚至不能期望通过尝试去随机化特定算法来消除困难性假设！Kabanets和Impagliazzo [KI04]的显著结果表明，即使去随机化命题7.2中体现的简单概率算法也意味着某些非平凡的电路下界。因此，硬度和随机性比任何人预期的都要紧密地交织在一起。实际上，

[23]This term refers to the encoding of Boolean functions as polynomials, an idea developed in the areas of program testing and interactive proofs (some mentioned in Section 10.1)

对于去随机化概率的 *proofs*,[24] 相关问题 [IKW02] 表明，看到算法没有帮助：白盒去随机化等同于黑盒去随机化。

Uniform de-randomization NW-生成器被设计用来去随机化概率 *non-uniform* 电路。因此，它们预计需要以 *non-uniform* 电路下界的形式来保证困难性。但也许去随机化 *uniform*，概率算法要容易得多（并且需要更弱的假设）？这有两个答案。一方面，在标准 *uniform* 困难性假设下（本质上，即 $\mathcal{EXP} \neq \mathcal{BPP}$），可以实现某种（平均意义上的）均匀去随机化，如 [IW98, TV02][25] 所示。另一方面，对于 *promise problems*[26]，即使均匀去随机化也需要电路下界，如 Tell [Tel18] 所证明（参见那里导致这一结果的有意思的思想发展）。

Using randomness despite de-randomization 随机性在计算机科学的许多领域中仍然是不可或缺的，包括概率证明、学习理论、密码学和分布式计算，这些内容我们将在本书的后续章节中遇到。此外，即使 $\mathcal{BPP} = \mathcal{P}$，概率算法可能更简单、更快。例如，最著名的概率素性测试算法所需时间少于 $O(n^3)$，而最著名的确定性算法需要时间 $\Omega(n^5)$。鉴于所有这些应用，值得思考：这些应用中要求的完美随机性在现实世界中是否存在？如果没有它，哪些应用将幸存？我们将在第9章回到这些问题。

[24] We shall meet many types of these in Chapter 10.
[25] Interestingly, this result still uses the (seemingly non-uniform) NW-generator, but in a more sophisticated way.
[26] Which care only about the answer to some, but not all, problem instances.

8 Abstract pseudo-randomness

在上一章中，我们看到了*computational pseudo-randomness*这一概念对于理解概率算法的强大之处至关重要。事实上，这一概念及其变体，以及更一般的概念*computational indistinguishability*，对计算复杂性的许多方面产生了巨大影响，包括电路复杂性、计算学习理论，当然还有密码学，它构成了大多数定义和结果的基础。

现在让我们将这个概念大大扩展；实际上，将其推向逻辑结论。我们不再考虑从*computationally bounded*观察者族中分布的不可区分性，而是简单地允许任意的观察者族。如果这些观察者无法区分具有该属性的随机对象和整个宇宙中的随机对象，我们将称（任何）对象宇宙的性质相对于任何这样的族为“伪随机”。也许令人惊讶的是，这将会变得极其有用！

伪随机性在这个广义上是一个计算理论数学之间不断扩大的互动领域。在本章中，我从（不同的）两个领域的观点出发，提出了对其研究动机，给出了许多伪随机性质的例子，在完全普遍性上定义了这个概念，并提出了明确展示和认证伪随机对象的问题。我们将看到，包括黎曼猜想和$\mathcal{P}$与$\mathcal{NP}$问题在内的两个领域的核心问题，可以非常自然地用伪随机性的语言来表达。然后，我们将讨论新兴的“结构对伪随机性”的证明技术，它已经在多个领域得到应用，但其力量才刚刚开始展现。

让我们从一些简单的伪随机性质例子开始。这些例子也将有助于突出一个相关主题，即关于优秀的文本，参见Alon和Spencer[AS00]）。这种方法证明了具有所需性质的物体的*existence*，而不需要明确描述它们。这项技术首次在20世纪40年代末的两篇独立论文中使用。一篇是Erdős [Erd47]的，证明了拉姆齐图的存在（并开启了现代*Ramsey theory*），另一篇是Shannon [Sha48]的，证明了良好纠错码的存在（并开启了*information theory*）。Erdős的更多例子进一步阐明了这种技术的力量。这种方法的核心是（巧妙地）选择一个物体集合U，并证明在U中的*random*物体以正概率具有所需性质（实际上，在大多数情况下，几乎肯定）。请注意，每个概率算法的正确性证明本质上都具有这种结构——证明几乎所有的硬币翻转选择都将导致算法得到正确的结果。

8.1 Motivating examples

我们查看伪随机属性的三个例子。对于每个例子，考虑以下将占据我们注意力的挑战——明确展示具有这些属性的对象。

我们从 Ramsey 理论中的两个例子[1]开始（参见 Graham，Rothschild 和 Spencer [GRS90]）。这个领域的典型元定理是，在 *every* “足够大”的系统中，必须存在一个非常结构化的非平凡部分。定量方面是这个子系统可以有多大。

Ramsey graphs. 一个包含20个顶点的图被称为“r-Ramsey”，如果它不包含任何团和大小为r的独立集。换句话说，每个包含r个顶点的子集至少有一对通过边相连的顶点，以及另一对不相邻的顶点。Erd s [Erd47] 证明了包含26个顶点的*almost every*图是(3 log n)-Ramsey.[2]

Weak tournaments. 一个 *tournament* 是一个有向图，其中每对顶点之间都存在一条有向边（例如，描述真实锦标赛中成对比赛的胜者）。如果每个其他玩家都至少输给了他们中的一个，则顶点集（玩家）被称为 *dominating*。在 n 个顶点上的锦标赛被称为 w-弱，如果它不包含大小为 w 的支配集。Erd s 证明了*almost every*锦标赛是（$\frac{1}{3}\log n$)-弱.[3]

[1]While combinatorial in nature, the origins of both examples is in first-order logic, the first leading to a decidability result in Ramsey's original paper [Ram30], and the other to extension theorems and 0/1-laws.

[2]This result is nearly tight. No graph can be $(\frac{1}{2}\log n)$-Ramsey.

[3]This result is nearly tight. No tournament can be $(\log n)$-weak.

对于下一个例子，我们转向编码理论，它是我们处理数字通信和媒体存储中噪声能力的基础。

Good codes. 一个维度为（比如说）$n/10$的$\mathbb{F}_2^n$上的子空间V是一个*distance-d*线性码，如果V中的任意两个向量在至少d个坐标上不同。遵循香农，Varshamov [Var57] 通过一个简单的概率论论证证明了这样的子空间是一个距离-$n/10$线性码.[4]

8.2 General pseudo-random properties and finding hay in haystacks

让我们从上面的例子中抽象出来。我们有一个有限的物体集合 U。在上面的例子中，这些物体分别是图、锦标赛或子空间。一个 *property* 简单地是一个子集 $S \subseteq U$，一个元素 (具有属性 $x \in U$ 当且仅当它属于 S)。我们三个例子中的属性分别是 r-拉姆齐、w-弱或距离-d。在所有三种情况下，相关参数的特定选择定义了在相应宇宙中的 *almost every* 物体的属性。请注意，在所有例子中，每个属性都是由许多“基本测试”组成的集合定义的，这些测试可以看作是一组观察者。对于 U (中的随机 [5] 物体，每个这样的测试都以极高的概率满足，并且通常所有这些测试都满足的证明来自于一个并集界限)。在前两个例子中，每个这样的测试只涉及顶点的一个小子集，而在第三个例子中，每个测试涉及一对向量。这种局部性、简单性或“低复杂性”的基本测试，它衡量一个物体看起来有多“随机”，在下面考虑的许多其他例子中将是典型的。

我们将一般地说，如果属性 S 包含 *almost all* 个 U 的元素，则称其为 *pseudo-random* 。这个名称的由来很简单，因为 U 的一个随机元素几乎必然满足 S 。

Definition 8.1 (伪 - 随机属性)**.** 属性 $S \subset U$ 是 ϵ-伪随机 m 如果 $|S| \geq (1-\epsilon)|U|$。

所以伪随机属性只是一个大集合。特别是，如果你需要一个具有特定伪随机属性的对象，很清楚该怎么做——只需在U *at random*中选取一个对象，它将以高概率具有所需的属性。当我们想要*deterministically*获得一个对象时，事情变得有趣。在许多情况下，甚至难以证明给定的对象具有该属性。像往常一样，我们将从渐近角度思考；U和S将是族，$U = \{U_n\}$和$S = \{S_n\}$。此外，正如上面的例子所示，U_n通常具有以n为指数大小的规模，因此暴力搜索是不可行的（而任何特定对象的描述只需poly(n)位）。ϵ的值，通常未指定，可以取为一个小常数，或者（如上面的例子所示）一个当n增加时趋于0的函数$\epsilon(n)$。

This general setting turns out to capture a host of problems in a surprising variety of areas in mathematics and in the theory of computation. 在两个领域，对于各种伪随机属性 $S \subseteq U$，都会出现同样的问题。一个（在数学上自然的）对象 $x_0 \in U$ 是否满足 S? 我们能否有效地描述 x 中的某些对象 S? 霍华德·卡尔罗夫巧妙地传达了这种挑战的本质，他将其描述为 **finding hay in a haystack!** 在许多这些问题中，尽管有大量的干草，已知的有效算法可能只能产生 *needles*。然而，这场寻找干草的探索是非常有回报的!

在以下章节中，我们将通过一系列示例来建立这种伪随机现象在数学和计算机科学中的普遍性和重要性。在此之前，让我们回顾上述三个伪随机性质的示例（按逆序），检查它们是否符合一般框架，并考虑这些问题对它们的状态。在所有三个示例中，相应宇宙中的对象可以用大约 n^2 比特进行编码（因此这些宇宙的大小为 $\exp(n^2)$)。

Good codes. 香农的开创性论文[Sha48]留下了如何明确描述良好码的问题。在这个草堆中寻找干草的复杂性在于，在实践中，我们不仅希望有一个好的线性码，还需要为其提供高效的编码和解码算法。对这些问题的研究创立了编码理论领域。使用这些方法首次高效地构造了良好的线性码

[4] This result is nearly tight. No such subspace has distance $n/3$.

[5] We implicitly use the uniform distribution over the set U. Much of the theory and many applications also work with other, and sometimes all, choices of distribution over U. Indeed, in some cases, this applies also to infinite universes endowed with appropriate probability measures.

额外属性由Forney [For66] 和Justesen [Jus72] 提供。现在我们有许多构建各种显式、高效码的替代方法（例如，参见教科书[Rot06, GRS16]）。为了看到这种设置符合我们的符号，请注意，一个维度为$n/10$的子空间可以通过一组基来描述，因此需要$O(n^2)$位，宇宙U_n是所有这些的集合，属性$S_n \subseteq U_n$包括所有距离为$n/10$的子空间，即所有好的线性码。

Weak tournaments. 这个例子也有一个快乐的结局。在 n 个顶点的完全有向图可以用 $O(n^2)$ 比特来描述，U_n 是所有这些锦标赛的集合，S_n 是 w-弱锦标赛，对于 $w = \frac{1}{3}\log n$。在这个设置中，一个明确的伪随机对象，我们将称之为 *Paley tournament*，由 Graham 和 Spencer [GS71] 提出，基于 Paley [Pal33] 的构造。其描述极其简单。为了简单起见，假设 $n = p$ 是一个素数，$p \equiv 3 \bmod 4$，并且让顶点是 $\mathbb{F}_p$ 的元素。回忆一下，$\chi(k)$ 表示在 $\mathbb{F}_p^*$ 上的二次特征函数，如果 k 是该域中的一个平方，则 $\chi(k)$ 为 1，否则为 -1。现在对于任何一对顶点 i, j，如果 $\chi(i-j) = 1$（这是对于像 $\chi(k) = -\chi(-k)$ 这样的素数的一个良好定义的锦标赛），则从 i 到 j 之间直接连接一条边。这里涉及的所有计算都很简单，因此 Paley 锦标赛可以在多项式时间内构造。

虽然构造简单，但分析使用了深层次的结果：Weil关于有限域上曲线有理点数估计[Wei49]。它所使用的形式，即一个*exponential sum*界限，本身就是伪随机性的典型陈述，它将自身作为伪随机对象展示出二次特征χ。我们将在第8.3节进一步讨论它，该节讨论黎曼猜想。

Ramsey graphs. 这里，通往良好显式构造的旅程已经超过70年。让我总结一下已知的内容。首先，正如之前一样，n顶点图有$O(n^2)$位表示，U_n是它们的集合。我们定义S_n为(3 log n)-Ramsey的性质。展示这样一个图仍然是一个难以捉摸的问题。因此，寻找具有较弱参数的伪随机对象是自然的，即对于较大的r (值，r-Ramsey图。当然，可以正式定义S_n^r为r-Ramsey的性质，它们对于$r \geq 3\log n$)都是伪随机的。即使对于小的$\alpha > 0$构造n^α-Ramsey图也不是微不足道的，Frankl和Wilson [FW81] 给出了一个美丽的构造，该构造对于$r = \exp(\sqrt{\log n})$是r-Ramsey的。改进来自一个非常不同的来源。第9章中讨论的中心伪随机对象理论*randomness extractors*产生了一系列非常不同（且不太优雅）的显式构造，首先由[BKS+05, BRSW12]（基于算术组合数学）提出，然后是[CZ16]的突破（参见[Coh17]），它使用了一种非常不同且复杂的组合方法。这些具有显著更好的参数；当前最好的结果是$r = \exp(\log\log n)^{O(1)}$，非常接近（并且正在下降）到最优界限。

8.3 The Riemann hypothesis

黎曼猜想，可能是数学中最著名的未解问题，其形式化表述非同寻常。通常的表述涉及 *zeta function*，这是一个需要一些高级、专业知识的复杂对象。在这里，我们提出了另一种（已知）的表述，使用伪随机性的语言，这种表述简单易懂，甚至对高中生来说也容易表述和解释。

首先，让我们正式讨论*drunkard's walk* (，也称为整数上的随机游走)。假设有一个酒吧在0点，喝了几杯啤酒后，一个醉汉开始随机地上下街道行走。更确切地说，当占据一个整数i时，醉汉以概率$i+$移动到$i-1$，以概率$\frac{1}{3}$移动到$i-1$，并以概率i停留在$\frac{1}{3}$。他在n步后离酒吧有多远？这可以表述为估计一系列n独立、无偏的随机$\{-1, 0, 1\}$变量的和。这是一个标准的计算，可以证明他将以高概率在酒吧$O(\sqrt{n})$距离内。

这表明了一个伪随机属性。宇宙是 $U_n = \{-1, 0, 1\}^n$，由所有可能的 n-walks 组成。定义一个 walk $z \in U_n$ 为 *d-homebound*，如果它最终在酒吧 d 的范围内，即 $|\sum_i z_i| \leq d$。正如提到的，对于任何 $d = d(n)$-homebound 来说，这是一个伪随机属性。我们能否找到一个具有这种属性的显式序列？当然可以；有很多简单的，比如全 0 序列，或者任何 1 和 -1 数量相等的序列。这里有趣的问题是，是否任何自然

数学序列具有典型的随机序列所具有的“平方根”消去性质。这里有一个著名的自然序列。

Definition 8.2 (Möbius序列)**.** 定义（无限）*Möbius sequence* $\mu = \mu(k)$ 对于每一个自然数 k 如下：$\mu(k)$ 如果 k 有一个平方因子，则为 0。否则，它为 -1 或 1，分别取决于 k 是否有奇数个或偶数个质因子。定义 μ_n 为 μ 的前 n 个符号。

如果我们让醉汉按照莫比乌斯序列的指示行进，这个序列是否总是会在d处d围绕$\sqrt{n}$保持d-定向？关于这个简单定义的序列的简单问题等价于黎曼猜想。更确切地说，Mertens [Mer97] 证明了以下定理。

Theorem 8.3 [Mer97]. *The Riemann hypothesis is true if and only if, for every $\delta > 0$, the sequence μ_n is $n^{\frac{1}{2}+\delta}$-homebound.*

当然，黎曼猜想尚未被证明成立。因此，自然地降低伪随机性要求（正如我们在拉姆齐图中所做的那样），并要求例如有 {v*} 非平凡消去。也就是说，{v*} 至少是 {v*}-homebound 吗？这个看似无辜的问题也证明等价于一个已知的陈述，在这种情况下是一个定理而不是一个猜想。也就是说，它等价于哈达玛德和德·拉·瓦莱-普桑著名的素数定理，该定理确定了小于整数 {v*} 的素数个数的渐近行为为 {v*} ln {v*}。

Theorem 8.4 (素数定理)**.** *The sequence μ_n is $o(n)$-homebound.*

Möbius序列与抛硬币序列有多接近？嗯，它是当然的确定性，因此完全可以预测。它的符号可以通过图灵机依次计算。等价地，Möbius序列与由某个确定性图灵机产生的序列完美相关。现在，按照本章的精神，将其置于比所有可计算序列更小的测试类别中是有意义的。让我们从底部开始。素数定理（定理8.4）可以解释为Möbius序列与绝对最简单的确定性序列，即常数序列1, 1, 1, . . .的相关性趋近于零。那么交替序列1, −1, 1, −1, 1 . . .呢？或者一个由有限自动机或实时图灵机产生的序列呢？[6]Sarnak的一个大胆猜想是，Möbius函数与由零熵率动力系统生成的序列*every*的相关性趋近于零。[7]这个猜想的一些非常一般的情况已经被证明（参见Bourgain，Sarnak和Ziegler [BSZ13] 以获得精确的定义和历史概述）。

虽然Möbius序列的平方根消去仍然是一个数学上的主要问题，但这种消去在其他重要定理中的其他重要序列中已被证明。让我们用一个例子来说明这一点，这个例子是我们需要的“弱锦标赛”示例中的定理。Weil定理[Wei49]的一个结果是以下*exponential sum*关于二次特征的χ界限。

Theorem 8.5 [Wei49]. *For every prime p, every degree $d > 0$, and every nonsquare polynomial $f \in \mathbb{F}_p[x]$ of degree d,*

$$\left| \sum_{x \in \mathbb{F}_p} \chi(f(x)) \right| \leq d\sqrt{p}.$$

因此，对于所有这些低次多项式“测试”，二次特征看起来就像一系列抛硬币的结果一样随机，至少从局部有界的角度来看。类似的结果也适用于其他特征。更重要的是，从Deligne关于有限域上代数簇的著名黎曼猜想[Del74, Del80]中可以得出这个定理的*multivariate*多项式推广。

指数和及其相关估计遍布数论、分析和遍历理论，也可以从这种伪随机性角度来观察。虽然不清楚这个角度是否足够强大以证明新的此类结果，但这种联系对各种应用极为丰富。例如，定理8.5在[AGHP92]中用于去随机化；在[BGW99]中用于下界；正如我们上面所看到的，

[6]Such a machine must output a symbol at every computational step.

[7]This is a considerably stronger model than a deterministic real-time Turing machine—it may be viewed as a probabilistic real-time Turing machine with access to $o(n)$ random coins before it outputs the nth symbol.

在[GS71]中对于弱锦标赛构造。一些伪随机对象（如二次特征）具有惊人的广泛适用性，我们将在下面的扩展器和提取器中看到对此的更强证明。

8.4 $\mathcal{P}$ vs. $\mathcal{NP}$

如何将 $\mathcal{P}$ 与 $\mathcal{NP}$ 的问题与伪随机性联系起来？简短的回答如下：几乎所有函数都难以计算；SAT 是否难以计算？这与我们之前关于黎曼猜想的通用框架完美类比：几乎所有序列都是自归约的；Möbius 序列是否自归约？进一步阐述这个类比，将序列的宇宙替换为布尔函数的宇宙，将自归约的伪随机性质替换为计算困难的伪随机性质，将 Möbius 序列替换为 SAT 函数，你就将黎曼猜想替换为 $\mathcal{P}$ 与 $\mathcal{NP}$ 的问题，作为一个伪随机性问题。我们希望读者现在能够将这个类比应用于其他设置：你所需要的只是三个要素，宇宙 U，一个伪随机（= 大）性质 $S \subset U$，以及一个元素 $x \in U$，其属于 S (的成员资格，即其对该性质的伪随机性 S) 是有疑问的。如前所述，这种设置，无论是明显还是隐蔽地，捕捉了许多不同的数学和计算机科学问题。

让我们回到 $\mathcal{P}$ 与 $\mathcal{NP}$ 的比较，并稍微正式一些，因为使这个例子符合我们的伪随机性框架需要特定的参数设置方式。正如我们将看到的，这种对 $\mathcal{P}$ 与 $\mathcal{NP}$ 的观点也将解释伪随机性对于我们在5.2.4节中遇到的“自然证明”下界障碍的重要性。

为了与本节中的符号一致，在这里将布尔函数的输入大小称为 k 将是有用的，因此我们考虑函数 $f : \{0,1\}^k \to \{0,1\}$。定义另一个所有函数 $f = \{f_k\}$ 可在 exp(k) 时间内计算的可计算复杂度类 $\mathcal{ECP}$ 也将很方便。

我们的宇宙 U_n 将是所有布尔函数，其 *truth table* 取 n 位；这些是布尔函数 $f : \{0,1\}^k \to \{0,1\}$，具有 $n = 2^k$。观察一下 SAT 的真值表，或者更确切地说，任何 $\mathcal{ECP}$ 中的函数，都可以在 exp(k) = poly(n) 时间内生成。现在，根据定理 5.6，几乎所有 U_n 中的函数都需要至少 $2^{k/3} = n^{1/3}$ 的大电路。因此，一个 k-位函数 f 的属性，要求任何 $h(n) \leq n^{1/3}$ 至少 h 的电路大小，是伪随机的。称此属性为“是 h-难”。根据定义，如果 SAT (或 $\mathcal{NP}$) 中的任何问题对于任何 $h(n) \gg (\log n)^{O(1)} = \text{poly}(k)$ 都是 h-难的，那么它将立即意味着 $\mathcal{NP} \not\subseteq \mathcal{P}/\text{poly}$，解决这本书中最重要的未解问题！在证明 SAT 或另一个显式函数之前，一个更容易的任务是有效地生成这样的伪随机 h-难函数。这也是复杂性理论的一个重要挑战，原因如下。观察一下，如果我们有一个在 poly(n) 时间内生成（给定 n）这样一个 h-难函数 f 的真值表的算法，那么这意味着 $f \in \mathcal{ECP}$。因此，这样的算法，对于 $h(n) \gg (\log n)^{O(1)} = \text{poly}(k)$，将意味着以下更弱但非常重要的猜想。

Conjecture 8.6. $\mathcal{ECP} \not\subseteq \mathcal{P}/$多边形。

总结来说，伪随机性可以自然地捕捉证明猜想电路大小下界。有趣的是，相同的伪随机性概念也可以解释我们在证明电路下界（例如，解决猜想8.6）中的困难。如第5.2.4节所述，如果分解k-位整数需要电路大小exp(k^c)，那么猜想8.6没有*natural proof*。现在我们已经发展了适当的伪随机性语言，我们可以更好地解释Razborov和Rudich [RR97]在第5.2.4节中提出的这个令人惊讶的结果。让我们探索伪随机性与证明计算难度困难之间的关系。

自然证明（关于电路下界）证明了布尔函数的计算难度，并且按照定义，对其中 *almost all* 进行了证明。换句话说，拥有自然证明是一种伪随机属性。但是，通过Goldreich、Goldwasser和Micali [GGM86]的重要结果（在7.14定理陈述之后讨论），假设的分解难度使得可以构造 *efficiently computable* 伪随机函数。这些函数按照定义“看起来”像随机函数，并且特别满足这种伪随机属性。然而，通过构造，这些函数实际上很容易计算！这种矛盾是证明电路下界挑战的核心，甚至超出了特定的

自然证明的概念。下界，隐含或显含地，必须找出一个（或标准）来区分困难随机函数和简单伪随机函数，尽管在非常强的意义上，这两个集合似乎是不可区分的。Razborov-Rudich自然证明的概念是一组（或标准）度量，对于这样的下界证明者来说是无用的（至少假设整数分解是困难的，或者更一般地说，如果存在任何单向函数）。

8.5 Computational pseudo-randomness and de-randomization

在这个部分，我们将看到本章的抽象伪随机性框架足够通用，可以捕捉到第7.2节中讨论的具体*computational*伪随机性（你可能需要回顾）。因此，我们可以将这个框架所捕捉到的重大未解问题清单中添加$\mathcal{BPP} = \mathcal{P}$问题。毕竟，“在干草堆里找草”*efficiently*的探索正是去随机化的问题。然而，明确指出这个通用框架中计算伪随机性的某些挑战和结果将是有用的。

首先，考虑特定概率算法的去随机化。设$A(x, y)$为一个确定性算法，对于每个输入x，在随机输入y的情况下，以高概率导致A输出正确答案（对于输入x上的某个固定函数）。我们希望找到针对同一问题的确定性算法。将此任务放入我们的通用框架中很简单。对于每个输入x，我们有一个宇宙U_x，包含所有可能的随机序列y。对于每个序列，我们有一个伪随机属性$S_x \subseteq U_x$，引导算法在输入x时给出正确输出。因为算法A以高概率成功，S_x包含U_x的大部分元素。对于每个给定的x，有效地找到这样的伪随机$y \in S_x$将去随机化算法A。重要的是要注意，针对同一问题的不同算法会导致不同的“干草堆”，以及不同的伪随机“干草”概念来寻找；这些去随机化任务中的一些可能比其他任务更容易。

这是一个这种方法的一个巨大的成功故事，即Agrawal、Kayal和Saxena的确定性素性检验算法[AKS04]。事实上，这篇论文通过去随机化一个特定的概率素性算法，即Agrawal和Biswas的算法[AB03]，解决了这样一个搜索问题。请注意，这个概率算法的设计是出于希望上述搜索问题可能有一个确定性算法（而且确实有！）。从这个算法产生的稻草堆的性质太复杂，无法在这里描述（我们将在第13.1节中给出更多细节）。但即使更早之前，素性检验提供了另一个稻草堆，这个稻草堆更容易描述（并且也更容易去随机化）*under a number-theoretic assumption*，正如我们现在要讨论的。事实上，高效素性检验的故事始于一个*deterministic*算法。1976年，Miller [Mil76]给出了一个有效的确定性素性检验*assuming*扩展黎曼猜想。在我们的通用框架中，并逆转历史，Miller的算法可以看作是对（后来的）Rabin的概率素性检验[Rab80]的去随机化，Rabin实际上设计了这个检验来消除Miller算法背后的扩展黎曼猜想假设.[8] Rabin算法的搜索问题更容易描述（我们将作弊，并讨论一个更简单的问题）。设x为一个整数。稻草堆U_x可以看作是所有模x的数。在这个集合中的数y是好的（即，在S_x中），如果它有雅可比符号-1。至少一半的y是好的（即，将导致Miller算法的正确输出）。Miller观察到，使用Ankeny [Ank52]的一个数论结果，必须在第一个y个整数中有一个好的n^2，其中n是x的二进制长度。因此，通过所有小整数进行简单搜索就可以在确定性多项式时间内解决搜索问题（当然，假设扩展黎曼猜想）。换句话说，第一个n^2个整数的集合Y（当考虑模x时）是一个完美的样本：它将包含每个U_x的n-位输入x的一个好的y。

接下来，我们将我们的通用框架重新表述，以捕捉第7.2节中$\mathcal{BPP}$的去随机化结果（这确实需要硬度假设）。目标是同时欺骗*every*高效的概率算法A在*every*可能的输入x上。为此，让我们设计一个高效的伪随机生成器，对我们这里的目的是通过其图像来捕捉：一个多项式小的n位序列集合Y（例如，$Y = \{y_1, y_2, \ldots, y_t\}$具有$t \leq \text{poly}(n)$），这是一个“几乎完美的样本集”。也就是说，均匀的

[8] Solovay and Strassen [SS77] designed a somewhat different probabilistic primality test earlier, independently of Miller and Rabin.

分布在此样本 Y 上是计算伪随机：它看起来像 *all* 序列上的均匀分布，对每个小电路.[9] 对于任何算法 A 和输入 x 的去伪随机化过程将简单地对 $A(x, y_i)$ 进行多数投票 $y_i \in Y$。Y的伪随机性质保证其大多数元素导致 A 输出正确答案，因此多数投票总是正确的。

让我们更精确地用具体参数在我们的通用框架中阐述这种方法的$\mathcal{BPP} = \mathcal{P}$?问题成分。$t = n^4$。

- 宇宙 U_n 是 $\{0,1\}^n$ 的所有 t 子集的集合。
- 伪随机子集 S_n 包含所有满足在 Y 上的均匀分布是伪随机的 $Y \in U_n$。[10]
- 集合 S_n 几乎包含 U_n 中的每一个 Y，因此确实是伪随机的。这可以通过标准计数论证来确立。
- T他相关的搜索问题是：给定 n，找到一个元素 S_n.
- 如果搜索问题有一个多项式时间算法 poly(n)，那么 $\mathcal{BPP} = \mathcal{P}$。

7.2节一般去随机化结果的核心是设计这样一个高效的算法来解决这个搜索问题，假设存在足够困难的函数。该算法构建的伪随机集Y是这样一个困难函数为基础的高效伪随机生成器的像。

是否有可能在无条件的情况下获得这样的通用去随机化结果？对于多项式时间电路，高效的伪随机生成器意味着我们目前无法证明的电路下界（在本节术语中，*tests* 集合过于强大）。因此，在这里，需要假设困难性以实现伪随机性。但遵循这一范例导致复杂性理论家考虑了比多项式大小电路更弱的大量有趣的计算模型。这些模型计算更少的函数，因此这个测试集更小，这使得伪随机属性更大，找到伪随机对象的可能性也更容易。这个想法极为富有成效，并导致了适用于广泛重要类别的 *unconditional* 伪随机生成器，包括内存受限算法、常深度电路和低次多项式。

我们以一个著名的例子来结束本文，这个例子将在第14章再次讨论。考虑使用很少内存的概率算法——它们在输入长度上是 *logarithmic* 的。一个相关的例子是执行图上随机游走的算法；它只需要在每一个阶段记住它当前占据的顶点的名称，其二进制长度是整个图大小的对数。现在让我们考虑这些算法可以对其随机输入进行哪些类型的测试。计数是一种很容易由有限内存算法执行的计算类型。因此，它们可以执行统计教科书中出现的许多标准统计测试，并在许多科学实验中使用。通常，这些测试计算序列中的各种小模式，并检查它们是否大致以随机序列的方式分布。即使是针对特定统计测试的（几乎完美）样本集的设计，也属于统计学中的实验设计领域。如果我们旨在欺骗所有有限内存算法，我们特别询问：能否同时无条件地欺骗 *all* 这些测试？令人惊讶的是，Nisan [Nis92] 在开创性工作中提供了一个明确的肯定答案，他设计了一个美丽的低内存伪随机生成器，可以对抗所有这些算法。此生成器使用略高于多项式时间（更精确的陈述见第14章关于空间复杂度）。

虽然解释这些重要结果超出了本文的范围，但请注意，在这里也存在一个导致硬度的来源，这暗示了伪随机性，一旦它被适当封装。Nisan的洞察[11]是，通过低通信的双方协议封装由小型内存算法执行的伪随机性测试是一个好方法（因此，主要资源，通信，实际上是信息论上的，而不是计算上的）。更精确地说，需要建立计算（甚至近似）

[9] These circuits capture the computation of all efficient algorithms A on all inputs x of a given length.

[10] In the sense of Chapter 7.2, with respect to all n^3-size circuits. Namely, every such set Y is a nearly-perfect sample set for every algorithm A running in time n^2 using n random bits.

[11] Which was sparked by a homework problem in a complexity class of Umesh Vazirani that he took at Berkeley.

某些关于两个参数 $h(x,y)$ 的函数需要在两个当事人之间进行大量通信，其中一方持有 x，另一方持有 y。虽然这很容易证明，但将其转换为伪随机生成器则需要更多的想法和努力！Nisan 生成器的这种观点在 [INW94] 中得到解释和扩展。研究这类下界问题的领域被称为 *communication complexity*，在第 15 章中讨论。

8.6 Quasi-random graphs

在这一节和下一节中，我们研究图。本节重点研究“密集”图（具有二次边数）和下一节研究“稀疏”图（具有线性边数）。*quasi-random graphs* 的理论起源于 Thomason [Tho87] 的论文（他实际上称它们为“伪随机图”）和 Chung、Graham 和 Wilson [CGW89]（他们的“准随机图”术语得以保留）。这是最早对伪随机性质进行全面研究的例子之一，并阐明了一些我们尚未涉及的问题，特别是此类性质的可减少性和完备性。

随机图及其性质的研究始于Erd s和Rényi的开创性论文[ER59, ER60]，并成为了一个庞大的研究领域。回想一下，在n个顶点上的随机图是通过让每对顶点之间以概率$\frac{1}{2}$存在边来定义的。换句话说，它是在集合上的均匀分布，我们自然将其命名为U_n，即所有n个顶点的无向图。[12]考虑几个不同的性质，所有这些性质都很容易证明是准随机的（即对于几乎所有图$G \in U_n$都成立）。请注意，前三个涉及到图的不同方面G；第一个计算大子集中边的数量G，第二个计算小“模式”图的出现次数G，第三个计算一个代数性质——G的邻接矩阵的顶特征值。总的来说，这些参数需要接近随机图中的期望值，随着n的增长而趋于一致。

S_1 : 对于顶点子集 *every* 的 $T \subset [n]$，T 中的边数为 $|T|^2/4 \pm o(n^2)$。S_2 : 对于固定的 *every* 图 *labeled*，H 中 H 的诱导副本数为 $(1 \pm o(1))n^v 2^{-\binom{v}{2}}$，其中 v 是 H 中的顶点数。S_3 : 图 G 的邻接矩阵的最大两个特征值满足 $\lambda_1 = (1 \pm o(1))n/2$ 和 $\lambda_2 = o(n)$。S_4 : G 中的边数为 $(1 \pm o(1))n^2/4$，G 中的 4-重数为 $(1 \pm o(1))n^4/16$ 。

Chung、Graham和Wilson的论文[CGW89]证明了以下惊人的陈述：所有四个性质都是等价的。

Theorem 8.7 [CGW89]。*If a (large enough) graph satisfies any one of these properties, it satisfies them all.*

确实，任何满足这些性质之一的图，也满足该论文中研究的许多其他性质。这表明对于伪随机性有一个关于 *completeness* 的概念，这确实在最后一个性质 S_4 中得到了最强大的体现。请注意，S_4 只测试图的两个参数，而 S_2 则测试这些参数以及无限多个其他参数。虽然很明显，满足 S_2 的图也满足 S_4，但令人惊讶的事实是，逆命题也成立。因此，给定大型图中边和 4-环出现的统计数据决定了（直到可忽略的误差项）每个有限子图的统计数据！

该论文[CGW89]还研究了哪些特定图在上述意义上是伪随机的。读者发现一个答案是规范示例，佩利图，可能不会感到惊讶。这实际上是我们在8.2节中看到的佩利锦标赛的变体，其中顶点数是一个素数$n = p$，但这次$p \equiv \mod 4$等于1，当且仅当i和j之间有边（这是有定义的，因为对于这样的p，我们有$\chi(k) = \chi(-k)$）。为了证明其伪随机性，只需对最简单的

[12] To have a fully consistent notation, we could have named this set $U_{\binom{n}{2}}$, as the bits describing this distribution are the edges of the graph. But I expect no confusion should arise.

属性，即为 S_4。并且这个属性成立（再次使用Weil定理8.5），因为每对顶点有 $n/4 \pm O(\sqrt{n})$ 个共同邻居，从而得出计数。

当然，可以研究其他类别的随机图及其在不同分布下的性质。有趣的是，还可以反向进行：从（将是）伪随机性质开始，并由此发展出随机图（或其他对象）的类别。一个典型的例子，发展成为重要的理论如下。取任何图序列 $G = \{G_n\}$，其中所有统计量在 S_2 中以适当自然的方式收敛（即，图 G_n 在极限上共享所有有限图 H 的发生频率）。然后这个序列产生了一个随机图模型（称为 $graphon$），它极大地推广了Erd s-Rényi理论，并可以从其中以自然的方式抽取任何大小的图（我们在这里不考虑）。在这个模型中，具有这些特定子图统计量的性质是一个伪随机性质！这一新兴的 $graph\ limits$ 理论非常令人兴奋。它通过 $property\ testing$ (领域与算法和编码理论相联系，通过Goldreich [Gol10])，并通过研究配分函数和吉布斯分布与统计物理相联系。最后，它允许在极值图理论的组合领域进行分析和使用分析工具（在经典意义上，如极限、收敛、紧致性等）。鼓励读者进一步了解（参见Lovász的书籍 [Lov12]）。

8.7 Expanders

扩容图可能是最通用的伪随机对象。它们在计算理论的几乎所有领域都扮演着关键角色：算法、数据结构、电路复杂性、去随机化、纠错码、网络设计等（实际上，它们出现在本书的许多章节中）。在数学中，它们以根本的方式触及分析、几何、拓扑、代数、数论以及当然的图论的不同子领域。确实，扩容图是复杂性理论与数学之间的重要桥梁；第13章致力于这些相互作用，描述了四个不同领域，在这些领域中图扩张起着关键作用（参见第13.3节、第13.4节、第13.5节、第13.6节）。

珍贵的非平凡数学对象很少能拥有类似的影响！Hoory、Linial和Wigderson的专著[HLW06]是关于扩张器的全面文本。然而，这个领域非常活跃，自它出版以来已经发生了许多事情；这里总结了其中一些令人兴奋的最近发展。

扩展图是稀疏图。因此，我们将我们的宇宙 {v*} 视为所有在 n 个顶点上的 d-正则图（即每个顶点都由 d 条边接触），其中 d 是对所有 n 相同的固定常数。这里是一些这些对象的自然伪随机属性。我非正式地陈述它们。再次注意，它们似乎完全不同，一个表达了组合/几何属性，第二个表达了代数属性，第三个表达了概率属性，一个 d-正则图的属性。

S_1 : 对于每个 t 和顶点子集 $T \subseteq [n]$ 的大小为 $|T| = t$，T 和其补集之间的边数大致为 $dt(n-t)/n$。S_2 : G 的邻接矩阵的所有非平凡特征值（绝对值）都远离第一个，即 d。S_3 : 在 G 上的自然随机游走经过 $O(\log n)$ 步（以指数速率）收敛到均匀分布。

这是检查每个这些属性对于几乎每个 d-正则图都成立的标准。同样，这里有一个令人惊讶的结果，即这三个属性都是等价的：如果任何图有一个，它就有其他（这是一个定义是 $basic$ 的明确迹象！）等价性是通过一系列1980年代的工作（S_1 和 S_2 之间的联系是黎曼流形中重要的 $Cheeger\ inequality$ 的离散类似物 [Che70]）而知的，这些工作具有不同未指定参数之间的特定定量关系。

一个图如果是这个意义上的伪随机图，则它是一个扩张图。首先定义扩张图并证明其存在性（通过概率方法）的是 Pinsker [Pin73]。能否显式构造扩张图？在第8.6节中，出现了一个伪随机图：佩利图。在当前的稀疏设置中，

构建远不那么明显。显式构造扩张器的问题吸引了来自许多不同领域的学者和技术。第一种显式构造归功于Margulis [Mar73]，他使用了“Kazhdan性质(T)”。如今，我们已知有各种方法，代数和组合的，来构造扩张器（主要方法列在该节末尾）。精确的定义、不同的构造和许多应用出现在专著[HLW06]中。更多应用在Wigderson [Wig10a]的综述演讲中有所概述。

让我们详细说明一种结构、一个开放问题和一种应用。这些应该会诱使读者了解更多关于这些非凡对象的信息。

An explicit family of expanders 设 p 为一个素数，考虑一个 3-正则图 G_p，其顶点为 $\mathbb{F}_p$ 的元素，并将每个顶点与其前驱、后继和逆元相连。换句话说，每个 x 通过一条边与 $x-1, x+1$ 相连，而 x^{-1}（作为 0 没有逆元，我们可以将其连接到自身）。以下定理来自 Sarnak [Sar90] 的第 3.3 节。

Theorem 8.8 [Sar90]. *The family G_p is a family of expanders.*

虽然这些图本身描述简单，但它们的扩展证明使用了非常复杂的工具（在这个领域很常见）。这是从Cayley图[13]在$SL(2,p)$上的扩展以及标准生成元，以及图G_p是这些群对射影线通过Möbius变换作用的Schreier图的事实得出的。$SL(2,p)$的扩展最初是从塞尔伯格著名的数论3/16-定理[Sel65]中推导出来的。Bourgain和Gamburd [BG08]给出了这个扩展的不同证明，使用了算术组合数学。

让我们观察这些图 G_p 的明确程度。如果 p 是一个 n-位整数，我们可以用 n-位序列来表示 $\mathbb{F}_p$ 的元素。邻域结构如此简单，以至于给定一个顶点 x，可以在 poly(n)-时间内计算出它的 3 个邻居。因此，我们有一个指数级大小的图，通过邻居算法具有如此简洁的描述。这种明确程度 [14] 在我们提出的应用中至关重要，并且实际上在许多其他应用中也至关重要。我们给出了一种方便的描述，说明了这些构造可以有多明确和高效。这里选择的扩张器的参数不是最通用的，但它们很方便，并且适用于大多数应用。[15]

Theorem 8.9. *For every constant c, there is a constant d and a 多(n) time[16] algorithm A, such that for every integer n, there is a d-regular graph G_N on $N = 2^n$ vertices[17] with the following properties.*

- **Explicitness:** *On inputs n and $x \in \{0,1\}^n$, A outputs the d neighbors of x in G_N.*
- **Eigenvalue expansion:** *All nontrivial eigenvalues of G_N are bounded above in absolute value by $d/2$.*
- **Vertex expansion:** *Every set $S \subseteq \{0,1\}^n$ of size $s \leq o(N/d)$ has at least cs neighbors in G_N.*

关于参数最优性的注释将引导我们到达我们的开放问题。对于扩展的特征值定义，c和d之间的最优关系已知并可实现。根据Alon-Boppana定理[Alo86]（参见[Nil91]中的证明），$c \geq^{2}\sqrt{d-}^{1}- o(^{1})$。这是通过Lubotzky、Phillips和Sarnak[LPS88]以及Margulis[Mar88]的显式拉马努金图实现的，它们满足$c \geq^{2}\sqrt{d-}$^{1}。然而，对于*vertex expansion*，随机图满足“无损扩展”[18] $c \geq (1 - o(1))d$，而Kahale[Kah95]已知的最优显式构造仅达到“仅”$c \geq (\frac{1}{2} - o(1))d$。显式实现$c > d/$^{2}的问题已经开放了20年。最好的部分结果是*bipartite*图的构造，

[13] The vertices of Cayley graphs are all elements of a given group, and two vertices are connected if their ratio belongs to a given set of generators.

[14] Actually, efficiently obtaining large primes p may be difficult, as we mentioned in Section 7.1, and so the description here is not fully explicit. However there are ways around this problem (which we don't discuss) that result in fully explicit graphs with the same properties.

[15] Some applications demand even more stringent explicitness and efficiency, and it is actually remarkable how cheaply these complex pseudo-random graphs can be generated — see e.g., [VW18].

[16] Indeed, even logarithmic space.

[17] We pick a power of two so as to label vertices by binary sequences, but one can pick any other integer.

[18] They have essentially as many neighbors as possible, as this number cannot exceed sd.

无损扩展在一个方向上[CRVW02]，这是一个已经足够超越特征值扩展所能达到的应用的属性。

Open Problem 8.10. 构建显式的无损扩展器（例如，使用 $c >$） .9d）。

An application of expanders to randomness-efficient error reduction 对于我们的应用，考虑以下问题，可以称为 *deterministic error reduction*。你有一个概率算法 A，你希望在输入 x 上运行。这需要 n 个随机比特，这正是你所拥有的。问题是算法保证的错误率，比如说，1/10，对你来说太高了。所以你希望降低它。我们在第7.1节中的概率算法中表明，错误可以很容易地降低（例如，对于任何 k 降低到 exp($-k$)）。想法是每次用独立随机性在 A 上运行 x 次，然后取答案的大多数投票。然而，这需要 kn 个随机比特，你却没有。仅使用你拥有的 n 个随机比特，能否降低错误？Karp、Pippenger 和 Sipser [KPS85] 给出了一个美丽的肯定答案，这是扩张器最早的应用之一。只需使用你的随机比特在上述图 G_p 中产生一个随机顶点 x。[19] 然后考虑所有距离 $d = O(\log n)$ 的顶点 x，其中每个都是 n-比特序列。使用它们中的每一个作为运行 A 在 x 时的随机性，并且像以前一样，计算答案的大多数投票。请注意，找到所有这些顶点的过程是高效的——只需重复应用在 G_p 中计算邻居的算法，以生成所有 $3^d = \text{poly}(n)$ 路径在 G_p 中，并取它们的端点。不那么明显，但由 G_p 的扩张性质得出的是，这个算法的错误将降低到任何 n^{-c} (的选择——c 的选择决定了 d) 定义中的常数。

Methods of proving expansion 扩展不仅是一个在数学和计算机科学中基本且广泛应用的概念，而且来源极其多样。到目前为止，我们已经拥有了惊人的丰富方法来显式（和非显式）地构建扩张图，每种方法都有其自身的优点和后果。特别是，这些方法为Lubotzky和Weiss [LW92]提出的广泛挑战提供了全面（尽管还不完整）的理解：找出哪些有限群，以及哪些生成集，会产生扩张的凯莱图。我对每种方法都说几句话，这可能会激发读者进一步深入探索。

- “母群”方法，由玛古利斯发起[Mar73]。在这里，有限扩张器的族由凯莱图组成，而底层群都是单个无限群的商。这个母群的性质决定了商的扩张。这种方法导致了上述提到的[LPS88, Mar88]中的特征值最优的*Ramanujan expanders*。

- “有界生成”方法，由Shalom [Sha99] 提出。它（与许多其他想法一起）导致了Kassabov、Lubotzky和Nikolov [KLN06] 的一个非常一般的定理，即每个{v*}非阿贝尔{v*}有限单群都有一个固定的生成元集合，使得凯莱图是扩张的。

- “之字形”积的方法，由Reingold、Vadhan和Wigderson [RVW02] 提出。这种方法迭代地从固定一个构造越来越大、越来越大的扩张器。这种方法与群中的半直接积 [ALW01] 的联系导致了某些非常非简单群的Cayley扩张器 [MW04, RSW04]。上述无损二部扩张器的构造 [CRVW02] 基于之字形方法。它还导致了计算复杂性的突破：Reingold的证明 [Rei08]，即 $\mathcal{SL} = \mathcal{L}$ (参见第14节)，这又反过来启发了Dinur的PCP定理组合证明 [Din07]（参见第10.3节）。

- “算术组合数学”方法，由Bourgain和Gamburd [BG08] 提出。在这里，展开图又是凯莱图。展开（以及其他重要思想）源于在群积下集合的增长，始于Helfgott [Hel08]，这又基于“和积定理”

[19] Assume here for simplicity that $p = 2^n$ (which is, of course, impossible). The case $p \neq 2^n$ can be handled as well with some care.

[20] The missing Suzuki group was added to complete this list in [BGT10].

在Bourgain、Katz和Tao的有限域中[BKT04]）。这种方法适用于证明有限秩的*all*简单线性群中的扩张，对于几乎每一对生成元（与其他方法中特别选择的生成元相比）[BGGT13]。这种强大的方法也是单位群中的扩张[BG10]、*monotone expanders*和*dimension expanders*的显式构造[BY13]以及*affine sieve*[BGS10]等应用的基础。

- “提升”方法，由Bilu和Linial [BL06] 提出。同样，这是一种组合迭代方法，通过提升从较小的扩张器生成较大的扩张器。通过Marcus、Spielman和Srivastava [MSS13a] 的“交错多项式”进行的最优分析，这种方法的建设性提升导致了全新的拉马努金扩张器（在[Coh16]中实现建设性），它还产生了其他重大后果，最值得注意的是解决了Kadison-Singer猜想，这在第13.3节中讨论。

Variations and extensions 上述讨论的扩展定义可能是研究最广泛、使用最频繁的，但只是众多有用且有趣定义中的一种。我们简要讨论另外两种，它们是热门研究课题，产生了美丽的结构、算法和应用理论。

第一个是 *small set expansion*。上面的定义主要关注大集合的扩展。但图（参见[HLW06]）的扩展 *profile*，即每个给定大小的子集的边和顶点边界，是自然要研究的，并且预期较小的集合具有更大的扩展。小集合扩展的算法和复杂度理论重要性，特别是在理解近似难度方面，开始在其对UGC（唯一游戏猜想）的研究中显现出来，这些研究在4.3.2节中讨论。特别是，对“2-2游戏”的某种较弱猜想的解决，关键取决于对 *Grassmann graph* (看见 [KMS18] 和其中的参考文献) 的小集合扩展的理解。

第二是 *high-dimensional expanders*。Linial 和 Meshulam [LM06] 以及 Gromov [Gro10] 独立启动的一项工作，定义并研究了图之外的扩展，在更高维的单纯复形中。明确构建这些对象的研究将代数、几何和拓扑美妙地联系起来。它已经发现了与诸如性质测试和量子纠错码等计算领域的联系和应用。该快速发展的领域的介绍是 [Lub14]。更近期的构造和应用出现在例如 [EK16, DK17, KO18, DHK+19] 中。

8.8 Structure vs. pseudo-randomness

本节仅揭示了不断增长的冰山的一角，其中*pseudo-randomness*及其与*structure* (的相互作用，均定义为适合场合)，成为数学和计算机科学众多领域中的一个非常强大的“元证明技术”。Tao [Tao06] 的一项美丽调查详细解释了这一技术在整数算术级数工作的序列中的存在：Roth定理、Szemerédi定理、Szemerédi正则引理、Furstenberg的遍历理论证明、Gowers的定量界限以及关于素数级数的Green-Tao定理。此外，它阐明了“结构对伪随机性”*dichotomy*定理对各种数学对象的需求和存在。Tao在[Tao08]的第3.1章中给出了更多其他领域的应用，包括数论、偏微分方程、遍历理论和图论。其他各种二分定理的计算来源还包括（在第8.5节中提到的）设计针对弱计算模型的伪随机生成器的尝试，其中一些我们将在下面提到。

让我们从一个一般设置开始，它可以专门化到上面讨论的许多例子中。设X为一个有限集，并让我们的宇宙U是X上的所有有界函数；具体来说，所有函数$f\colon X \to [-1,1]$。例如，当U是n顶点上的所有图时，那么X将是所有对$i \neq j \in [n]$的集合，一个图G由以下这样的函数f表示：$f(i,j) = 1$如果(i,j)是G的边，否则$f(i,j) = -1$。请注意，允许范围$[-1,1]$实际上允许我们考虑“带权重的图”，或者等价地，图的凸组合。

对于任何两个函数 $f, g \in U$，当底层的 X 分布是均匀时，简单地定义它们的 *correlation* 为 $\langle f, g\rangle = \mathbb{E}_{x\in X}[f(x)g(x)]$。我们将使用相关性来定义伪随机性。一个伪-

随机属性将由 {v*} 中的测试函数族 $\mathcal{F}$ 在 U 中如下定义。伪随机函数将是那些几乎与 $\mathcal{F}$ 中的每个测试函数正交的函数。更确切地说，如果对于每个 $f \in \mathcal{F}$，$|\langle f, g\rangle| \leq \epsilon$，则称函数 $g \in U$ 为“$(\epsilon, \mathcal{F})$-伪随机”。这种定义伪随机性的机制非常通用。事实上，前几节中列出的大多数伪随机属性都可以用这种方式表达。例如，第8.7节中扩张器的属性 S_1 是通过对于每个顶点的子集 T，取 T 和其补集之间顶点对的指示函数，并从其中减去随机图中该大小集合的期望值来获得的。因此，在这个例子中，每个集合 T 都产生一个测试函数。

以下定理是结构（或“简单性”）与伪随机性之间期望的*dichotomy*定理的一种基本类型的模板。

Theorem 8.11 (模板二分定理)**.** *Let U and $\mathcal{F}$ be as above. Then every function $g \in U$ can be decomposed as*

$$g = s + e,$$

where s is a "simple" function, and e (for "error") is a pseudo-random function, both with respect to F.

More precisely, there is a real function m such that for every $\epsilon > 0$, there is such decomposition with the following properties. First, the function e will be $(\epsilon, \mathcal{F})$-pseudo-random. Second, the function s is composed from at most a finite number $m(\epsilon)$, depending only on ϵ, of functions from $\mathcal{F}$. Namely, $s = h(f_1, f_2, \ldots, f_m)$ with $m \leq m(\epsilon)$, $f_i \in \mathcal{F}$, and h is some (combining) function.

让我们看看这样的二分定理如何有助于证明关于 *all* 对象在 U 中的陈述。基本上，它允许我们分别处理简单对象和伪随机对象（在统计意义上的简单）。这听起来很天真，确实，我们以天真、高层次的方式描述它，但它应该能让人感受到上述一些强大定理的证明方式。所以，假设你想证明关于 U 的某个普遍陈述，即它包含的 *every* 对象具有某种期望的性质。例如，你可能想证明 Szemerédi 定理：对于每个固定的 $\delta > 0$，以及每个整数 k，*every* 是前 n 个整数（在 δ, k 的意义上足够大）的测度 δ 的子集，必须包含一个 k-项算术进步。在这个例子中 U_n 是 $[n]$ 的所有 δ-密集子集的集合。

第一步是理解为什么来自 U 的 *random* 对象满足所需的属性。寻找满足这一条件的充分条件可能表明伪随机“测试函数” $\mathcal{F}$，这反过来又根据二分法定理，暗示了简单的“结构”是什么。在算术级数示例中，很容易看出，在期望中，具有测度 δ 的 $[n]$ 的随机子集将包含大量的 k-项级数，大约有 $\delta^k n^2$ 个。因此，所选的伪随机属性可以自然地尝试强制执行这些统计数据。Roth 的定理 [Rot53] 关于整数密集子集中的3项算术级数，该定理启发了整个发展，正是这样做的。Roth 观察到，如果子集与任何周期函数的相关性很小，则此类级数的统计数据是成立的，因此他将 $\mathbb{Z}_n$ 的字符作为他的伪随机性测试族。对于更大的 k，Gowers [Gow 01] 发明了他的 *Gowers norms*，伪随机性测试，这些测试类似于在随机子集中强制执行 k-项级数的统计数据。当然，困难的部分是正确选择伪随机测试函数——在结构化和随机外观部分之间取得平衡——以证明每个（及其总和）都具有所需的属性。

让我们回到二分定理本身，并尝试理解我们何时可以证明这样的定理。首先，让我们阐述一个非常基本的定理，实际上，使用线性字符。这非常简单，可以将其视为关于离散傅里叶变换的大学数学作业问题。设 $X = \mathbb{F}_2^n$ 和 U 是 X 上所有函数的集合。对偶群 $\hat{X}$ 有 2^n 个字符，χ_T，每个子集 T 都有一个。我们将这个集合视为我们的测试函数集，即 $\mathcal{F} = \hat{X}$。在这个设置中，给出一个形式结构vs.伪随机性定理，按照定理8.11的模板，是很容易的。回想一下，$\mathcal{F}$ 中的函数构成了 $\mathbb{R}^{2^n}$ 的一个正交归一基。因此，每个函数 $g \in U$ 在这个基中都有唯一的表示，即 $g = \sum_T c_T \chi_T$，其中系数 c_T 被称为 *Fourier coefficients*，并通过 $c_T = \langle \chi_T, g\rangle$ 计算。现在分解本身就很明显。给定 $\epsilon > 0$，如果 $|c_T| \geq \epsilon$ 则称 T *large*，否则称 *small*，并将简单部分定义为 $s = \sum_{T \text{ large}} c_T \chi_T$，伪随机部分定义为

[21] Which is very similar to attempts to de-randomize particular probabilistic algorithms, where we seek sufficient conditions on properties of the random input which will cause the algorithm to give the correct answer.

$e = \sum_T$ 小 $c_T \chi_T$。显然，$g = s + e$。根据 *small* 的定义和字符的正交性，函数 e 是 $(\epsilon, \mathcal{F})$ 伪随机的。以下是对 s 的简单性的论证。g 的范数 $\langle g, g \rangle$ 为 1，因此 Parseval 的恒等式意味着 $\sum_{T \text{ large}} (c_T)^2 \leq 1$。由于每个 $|c_T|$ 至少是 ϵ，因此在简单部分中不可能有更多的 $m(\epsilon) = \epsilon^{-2}$ 函数。请注意，这里的组合函数 h 在 s 中非常简单且高效（即线性组合）。

应该清楚，这里的傅里叶特征并没有什么特殊之处——我们使用的只是 $\mathcal{F}$ 中函数的 *orthogonality*。换句话说，当 $\mathcal{F}$ 是 U 的正交归一基时，模板二分定理成立。这似乎，实际上也是一个极其简单的情况。我们期望这样的二分定理在多大程度上成立呢？嗯，在 *full* 通用性上！令人惊讶的是，*every* 的 X 和 $\mathcal{F}$ 选择提供了这样的分解！它的许多特殊情况都出现了，包括在 Szemerédi 的正则性引理中，Green-Tao 在素数算术级数上的工作，以及一般情况（但形式不同）在 [TZ08, RTTV08] 中。以下版本给出了最佳参数，归功于 Trevisan，Tulsiani 和 Vadhan [TTV09]。

Theorem 8.12 [TTV09]. *The template dichotomy theorem holds for* 每 *choice of X and $\mathcal{F}$. Moreover, the bound $m(\epsilon) = O(\epsilon^{-2})$, and the combining function h uses at most ϵ^{-2} simple operations: addition, multiplication, and threshold.*

证明基本上是贪婪的，大致如下。从常数的零函数开始，分阶段构建（简单的）函数 s 来逼近给定的 g。如果当前的 $g - s$ 与 $\mathcal{F}$ 中的 *all* 函数的相关性低于 ϵ，则完成。如果不，并且 $g - s$ 与某些成员 $f \in \mathcal{F}$ 至少有 ϵ 的相关性，那么我们向 s（添加一个适当的）常数倍 f。最后，使用一个简单的势函数来限制迭代次数。这个强大的想法及其变体在许多算法和其他应用领域（通常在“提升”或“乘性权重更新”的名称下）中找到了应用（通常在伪随机性之外）。本书后面讨论了两种这样的（相关）应用，第16章用于在线预测，第17章用于增强学习算法的质量。关于这个元算法的优秀综述是 [AHK12]。

让我们演示一下二分法定理在计算伪随机性中的一个（间接）应用。*Every*布尔函数可以通过一个易于计算的函数“近似”，在它们的对称差是计算伪随机性的意义上。

Corollary 8.13. *Let U to be the set of all Boolean functions on n bits. Also, for some fixed c, let $\epsilon = n^{-c}$ and let $\mathcal{F}$ be the set of all n^c-size circuits on n input bits.*

For every Boolean function g, we have $g = s \oplus e$, where $s \in \mathcal{P}/$多, and e is computationally pseudo-random—no function in $\mathcal{F}$ has correlation $\geq \epsilon$ with e.

特殊情况下的一般二分定理，与之前以类似方式证明的定理，实际上是其动机来源，这些定理是在研究具有不同动机的各种对象时出现的，包括图论中的“弱正则引理” [FK96] Frieze 和 Kannan，用于素数进展的“稠密模型定理” [TZ08] Tao 和 Ziegler，以及来自计算伪随机的“核心集”定理 [Imp95a] Impagliazzo。Trevisan、Tulsiani 和 Vadhan 的论文 [TTV09] 提供了三个相当不同的证明，代表了不同的起源；一个使用来自计算学习理论的“提升”技术，一个使用博弈论中的最小-最大定理，一个使用 Szemerédi 的递归细化论证 {v*}。

如上所述，在随机性和结构之间，存在大量二分（或“分解”）定理，在其他设置中，这些定理在形式上可能不同，但本质上是相同的。以下列出了最近一些证明此类结果数学对象，其中一些比上面的模板复杂得多，但在某些情况下，例如下面列出的第一项，情况甚至更简单，因为每个对象本身要么是有结构的，要么是伪随机的：

- 有限域上的有界度多项式 [GT09, KL08]。
- 有界度多项式在高斯变量中 [Kan12, DS13]。
- 有界灵敏度的布尔函数 [Hat10]。

- 有界度多项式阈值函数 [DSTW14]。
- 超图 [RS06]。
- Gowers范数的逆定理 [GTZ12, Sze12]。

9 Weak random sources and randomness extractors

概率算法和许多其他随机性的应用都是在假设可以访问无限的 *independent, unbiased* 比特的情况下进行分析的。现实真的提供了这样的完美随机性吗？假设自然界是决定性的，完美随机性根本不存在。即使如此，第7.2节证明了可信的 *hardness assumptions* 意味着 $\mathcal{BPP} = \mathcal{P}$ (，即每个概率算法都可以有效地去随机化)，因此所有这些算法应用在决定性世界中仍然存在。但是，如果我们想要 *unconditional* 结果。关于自然的最小假设是什么将提供相同的算法应用？

一个关于自然的合理折中方案，似乎得到了经验的支撑，即即使它不能为我们提供完美的随机比特，许多其过程在某种程度上是*unpredictable*。这包括天气、股市、互联网流量、太阳黑子、放射性衰变、量子效应以及许多其他可以用于随机性的过程。实际上，许多计算机系统正是以这种方式生成用于各种算法所需的随机比特。从这样的过程中采样会产生可能*correlated, biased*的随机比特流。[1]这引发了一些明显的问题。对于这种*weak*随机性，一个好的数学模型是什么？我们能否将其用于需要完美随机性的应用？如何？

三十年研究产生了一种美丽的理论，回答了这些问题以及其他问题。这些发展在[Nis96, Sha04, Vad11]中进行了详细调查，我在下面简要总结。首先，我们讨论弱随机源的形式数学模型。然后，我描述本节的主角 *randomness extractor*，这是一个确定性的算法，其作用是将任何弱随机源中的随机样本“净化”为来自均匀分布的完美（或近似完美）样本（这反过来又可用于应用）。
As it happens, randomness extractors, born for the this purpose, turned out to be useful, even essential, in a variety of diverse (theoretical and practical) application areas, including error-correcting codes, data structures, algorithms, de-randomization, cryptography and quantum computing.[2]有趣的是，在这些应用中，随机性作为资源或动机完全不存在。尽管如此，提取器的*pseudo-random*属性使它们适用于许多其他需求，就像第8章中扩展开图的情况一样。对于所有这些应用，我们需要*efficiently computable*提取器，我将描述这种高效提取器的显式构造。请注意，这里讨论的提取器有时被称为*seeded extractors*，以区分它们与一个更受限制的近亲*deterministic extractors*。后者仅涉及受限制的源系列，正如另一个相关研究领域——*data compression* [ZL78]所做的那样。在这里，我们仅调查最一般类别的弱随机源的工作，这类源具有一些熵，但在其他方面没有结构限制。

为了了解弱源可能的样子以及纯化的挑战，这里有一些例子。热身：如果你不知道偏差，只知道它在范围[0.1, 0.9]内，你会如何纯化一系列*independent*次相同偏置硬币的投掷？具体来说，找到一个确定性算法，将此类源中的n位序列转换为另一个（可能更短）序列，该序列是均匀分布的（尝试自己完成，或查看脚注）。[3]以下是一些“弱”源的其他例子。首先，考虑一个与上述相同的独立硬币投掷序列，但每次投掷在这个范围内可能有一个不同的未知偏差。接下来，考虑一个由对手逐位生成的序列，该对手根据过去的投掷结果，从这个范围内选择一个偏差，并用这个偏差投掷下一个硬币。最后，考虑一个n位序列，其中对手在（例如）$n/10$个选择的位置投掷独立的、无偏置的硬币，然后使用结果来确定剩余位的位置。

请注意，虽然此类分布确实可能来自对弱自然源的采样，但它们也可能

[1] Note that even though quantum mechanics predicts that the measurements of (say) successive photon spins yield perfectly random bits, the physical devices generating and measuring these will not be perfect.

[2] A comprehensive survey of these amazingly diverse applications of extractors is sadly missing. One can base a fantastic graduate course on the extractor application areas which can be found in the papers [RR99, TSZ04, Vad04, Zuc06, Vad09, Wig09, GZ11, MS16, Li17] and the references therein.

[3] The idea, going back to von Neumann, is simple. Pair up the bits of the input sequence, and consider them left to right. Ignore the pairs 00 and 11. For each pair 01 output, say heads and for each 10 output, call it tails. Note that each output symbol is independent of the others and has probability $\frac{1}{2}$.

在实用计算场景中产生。例如，如果你为加密目的选择一个完全随机的 n-位密钥，但其中一部分泄露给对手（可能是每个比特的小偏差，或者可能是 $n/10$ 的未知子集的值），则你的密钥的条件分布是上述提到的类型。在构建空间有界计算的伪随机生成器（在第14章中讨论）中，会出现一个完全不同的场景，导致类似的分布。这些情况说明了为什么随机提取器的使用超出了其最初的动机，如上所述。

从如此弱源中提取纯随机性的意义是什么？所选的净化方法应该适用于类别中的*every*分布——我使用“*adversary*”这个词来强调分布是按照净化方法选择的。请注意，上述所有概率分布的熵[4]都非常大：长度n的恒定比例。然而，如何使用这一事实远非明确，因为净化者不知道这个熵“隐藏”在哪里。重要的是要强调，净化者只从未知分布中获得*single* n-位样本——自然界或对手都不会为我们提供几个独立的样本。

9.1 Min-entropy and randomness extractors

9.1.1 Min-entropy: Formalizing weak random sources

约翰·冯·诺伊曼可能是第一个在20世纪40年代提出如何使用不完美随机源的问题的人。冯·诺伊曼实际上需要完美的随机位来进行他在“IAS机器”上的蒙特卡洛模拟（最早的计算机之一）。我在上面的预热示例中解释了他的解决方案。在20世纪80年代，从[Blu86]开始，出现了一系列弱随机源的不同模型（以及“提高其质量”的方法），重要的是[SV86, CG88, VV85]。Sipser [Sip88]给出了研究弱源复杂性理论动机。最后，Zuckerman [Zuc90]给出了我们可能希望纯化的分布的最终定义；一个弱随机源简单地被建模为一个*arbitrary*概率分布，在$\{0,1\}^n$上，其中包含一些k的熵。结果证明，正确的熵概念是下面定义的*min-entropy*，而不是经典的香农熵.[5]最小熵仅仅是概率分布的L_∞范数的对数.[6]

Definition 9.1 (*Min-entropy*)**.** 设 D 是在 $\{0,1\}^n$ 上的一个概率分布，令 D_x 表示序列 $x \in \{0,1\}^n$ 的概率。*min-entropy* 的 D，记为 $H_\infty(D)$，是在所有 $x \in \{0,1\}^n$ 中 $-\log_2 D_x$ 的最小值。

Definition 9.2 (k-*source*)**.** 我们称 D 为 k-*source*，如果 $H_\infty(D) \geq k$，即如果每个序列 x 在 D 中出现的概率最多为 2^{-k}。

有教育意义的是让自己相信上述提到的所有四个弱源都是 k-源对于 $k = \Omega(n)$。考虑一个 k-源 D 的便捷方式，这在后续中并未失去一般性，就是将其简单地视为在某个（未知）大小至少为 2^k 的子集 $S \subset \{0,1\}^n$ 上的均匀分布。

当然，我们继续以渐近的方式思考，因此在讨论固定 n 和 k 时，我们实际上考虑的是分布的集合 $D = D_n$，并允许最小熵 $k = k(n)$ 依赖于 n (，例如，为 $\sqrt{n}$)。此外，纯化算法在 n 方面也必须高效。

9.1.2 Extractors: Formalizing the purification of randomness

净化算法被称为 *randomness extractor*，或者简称为 *extractor*。让我们尝试一个简单的公式，观察其缺陷，然后修正它以成为正确的定义。因为提取器必须是

[4]For now the reader can think about entropy informally as “randomness content” or formally as Shannon’s entropy. We will soon define the notion of entropy that is actually relevant to this setting.

[5]The reason for this choice is that there are distributions with extremely high Shannon entropy in which a single sequence has high probability, and this makes randomness extraction impossible.

[6]This notion appeared first in this context of randomness extraction (but was used to define a more restricted class of sources) in [CG88].

确定性的情况下，自然地考虑一个函数 $f:\{0,1\}^n \to \{0,1\}^r$ 是一个 (n,k)-提取器，如果对于 *every* k-源 D 在 n 位上，$f(D)$ 是 *statistically close* 到 U_r，r 位的均匀分布。换句话说，我们有 L_1 距离 $|f(D)-U_r|_1$ 至多为 ϵ，在本节中最好取为 $\epsilon = 1/\text{poly}(n)$（尽管对于某些应用，一个小常数就足够了）。

显然，我们必须有 $r \le k$，因为确定性过程不能增加熵.[7] 不幸的是，这样的函数 f 简单地不存在。即使在极端情况下 $k = n-1$（即，熵几乎是所有东西），它们也不存在，无论如何，我们只是在尝试提取一个比特，即 $r = 1$。原因是对于布尔函数 f，至少有一个 $f^{-1}(0)$ 或 $f^{-1}(1)$ 的大小不超过 2^{n-1}。因此，让 D 是两个较大集合上的均匀分布，并注意它至少有 $n-1$ 的最小熵。然而，分布 $f(D)$ 是常数，并且在统计上尽可能远离 1 比特的均匀分布。

随机提取器的正确定义，即在它们存在并且仍然有用（如我们将看到的）用于纯化的目的，是由Nisan和Zuckerman [NZ96]给出的。它允许使用不止一个函数 f，而是许多函数，并要求其中大多数能够纯化任何给定的源。

Definition 9.3 (*Extractor*)**.** 一个具有 f_i 的函数序列 $F = (f_1, f_2, \ldots, f_t)$：$\{0,1\}^n \to \{0,1\}^r$ 被称为 $(n,k) - extractor$，如果对于每个 k-源 D，除了 ϵ-分数之外的所有 f_i 都满足 $|f_i(D) - U_r|_1 \le \epsilon$。

Remark 9.4. 在大多数关于该主题的论文中，提取器 F 被定义为不同的，但本质上等效，作为一个两个参数的函数：来自弱源的样本 D，以及一个（均匀分布的）索引 $i \in [t]$，因此 $F(i,x) = f_i(x)$。该索引被称为 *seed*，而 F 通常被称为 *seeded extractor*。

Enhancing weak random sources with extractors 让我们讨论如何使用提取器来模拟 $\mathcal{BPP}$ 算法，同时只访问具有足够熵的弱随机源。为某个决策问题固定一个概率算法 A 和它的一个输入 x。假设在一个真正的随机序列 $y \in \{0,1\}^r$ 上，它的输出 $A(x,y)$ 以（比如说）最多 $1/5$ 的概率出错。进一步假设我们有一个上述的 (n,k)-提取器 F，输出大小为 r 和错误为 ϵ。模拟器将利用某个 k-源 D 来获取一个 n-位样本 z，并从中计算出一组 r-位序列 $y_i = f_i(z)$。然后，它将每个 y_i 插入 A 中，计算所有输出 $A(x,y_i)$，然后返回它们的多数值。让我们分析这个新算法的质量。我们知道，对于 ϵ-分数的种子 i，当应用于 D 的样本时，函数 f_i 可能无法产生一个几乎均匀的（最多 ϵ）序列。这些种子可以产生一个错误值。但其他每个种子都生成一个几乎均匀的序列 y_i，因此当用于 A 时，出错概率最多为 $1/5 + \epsilon$。根据马尔可夫不等式，因此至少一半的种子导致错误的概率最多为 $2/5 + 4\epsilon < 1/2$。

注意，为了使此类仿真高效（即，在 r 中的多项式时间内），对提取器 F 产生一些限制。首先，我们必须有 $t \le r^{O(1)}$。接下来，每个 f_i 应该能在 poly(r) 时间内计算。因此，特别是，我们必须有 $n \le r^{O(1)}$。最后，作为 $k \le r$，这表明我们只能希望使用具有多项式熵 $k \ge n^{\Omega(1)}$ 的源。

Important parameters of extractors 如上所述，上述在概率算法中使用弱随机源的应用是提取器在理论计算机科学和实用计算机科学中的第一个应用，但绝不是最后一个应用。随机提取器的后续应用对不同的参数比上述参数更有趣，因此让我们考虑*all*在此提取器定义中可能的参数以及优化它们时的自然目标。我们将所有参数表示为n的函数，即从分布D中抽取的样本大小。首先是最小熵k；从任何k (源中提取是很不错的，尽管很明显，k越大，任务就越容易)。接下来是输出长度r；产生多少（几乎）纯随机位。如前所述，$r \le k$，并且让它尽可能接近k是自然的。接下来是“错误”参数ϵ，我们希望将其尽可能小。最后，使用的函数数量t；同样，这应该最小化，因为我们必须评估它们所有（为了效率，一个自然的目标是使t在n中是多项式的）。

[7] This is a well-known fact for Shannon's entropy and is actually easy to see for min-entropy as well.

早期结果 [Sip88, RTS00] 表明，至少 *existentially*，可以同时获得所有世界的最佳效果。也就是说，可以得到所有参数的最优值：输出熵 r 几乎等于输入熵 k，多项式数量的函数 t，以及逆多项式误差 ϵ。当然，这样的存在性结果并没有给出提取器的效率界限，而是阐明了我们在高效构造中应该追求的极限。

Theorem 9.5. *For every n and $k \leq n$, there 存在 an (n,k)-extractor F with output length $r \geq .99k$, $\epsilon \leq 1/n$, and $t = n^{O(1)}$. Indeed, a 随机 family of t functions F will be such an extractor with probability approaching 1 as n grows. Furthermore, these parameters are essentially the best possible.*

9.2 Explicit constructions of extractors

当然，主要问题是我们需要实际使用提取器，因此存在定理是不够的——我们需要对 {v*} 提取器 {v*} 进行显式构造，并具有良好参数。实现这一目标的道路，如上述综述中详细所述，是漫长而曲折的，涉及从较弱的源类（如“块源”和“某处随机源”）中进行提升，并使用各种较弱的提取器概念（如“冷凝器”和“合并器”）。所有这些结构都产生了新的和不同的归约概念。我现在只强调几个显式提取器的里程碑，它们逐渐降低了源熵，从线性到多项式，到任何东西。我还强调产生了不同构造的多样知识和技术来源，这进一步指出了伪随机世界的相互关联性。在以下讨论中，为了简单起见，假设 {v*} 1{v*}。我们只关注最小化源熵 {v*} 和函数（或种子）{v*} 的数量，并最大化源输出长度 {v*}（不能超过 {v*}）。

在本文的历史叙述结束时，我给出了一种提取器的显式构造，以及提取器在弱源未提及的另一种应用。

第一个显式提取器由Zuckerman [Zuc90, Zuc91] 提供。它只能处理熵 $k \geq \Omega(n)$ 和输出 $r \geq k/10$ 几乎均匀的比特。它涉及随机性高效采样和散列的复杂组合。经过几次改进后，Ta-Shma 引入并使用了 *mergers* 实现了巨大的飞跃。他展示了如何从一个具有任何最小熵 $k \geq \text{poly}(\log n)$ 的源中提取，输出长度 r 实际上与 k 一样大，只需略超多项式数量的种子 $t = \exp(\text{poly}(\log n))$。目标是减少种子的数量。

下一个里程碑是Trevisan [Tre99]的显式提取器，它实现了最优$t = \text{poly}(n)$，但仅适用于多项式熵 $k = n^{\Omega(1)}$，并且仅以输出长度$r > k^{.99}$。这足以完全解决上述弱源讨论的$\mathcal{BPP}$模拟问题。此外，这种构造在概念上与所有以前的结构都大不相同。事实上，它是一种简化。Trevisan的提取器将输入解释为 *computationally hard function* g的真值表，而函数f_i在司法选择的域元素上输出g的值。构造和分析遵循第7.2节中提到的NW-generator构造[NW94, IW97]。让我谈谈这个构造是多么有洞察力和令人惊讶。首先，它使用一个对象（伪随机生成器），按照定义只在计算设置中工作，并将其转换为另一个对象（提取器），按照定义是信息论上的。此外，这两种类型的概念似乎在相反的方向上工作：伪随机生成器从少量真正随机的比特开始，在许多比特上生成低熵分布，而提取器从具有一些熵的许多比特的分布开始，生成少量纯粹随机的比特。尽管如此，Trevisan表明，从正确的角度来看，NW-generator基本上就是提取器！

这个故事在许多想法和论文的推动下不断演变，最终迎来了一个美好的结局：在所有参数上本质上最优的提取器。

Theorem 9.6. *For every $k = k(n)$, there is a polynomial-time computable family $F = \{F_n\}$ of (n,k)-extractors, with output length $r \geq .99k$, $\epsilon \leq 1/n$, and $t = n^{O(1)}$.*

[8] Note that this is another instance of the “hay in a haystack” problem discussed in Chapter 8.

[9] Which we assume for notational simplicity is at least $10 \log n$.

Guruswami、Umans和Vadhan [GUV09]给出了这种显式构造的第一个例子。他们的提取器将输入解释为 *message in an error-correcting code*，而函数f_i输出该消息编码中的不同符号。该构造和分析特别依赖于[PV05、GR08b]中的最优列表可解码码。

不久之后，Dvir和Wigderson [DW11]给出了不同的构造。他们的提取器将输入解释为 *low-degree curve over a finite field*，而函数f_i输出曲线上的不同点。该构造和分析依赖于*polynomial method* 及其在Dvir [Dvi09]的证明中的应用，该证明涉及有限域几何中的有限域Kakeya猜想。

最后两个结果借鉴并连接到不同的数学领域，这是许多关于提取器工作的共同点，尤其是在本章开头提到的众多应用领域。

Extractors from expanders 我们以一个从不同起源的提取器显式构造作为结论：扩展图上的随机游走。实际上，这可以被称为原提取器，因为它在提取器被定义之前就已经存在，但后来才被认识到是一个提取器。此外，它还被用作许多后续提取器构造的一部分。这个提取器的参数相对较弱；对于某个固定的常数 α，源 k 的熵必须至少为 $(1-\alpha)n$，输出长度为 $r \geq \alpha k$，并且有常数误差 ϵ。尽管如此，即使在没有先见之明的情况下获得这个（输出）也是高度非平凡的。

我们需要定理8.9的显式展开式，例如，参数为c =2和d =16。设G_r为该定理中在2^r上的16-正则展开式.[10] 设$t = r/\epsilon^2$和$n = r +$为4($t -1$)。注意，每个n-位序列都可以解释为G_r中的长度-t路径，其中前r位指定第一个顶点，每个连续的4位段指定前一个顶点在16个可能邻居中的一个。注意$r \approx \epsilon^2 n/4$。

定义 t 函数 $F = (f_1, f_2, \ldots, f_t)$ 如下：f_i：$\{0,1\}^n \to \{0,1\}^r$，如下：$f_i(x)$ 简单地是 (r-位名称的) i 个顶点在由 x 指定的路径中。根据定理 8.9，F 可以在多项式时间内计算。

Theorem 9.7. *Set $\alpha = (\epsilon 2t)/(32n) = \Omega(\epsilon^2)$. Then F is an explicit (n,k)-extractor for $k = (1-\alpha)n$, $r \geq \alpha k$, and error ϵ.*

这个定理的证明来源于扩展开图随机路径的一个显著采样特性：它们的 t 个顶点，尽管是高度相关的 r 位字符串集合，但在用于计算 $\{0,1\}^r$ 上任何有界函数的样本平均值时，表现得像完全独立的；与真实平均值的偏差随着样本数量的增加而指数衰减。这首先由 Ajtai、Komlós 和 Szemerédi [AKS87]（为了去随机化小空间概率算法）以较弱的形式发现。在 [CW89, IZ89] 中为了增强概率算法中的误差，得到了加强，最后，Gillman [Gil98] 证明了本质上最优的界限（在 [Hea08] 中简化并扩展，我们从其中引用了我们记号中的一个特殊情况，即他的 $\lambda = 1/2$）。

Theorem 9.8 [吉尔98，希亚08]。*Let G_r be the expander above. Fix any function g: $\{0,1\}^r \to [-1,1]$ with zero expectation. Then for every $\epsilon > 0$ and every t, if $y_1, y_2, \ldots, y_t$ are the vertices of a uniformly random path in G_r, then*

$$\Pr\left[\left|\sum_i g(y_i)\right| > t\epsilon\right] < 2^{-\epsilon^2 t/8}.$$

观察当 y_i 独立时，这是经典的大偏差（Bernstein/Chernoff）界。巨大的差异在于独立样本需要 rt 个随机比特，而该定理表明，只需 $r + O(t)$ 个比特即可实现相同的估计误差，这是最佳可能的。上述概率算法错误放大的应用表明，任何使用 r 个随机比特且具有错误（例如）1/3 的算法可以通过仅使用 $O(r)$ 个比特转换为具有错误 $\exp(-r)$ 的算法，这与明显的 r^2 相比。这可以被视为比第 8.7 节中讨论的错误减少[11] 更为令人印象深刻的结果。

[10] Note that we change n of that theorem to r here, as we reserve n for the input length of the extractor.
[11] There the error was reduced to $1/\text{poly}(r)$, but without adding any extra random bits at all.

我们通过连接最后两个定理来得出结论。它们本质上等价（一旦匹配参数）。这种提取器和（遗忘的）采样器之间的等价性由Zuckerman在[Zuc97]中提出，简单的证明几乎可以说是定义性的。Vadhan的综述[Vad11]包含了关于提取器与采样器、散列函数、纠错码和其他伪随机对象之间联系的详细讨论。

9.3 Structured weak sources, and deterministic extractors

本章重点介绍了*seeded extractors*，它需要一些额外的真正随机的比特来净化弱随机源。正如所解释的，这是他们所追求的崇高目标所必需的，即净化给定最小熵的*arbitrary*弱随机源，不对这些源提出结构要求。事实上，正是这种能力赋予了种子提取器其惊人的多样性应用。

但是，在许多已知或可以假设的弱源结构的情况下；在某种意义上，限制了熵在由源定义的分布中抽取的序列上的分散形式。对于这样的（家族）源，能否构建完全确定性的提取器（无需额外随机种子）来净化它们？答案是肯定的，在许多不同的情况下，我们简要列举了这些情况，并为感兴趣的读者提供了一些关键参考文献。不用说，确定性提取器也有各种应用，它们的构建与数学和计算机科学的其他领域（尤其是有种子提取器的构建）美妙地相互作用。

研究过的结构化随机源类型以及为它们构建的确定性提取器大致分为三类。第一类是 *algebraic*，其中分布由自然代数函数生成。这些包括“仿射源” [GR08a, Yeh11, Li16]，“多项式源” [DGW09] 和“变体” [Rem16]。第二类是 *computational*，其中分布由具有有界资源（时间、空间或其他）计算的函数生成 [TV00, KRVZ11]。第三类是 *independent sources*，其中分布从无结构源 *several* 中采样 [CG88, BIW06, CZ16]（最后一篇论文是第8.2节中提到的关于显式拉姆齐图的突破）。

这是一个尝试对允许确定性提取器的结构化来源进行分类的非常有趣的研究方向。Chattopadhyay和Li [CL16]给出了一个重要的定义，称为*sumset sources*，它捕捉了许多上述结构。

10 Randomness and interaction in proofs

随机性和交互性的引入，以及 *proofs* (和证明系统)，完全焕发了数学这一核心原则的活力。它对理论计算机科学以及许多相关领域产生了显著影响，带来了许多意想不到的后果。特别是，它导致了$\mathcal{NP}$和其他复杂类的新、强大的特征描述。

在这一章中，我们概述了概率性和交互式证明的主要定义和结果，以及它们的含义和影响。《Gol99》专著提供了更多细节和精确性；对过去20年中该领域如何演变进行全面综述将受到欢迎！我们注意到，在本节中，我们讨论了*not*，一种强大的证明*probabilistic method*。关于它的优秀文本是[AS00]。最后，鼓励读者回忆第1.5章中$\mathcal{NP}$作为证明系统的含义，以及第15.2.4章中的命题证明系统概念，这些概念将在这里通过随机性和交互性进行扩展。

让我们从一个例子开始。考虑第3.3章中提到的图同构问题：给定两个图{v*}和{v*}，确定它们是否同构。对于这个问题，还没有已知的多项式时间算法.{v*}现在假设有一个无限强大的老师（特别是可以解决这样的问题），想要说服一个有限的、多项式时间的学生，两个图{v*}是同构的。这很简单——老师只需提供两个图的顶点之间的双射，学生就可以验证边是否被保留。这只是对事实的重新表述，即{v*}，所有同构对集合({v*})，包含在{v*}中。但是，老师是否有类似的方法来说服学生，两个给定的图是{v*}同构的？是否已知{v*} co{v*}，所以我们没有这样的非同构的简短证书。能做什么呢?

这里是从[GMW91]中得到的想法，它允许学生和教师进行更详细的互动，以及抛硬币。学生如下挑战教师。她（秘密地）抛硬币来选择两个输入图G或H中的一个。然后她通过随机排列顶点名称（再次使用秘密硬币抛掷）创建所选图的*random*同构副本K。然后她向教师展示K，教师被挑战确定K是从G还是H生成的。观察如果G和H确实是非同构的，如所声称的，那么答案将是唯一的，教师（拥有无限的计算能力）总能应对这个挑战。然而，如果G和H是同构的，*no*教师可以以大于1/2的概率猜测K的来源。简单来说，这两个随机变量——G的随机同构副本和H的随机同构副本——是*identically distributed*，因此无论计算能力如何，都无法区分。现在，为了减少错误，让学生独立重复这个实验100次，除非教师成功完成所有实验，否则宣布这两个图不是同构的。因为当G和H是同构的时候，100次成功的概率是2^{-100}，这限制了学生错误地接受错误证明的概率。换句话说，这种重复的成功描述了一个压倒性的令人信服的*interactive proof*，表明这两个图确实是非同构的。

注意，隐藏学生的抛硬币结果给老师是这个证明系统的绝对必要功能。事实上，很难想象如果老师能够从学生的肩膀上窥视并确切知道所有抛硬币的结果，能否实现类似的壮举。这种直觉是错误的！Goldwasser和Sipser [GS89] 的一个显著结果是给出了另一个（更加复杂的）非同构的交互式证明系统，其中学生的所有抛硬币结果都可供老师查看。事实上，他们给出了一种完全通用的方法，将任何“私有硬币”交互式证明系统（其中老师无法看到学生的抛硬币结果）转换为“公共硬币”系统（其中老师可以看到它们），其效率相似。鼓励读者尝试找到一个图非同构的交互式证明，其中验证者向证明者发送的唯一消息是随机比特.[2]

让我们现在回到一般讨论。我们已经在第3.3节和第6章中讨论了证明系统。在这两个地方，一个给定的证人对于给定的断言确实是证明的这一*verifier*要求必须是一个高效的*deterministic*过程。在前面章节关于随机性的精神下，我们现在放宽这一要求，允许验证者掷硬币并以极小的概率出错。

[1] Although the recent breakthrough of Babai [Bab15] (see also the exposition in [HBD17]) comes quite close!

[2] Insufficient hint: The public-coin proof, like the private-coin proof above, should rely on the fact that the number of isomorphic copies of G and H together is twice as large when they are nonisomorphic than when they are isomorphic.

为了使本定义中的量词清晰，以及允许证明者和验证者之间有更一般的交互，将一个集合S（（例如，可满足公式集合））的证明系统视为一个$game$，在全能证明者和（有效、概率性）验证者之间：两者接收一个输入x，证明者试图说服验证者$x \in S$。完备性规定证明者对每个$x \in S$都成功。正确性规定对于每个$every$证明者都失败。在$\mathcal{NP}$的定义中，这两个条件都应该成立$with\ probability\ 1$（在这种情况下，我们可以将验证者视为确定性的）。在概率性证明系统中，我们放宽了这个条件，只要求正确性和完备性以高概率成立（例如，2/3，因为错误可以通过迭代和多数投票任意减少）。换句话说，对于每个输入，验证者很少会掷硬币，这会导致它错误地判断断言的真实性。

此扩展标准 $\mathcal{NP}$ 证明的建议在两篇论文中独立提出——一篇是Goldwasser、Micali和Rackoff [GMR89]（其动机来自密码学，其中此类交互很普遍）的，另一篇是Babai [Bab85]（其动机是为群论中的自然问题提供此类交互“证书”，这些问题尚未被证明属于co$\mathcal{NP}$）。虽然原始定义不同（验证者的掷币是否为证明者所知），但上述Goldwasser和Sipser [GS89]的论文表明这两个模型是等价的。

这个证明的放宽并不建议作为数学真理概念的替代。相反，与概率算法类似，建议在微小[3]错误无关的情况下，大大增加可以高效证明的主张集合。正如我们下面将看到的，*probabilistic proof systems yield enormous advances in computer science, while challenging our basic intuitions about the very nature of proof.* 我们展示了该陈述的三个不同显著表现：

- 许多定理可以高效地证明。
- 每个定理都可以证明，而不透露关于证明的 *anything*，除了其有效性。
- 每个定理都有书面证明，验证者只需检查少量即可验证。 位。

10.1 Interactive proof systems

当验证器是确定性的时，交互不会增加能力，因为证明者可以预测所有未来的问题。因此，在这种情况下，我们可以始终假设证明者简单地发送一条消息（所谓的“证明”），并且基于这条消息，验证者决定是否接受或拒绝将共同输入x作为目标集S的成员。换句话说，在确定性验证器的情况下，交互式证明只能证明$\mathcal{NP}$中的陈述。

当验证器是概率性的时，$interaction$可能会增加功率。因此，我们允许双方抛硬币，并考虑他们之间的（随机化）交互。这可以被视为一个坚持不懈的学生对老师连续提出一系列“难题”，以确信正确性（或捕捉到错误）。由于验证器应该高效（即，在$|x|$的多项式时间内运行），这种问题轮次的数量被多项式限制。[4]

Definition 10.1 (该类 $\mathcal{IP}$ [GMR89,Bab85])**.** 简称“交互式证明”，包含所有集合 S，对于这些集合存在一个概率多项式时间验证器，它以概率 1 接受每个 $x \in S$（在与 *some* 充足的证明者交互后），但以至少 1/2 的概率拒绝任何 $x \notin S$（无论证明者采用何种策略，或策略的计算强度如何）。

我们已经看到了这种证明在上述图非同构示例中的潜在力量（在本章引言中），并给出了几个其他示例。但完整的威力

[3] And we remind the reader that the error can be made exponentially tiny without affecting efficiency by much!

[4] Restricting the number of rounds to a constant, as was suggested in the original paper of Babai [Bab85], leads to the extremely interesting “Arthur-Merlin” complexity classes $\mathcal{AM}$ and $\mathcal{MA}$, which sit just above $\mathcal{NP}$. We will not define and study them here but note that they were extensively studied, and that the interactive proof above for graph isomorphism above puts this problem in the class $\mathcal{AM}$.

$\mathcal{IP}$ 仅在被称为“$\mathcal{MIP}$”的更强大的证明系统被Ben-Or等人提出[BOGKW89]（受密码学考虑的启发！）之后才开始展开。在$\mathcal{MIP}$（（“多证明者交互证明”的简称）中，验证者与*multiple provers*互动，*multiple provers*不允许相互通信。我们描述了一些导致这种理解的工作和思想的演变。Babai [Bab90]（参见[O’D05]）给出了这一快速进展的生动描述。

一个里程碑，由Lund、Fortnow、Karloff和Nisan [LFKN90]提出，表明$\mathcal{IP}$证明可以用于*every*集合在co$\mathcal{NP}$（中，实际上，对于更多的情况，但对我们在这本书中没有定义的类)。因此，特别是，*tautologies*有简短的交互式证明。回想一下，我们并不期望这些有标准的$\mathcal{NP}$-证明，因为这会意味着$\mathcal{NP} = \text{co}\mathcal{NP}$（参见第3.5节的讨论，特别是，猜想3.8)。

Theorem 10.2 [LFKN90]. $\text{co}\mathcal{NP} \subseteq \mathcal{IP}$.

如所述，这个定理只是[LFKN90]中主定理的一个特例，我在本节末尾陈述并概述了它。

这篇论文随后是Shamir [Sha92] 对 $\mathcal{IP}$ 的完整表征。他证明了它与 $\mathcal{PSPACE}$ 等价，即可以用多项式存储（和可能指数时间）计算出的函数类。请注意，这个类包含似乎比 $\mathcal{NP}$ 和 co$\mathcal{NP}$（更难的问题，例如寻找游戏的最佳策略。

Theorem 10.3 [Sha92]. $\mathcal{IP} = \mathcal{PSPACE}$.

这是一个启发性的定理的非正式后果，我觉得令人震惊。假设某个高级外星生物（让我们简称它为“E.T.”）来到地球，声称他们的文明已经研究过象棋，并发现 *“White has a winning strategy”*.[5]。有办法验证或反驳这个说法吗？虽然我们有一些非常优秀的棋手和程序，但我们自己的文明至今没有手段来确定这样的说法。简单来说，我们唯一已知的（算法）方法是穷举法；扩展象棋的完整游戏树。但这个树在移动数量上是指数级的，一个巨大的数字，使得这种计算超出了任何可预见的未来技术。当然，我们可以提供我们最好的选手（无论人类还是其他）与E.T.对弈。但假设他们在所有比赛中都输了；我们只能得出结论，E.T.是一个更好的选手，这远远不足以验证它关于白方有获胜策略的说法。然而，定理10.3确实提供了一种验证它的有效方法!

为了解释这种微妙的关系，首先考虑一个荒谬愚蠢的验证尝试。让我们将E.T.与我们的最佳选手而不是最愚蠢的选手匹配；一个完全随机选择每个移动的人，从给定配置的所有可用合法移动中选择。称这种概率策略（对于黑方）为*Random Play*（或R.P.（随机策略）)。不用说，即使是作为白方的初学者也会在棋类游戏中击败R.P.，E.T.当然也会。下一个观察是，尽管R.P.看起来很愚蠢，*is*但对于某些其他游戏（例如石头剪刀布）来说是最优的[6]。但当然，石头剪刀布与棋类游戏无关。

令人惊讶的是，Shamir定理使我们能够在与象棋密切相关游戏中使随机游玩策略变得有用！该定理提供了一种新的归约方法：一种将象棋转换为新的两人游戏G的方法，具有以下特性。

1. 首先，转换是高效的：游戏的规则 G 对我们这样的凡人来说很容易理解。特别是，在 G 的每种配置中哪些移动是合法的很容易计算，因此在这个游戏中实现 R.P. 很容易。

2. 第二，这两款游戏在以下意义上是等价的。在新游戏 G *iff* 中，白方有获胜策略。白方在棋类游戏中也有获胜策略。因此，特别是，E.T. 可以将其声称的获胜棋类策略（如果存在）转换为 G 的获胜策略。

[5]Recall that a *strategy* for a player in chess, or any perfect-information game, is simply a prescription of a legal move for that player in every possible configuration of the game. It is a *winning strategy* if it guarantees a win for that player, regardless what strategy the opponent chooses.

[6]Ignore the fact that this game is of a somewhat different type of game, in which moves are simultaneous. There are other examples. Indeed, the result of Goldwasser and Sipser [GS89] mentioned in the previous Section 10 reveals surprising power of random play in a related context.

3. 最后，G具有以下性质：黑方的随机游戏R.P.是*nearly optimal*——它以1/2的概率做得和最佳黑方策略一样好。

因此，让我们明确这种减少意味着什么。如果白方在棋局中有一个获胜策略，并且E.T.利用它以最优方式在G中玩游戏，因为它可以通过属性（2）做到，那么R.P.在G中仍然会每次都输。但如果白方在棋局中没有获胜策略，那么根据属性（3），E.T.在对黑方最优策略的情况下，无法以高于1/2的概率赢得G。因此，如果E.T.连续赢得100场对R.P.的比赛，我们就有概率$1 - 2^{-100}$认为白方确实在棋局中有一个获胜策略。

三个评论可能回答您关于这个结果的一些疑问。首先，同样的推理适用于围棋而不是象棋，实际上适用于我们玩的任何合理游戏。其次，如果我们适当地推广象棋或围棋（并将其作为类$\mathcal{PSPACE}$的完整问题，这类2人完美信息游戏的自然家园），那么这种博弈论解释实际上等价于定理10.3。第三，如果$\mathcal{P} = \mathcal{PSPACE}$ (这会暗示$\mathcal{P} = \mathcal{NP}$，尽管没有人相信但仍然是一种数学上的可能性)，那么就有一个快速算法可以确定象棋、围棋和类似游戏中的最优策略。

这个故事，完全理解交互式证明的惊人力量，需要来自计算复杂性不同“角落”思想的汇聚和融合，再次展现了其方法论的力量以及不同子领域间思想交叉融合的流程。导致原始证明的结果序列使用了以下成分（对这些成分的详细阐述超出了我们的范围）：{v*}

- *Program checking and testing* [BK89, BLR93]。
- *Hardness amplification* 从平均情况到最坏情况难度 [Lip91, BF90]。
- 计数的力量和永久多项式来捕捉 co$\mathcal{NP}$ 以及更一般的有限交替 [Tod91]。
- 多证明者模拟[BOGKW89]基本交互证明模型，受零知识证明（见下文）的启发。
- 复杂类 $\mathcal{PSPACE}$ 的结构。特别是，非确定性不会增加此类的能力 [Sav70]，以及为其特定的完备问题（即 QBF：量化布尔公式的可满足性）的结构。

一个在10.3节（以及新的下界证明中）出现并发挥重要作用的中心技术工具是 *arithmetization*，即通过多项式对布尔公式进行算术编码以及其属性的极快验证。算术化及其后果对于某些电路下界证明起到了关键作用；其影响和局限性在5.1节的末尾被提及。

我总结说，更强的$\mathcal{MIP}$证明系统的确切能力也被Babai、Fortnow和Lund [BFL91]完全确定。在这里，这也相当于一个自然的复杂性类，在这种情况下是$\mathcal{NEXP}$ (所有在非确定性指数时间内计算的语言的指数时间类似物。

Theorem 10.4 [BFL91]. $\mathcal{MIP} = \mathcal{NEXP}$.

10.2 Zero-knowledge proof systems

本节可能包含你将遇到的最为反直觉的数学结果！

假设你是一位刚刚发现黎曼猜想证明的初级数学家。你想说服数学界你的成就，但你极度偏执，担心如果将证明透露给任何人（可能是一位资深专家），他或她会声称那是他们自己的。虽然可能性不大，但这可能会对你的职业生涯造成毁灭性的打击。有没有办法防止这种情况发生？你能否在不透露任何线索的情况下让每个人都相信你知道一个证明？等等。

本节的重点不是证明更多定理，而是寻找具有额外属性的证明。随机和交互式验证程序（例如第10.1节中的那些）允许有意义的引入*zero-knowledge proofs*，这些是除了它们自身的有效性之外还能产生*nothing*的证明。

这样的证明似乎是不可能的——在不给他们任何信息的情况下，你如何说服任何人相信他们已经知道的事情？在数学中，当我们自己无法证明一个定理时，我们觉得看到证明会*necessarily*教会我们一些我们不知道的东西，而不仅仅是它是真的。

嗯，上面给出的交互式证明，即两个图是非同构的，至少表明在某些特殊情况下，零知识证明是可能的！请注意，在证明的每一轮中，学生都知道他挑战的答案是什么，所以他并没有从老师的答案中学到任何东西。换句话说，*if*图确实是非同构的（即要证明的陈述是真实的），学生可以与老师生成对话，*without*实际上与他互动！毕竟，在这种情况下，学生知道他生成的每个问题的唯一正确答案。而且，从你可以与自己进行的对话中，你无法获得新的知识。尽管如此，他们之间实际进行的对话，老师在许多挑战中重复成功识别正确的图，实际上说服了学生，确实，这些图是非同构的。简而言之，这种交互式证明（至少直观上）是一种零知识证明.[7] 它是有说服力的，并且没有向学生透露任何他不知道的东西，除了陈述本身的真相。

如何定义一般交互式证明中的这个*zero-knowledge proofs*概念？这种动机、形式定义以及这个非凡概念的某些例子都在定义交互式证明的同一篇开创性论文 [GMR89] 中给出。定义相当微妙且技术性，我们只以高层次术语概述其精髓（更多细节见第18章关于密码学）。从上面的例子中扩展直觉，我们可以要求验证者在每个正确的断言上能够有效地生成，*by herself*，（与证明者的）对话的概率分布。这实际上是不必要的严格。事实上，如果我们对验证者自己能够从实际对话（在第7.3节中形式定义）中生成的*computationally indistinguishable*满意，那就足够了。从这个意义上说，零知识证明意味着证明者不会泄露任何知识，这些知识可以被任何有效算法（如验证者）利用。

现在，哪些定理有零知识证明？嗯，如果验证者无需帮助就能确定答案，那么这是显而易见的。因此，$\mathcal{BPP}$中的任何集合都有一个零知识证明，其中证明者什么也不说（验证者自己决定）。一些被认为在$\mathcal{BPP}$之外，如图非同构，已知有这样的证明无条件。

也许令人惊讶的是，使用密码学的标准假设，即*one- way functions*存在（见第4.5节），那么可以为*every*的感兴趣定理给出零知识证明！Goldreich，Micali和Wigderson [GMW91] 证明了以下内容。

Theorem 10.5 [GMW91]. *Assume the existence of one-way functions. Then every set in $\mathcal{NP}$ has a zero-knowledge interactive proof.*

加密假设是必不可少的；该定理的逆命题在[OW93]中（被）公式化并证明了。

零知识定理的证明再次证明了归约和完备性的力量。它在[GMW91]中分为两部分证明。第一部分给出了形式为*"a given graph is* 3*-colorable"*.[8]的陈述的零知识证明（仅限于此）。第二部分通过使用图3着色问题的$\mathcal{NP}$-完备性，从第一部分推断出所有$\mathcal{NP}$集合都有一个零知识证明。我们简要讨论了这两部分。

[7] There is a subtlety, explained in [GMW91], which necessitates altering the original proof so as to formally make it zero-knowledge. This is an important subtlety, which arises from the need to prove zero-knowledge for "'cheating verifiers"; a verifier might deviate from the protocol (e.g. choose questions using a different probability distribution than is prescribed) in order to extract knowledge from the prover's answers.

[8] It is important to note that this first part uses *specific* combinatorial properties of the 3-coloring problem, in much the same sense as the combinatorial properties of graph nonisomorphism were used in its zero-knowledge proof.

[9] Specifically, it uses the strong form of reductions, mentioned after Theorem 3.13, which allows efficient translation of witnesses, not just instances, of problems in $\mathcal{NP}$.

对于第一部分，我们需要一个协议，其中证明者说服验证者一个给定的图是3可着色的，而不透露关于着色（或任何其他信息）。相同的协议应保证如果图不是3可着色的，验证者将以非可忽略的概率“抓住”任何作弊的证明者。以下对协议的简单描述捕捉了其本质，但失去了许多对于正式保证这两个重要（且直观上冲突）属性所必需的重要细微差别。主要思想是这样的。密码学假设允许证明者将$commit$到一个完美$hides$顶点着色，类似于将每个顶点的颜色放入一个单独的密封信封中。然后验证者随机选择图中的一条边，并要求证明者“打开”（如密码学所允许）其端点的这对信封，揭示这两个相邻顶点的颜色。如果图是3可着色的，证明者可以生成一个“足够随机的”合法3着色，这确保了任何被揭示的颜色对都是完全随机且不同的；这当然不会向验证者泄露任何信息。如果图不是3可着色的，那么至少有一对这样的信封将包含证明者作弊的明显证据（两个相同的颜色，或可能是非法内容），并导致验证者拒绝.[10]

对于第二部分，我们需要给出一个从零知识中证明任意 $\mathcal{NP}$ 声明的归纳，使用为 3-着色图执行此操作的协议。让我们看看这种归纳是如何实施的，以证明“应用”上述零知识定理到（安全地）说服同事你已取得数学突破的合理性。假设你确实已经证明了黎曼猜想，并且担心在听众之前公布证明。通过 3-着色图的零知识证明，你可以使任何人确信无疑地相信你确实有这样一个黎曼猜想的证明，而不会泄露任何关于它的信息。你按以下步骤进行。首先，使用隐含在证明 $3COL$ 是 $\mathcal{NP}$-完备的证明中的有效算法，将黎曼猜想的陈述转换为图，并将你的证明转换为该图的适当合法 3-着色。现在使用 [GMW91] 的 $3COL$ 协议来说服你的听众这一事实。请注意，听众可以自己执行归纳的第一部分（从黎曼猜想到图），因此他知道你正在证明一个等价陈述。

但是定理10.5的巨大影响并不在于其（重要）应用于版权和抄袭保护。零知识证明是强迫参与加密协议的参与者正确行为的主要工具，也就是说，不侵犯任何人的隐私[GMW91]。本质上（忽略了许多复杂性），零知识证明允许加密协议中的各方通过在零知识中证明（这些秘密是永远不会被揭示的证明）来使他人相信他们的消息（部分取决于他们的私人秘密）是根据协议计算的，而不泄露这些秘密。让我们详细阐述这一和后续工作在简化通用加密问题协议设计任务方面的概念贡献。请注意，甚至定义以下许多直观概念也是高度非平凡的；读者再次被敦促查看第18章的密码学以获取更多细节。

论文[GMW91]给出了一种（新类型的）化简：它将一个只有在所有参与者都遵循它时才能保证隐私的协议[11]，自动生成一个即使在某些参与者出现故障或恶意偏离协议的情况下也能保证隐私的协议[12]。不久之后，姚[姚86]设计了著名的$secure\ evaluation$协议，该协议适用于诚实参与者（从2个参与者扩展到[GMW87]中的任意数量参与者）。这些协议[13]提供了一种完全不同的化简，将任意电路（其输入分布在不同的参与者之间）转换为（诚实参与者）的协议，该协议在这些输入上评估此电路，而不向任何玩家子集泄露任何信息，这些信息超出了输出本身所揭示的信息。作为一个简单的例子，从姚的论文中可能可以展示这一成就，尝试为两个参与者设计这样的协议，每个参与者持有n-位整数，代表两位试图找出谁更富有而没有透露自己价值的大富翁。另一个例子是举行选举：n人每人持有一个比特（比如说），并试图找出多数投票。再次，每个人都诚实，会严格遵循协议的每一条指令！他们只是好奇，因此协议应该被设计成

[10] Amplifying this non-negligible probability of catching the verifier is another important issue we do not discuss.
[11] Designing such a protocol for *honest* parties is a highly nontrivial task by itself, which we presently discuss.
[12] As in all such reductions, what is actually shown is that the ability of some parties to gain access to secrets of others entails an efficient algorithm to invert a one-way function.
[13] Which rely on the existence of trap-door functions.

没有它们的任何子集能学习到*anything*，即输出本身不揭示关于其他人的输入信息。

The combination of a secure evaluation protocol for any function that is private assuming honest players, and the compiler of such a protocol making it resilient against malicious parties, yields a *private and fault- tolerant* implem entation of just about any cryptographic task. For a good example of the complexity of such tasks, which now becom e implementable, consider how a group of untrusting parties can play a game of poker *over the telephone*. No physi cal implements (like cards with opaque backs for poker) are allowed (or needed)—only digital communication and tr ap-door functions. As mentioned, this example and the general problem of secure function evaluation are discussed a t length in the cryptography Chapter 18. 组合任何假设诚实玩家私有的函数的安全评估协议，以及使该协议对恶意方具有弹性的编译器，可以实现对几乎所有加密任务的 *private and fault- tolerant* 实现。为了说明此类任务的复杂性，这些任务现在可以实施，考虑一群不信任的各方如何玩扑克牌 *over the telephone*，不允许（或需要）任何物理工具（如扑克牌背面不透明的扑克牌）——只需数字通信和陷阱门函数。如前所述，这个例子和安全的函数评估的一般问题在密码学第18章中进行了详细讨论。

我们在这里只提到一个也许最令人惊讶的、回想一下本节的证明结果，定理10.5，它建立了每个定理的零知识 *interactive* 证明。事实证明，可以定义一个模型，其中零知识证明可以是 *non-interactive*，即正常的书面证明！这个模型以及与之相伴的结果，即其中每个定理都有这样的非交互式零知识证明，出现在[BDSMP91]中。

10.3 Probabilistically checkable proofs (PCPs), and hardness of approximation

在这个部分，我们转向关于概率可验证证明（PCPs）的力量及其对近似算法极限的影响的最深刻和最令人惊讶的发现之一。

我们回到非交互式模型，其中验证者收到一份（声称的）书面证明，如$\mathcal{NP}$所示。然而，我们现在限制验证者对证明的访问，以便只能读取其中的一小部分（可能是随机选择的）。实际上，将证明出，只有一份书面证明中适当格式化的常数位数量就足以以高概率验证其正确性！这个极具反直觉的PCP定理，是复杂性理论的主要亮点之一，是本节的重点。

值得注意的是，尽管*non-interactive*证明在书写上自然，但对于我们来说，本节讨论的模型非常间接地产生。它是从第10.2节中提到的[BOGKW89]的多证明者*interactive*证明模型$\mathcal{MIP}$推导出来的。回想一下，在$\mathcal{MIP}$中，验证者与*several*、相互不通信的证明者[14]进行通信。

多证明者系统与书面证明之间的联系是[FRS88]提出的简单而强大的观察。他们证明了$\mathcal{MIP}$（（*unlike*其单证明者兄弟$\mathcal{IP}$）），是*equivalent*于一个非交互式、书面证明系统（如$\mathcal{NP}$），但该系统（与$\mathcal{NP}$不同）将验证者限制为对证明进行少数随机探测.[15]

一个出色的、熟悉的“懒验证”设置是当裁判员试图通过抽样检查证明中的几行（例如，10行）来决定一个长（例如，100页）证明的正确性。这似乎是无用的；除非阅读整个证明，否则一个人怎么能希望检测到单个错误呢？然而，这种直觉只有在“自然”的证明书写方式中才是有效的，在这种方式中，单个孤立的错误确实可能存在。令人惊讶的是，当使用*robust*格式的证明时（以及，像往常一样，当我们容忍微小的错误概率时），这种直觉就失败了。

这样的鲁棒证明系统被称为PCPs（代表“概率可验证证明”）。粗略地说，集合$S \subseteq \mathbf{I}$的PCP系统由一个具有访问权限的概率多项式时间验证器组成

[14] Curiously, the motivation of [BOGKW89] for defining $\mathcal{MIP}$ was also indirect, arising from cryptography and zero-knowledge proofs (discussed in the previous subsection). As mentioned there, for the single-prover proof system $\mathcal{IP}$, achieving non-trivial zero-knowledge necessitates having one-way functions. $\mathcal{MIP}$ was born in [BOGKW89] to provide a natural alternative proof system allowing *unconditional* zero-knowledge proofs. But now that it was born, complexity theorists had a new toy to play with (well, a new complexity class $\mathcal{MIP}$ to understand), and doing so created the snowball described below that eventually led to PCPs and their applications.

[15] The proof of this observation goes roughly as follows. In the easy direction, a multi-prover proof can be converted into a written one by writing down all provers' answers to all possible queries by the verifier. In the subtle direction (which does not work with a single prover), a written proof becomes the strategy of the (say) two provers, using which they respond to the verifier's queries. Of course, the verifier should not trust them, and so must make sure that they give consistent answers on the queries he would have made to the written proof. The fact that the provers cannot communicate underlies the soundness of this argument. Note that in both directions, the number of bit queries to the written proof is roughly the same as the number of communicated bits between the verifier and the provers.

对表示（所谓）证明的字符串中的各个比特位进行操作.[16] 在输入 x 和一个（所谓证明）y 时，验证者抛掷硬币，并相应地只访问所谓证明中的 $constant(!)$ 个比特位y。就像任何证明系统一样，它满足通常的完备性和正确性条件，尽管几乎不查看所谓证明！对于 $every\ x \in S$，存在一个序列 y (正确的证明)，验证者以概率 1 接受。然而，对于 $every\ x \notin S$，验证者以至少 1/2 的概率拒绝（无论给定的“所谓证明”y 是什么）。

一系列长篇思想和论文，由Arora [Aro94] 和 Sudan [Sud96] 进行调查，其中对所写所谓证明的随机探测次数最终减少到固定常数，最终导致了PCP定理，这是Arora、Lund、Motwani、Sudan和Szegedy [ALM+98] 对 $\mathcal{NP}$ 的一种强大新特征描述。**Theorem 10.6** (PCP定理 [ALM+98])。
Every set in $\mathcal{NP}$ has a PCP system. Furthermore, there exists a polynomial-time procedure for converting any $\mathcal{NP}$-witness to the corresponding robust PCP-proof.

确实，PCP定理的证明提出了一种新的编写鲁棒证明的方法，其中任何错误都必须“传播”到各个地方。等价地，如果找到这些由验证器扫描的少量位中存在的错误的概率很小（比如说，$\leq$ 1/10），则该定理是正确的。令人瞩目的PCP定理是通过一个相当复杂和技术的（主要是代数的）证明被证明的，这个证明在十多年里抵抗了重大的简化。然而，Dinur [Din07]后来给出了一种概念上不同的（组合）证明，这个证明非常优雅且简单得多，他使用了扩张图（见第8.7节）以本质的方式。

10.3.1 Hardness of approximation

自20世纪70年代以来，我们有一个详尽的理论来解释优化问题的困难：对于大多数自然问题，我们知道它是在$\mathcal{P}$或$\mathcal{NP}$-难。但是，无法通过有效算法高效地找到精确的最优解自然导致人们寻求最优解的有效近似。存在许多保证不同问题具有各种近似因子的有效近似算法，但在PCP定理之前，我们没有并行理论来解释这些算法有多好，或者有效近似性的极限是什么。这是一个重大的概念和实践问题，在二十年内几乎没有进展。令人惊讶的是，正是交互式证明的进展是这一重大挑战取得进展的源泉。

PCP定理彻底改变了我们论证某些优化问题不仅难以精确求解，甚至难以得到粗略近似的能力。事实上，它引领了一个美丽的理论，对于许多自然问题，可以确定它们如何被有效地近似。

The connection between the probabilistically checkable proofs and *hardness of approximation* was discovered by Feige et al. [FGL+96] and is elaborated on in the surveys mentioned above. We note that while we present this connection *assuming* the PCP theorem, it was actually discovered earlier, from (scaled down) versions of Theorem 10.4. This connection in turn presented a major motivation to the discovery of the PCP theorem!

让我们探索如何从PCP定理中得到一个$\mathcal{NP}$-难近似问题。考虑PCP验证器在给定实例（例如，SAT）上的行为；它可以由一组对给定PCP证明的局部测试来描述；每个测试指定要读取的位子集，以及这些位置中的值集合，这些值会导致验证器接受。现在，如果我们把所谓的PCP证明中的位视为布尔变量，验证器接受的问题就变成了一个约束满足问题（CSP；参见第4.3节）。PCP定理保证，要么所有约束都是可满足的（如果它是一个“是”实例），要么最多有1/10个是（如果它是一个“否”实例）。这构成了从SAT到这个CSP的归约。将这个CSP中满足约束的最大比例近似到因子<10以内，将导致一个求解SAT (exactly)的算法，因此近似这个CSP是$\mathcal{NP}$-难的。

这个近似问题似乎很牵强，也许在实践中不会出现。但再次强调，一旦我们有了它，我们就可以尝试使用它，并证明其他更自然近似问题的困难性。实际上，近似问题之间的归约通常比标准问题更难证明。

[16] In the case of $\mathcal{NP}$-proofs, the length of the proof is polynomial in the length of the input.

$\mathcal{NP}$- 完整性结果，并且它们通常需要大量的分析工具。我们提到了两个最强此类 *inapproximability* 结果的例子，都归功于 Håstad [Hås99, Hås01]。这两个结果几乎都是紧的，即通过给定的因子逼近解是 $\mathcal{NP}$-难的问题，但用稍大的因子来做是平凡的。在这两个例子中，$\varepsilon > 0$ 可以是一个任意小的常数。

- **Linear equations.** 给定在 $\mathbb{F}_2$ 上的一组线性方程组，近似求解满足条件的最大数量，误差在 2 倍以内 $-\ \varepsilon$ (显然，2 倍的误差是平凡的：随机分配即可)。

- **Clique.** 给定一个有 n 个顶点的图，将其最大团的大小近似到 $n^{1-\varepsilon}$ (的因子内，显然，一个因子 n 是平凡的：一个顶点就可以。

尽管这些令人惊叹的结果，精确确定许多问题有效逼近的确切极限，对于许多其他问题（例如 *max-cut, vertex cover*），已知的最佳逼近比不匹配PCP定理提供的（目前）最佳$\mathcal{NP}$难度结果。这些差距导致了独特游戏问题和UGC的发展，这在第4.3节中讨论，它关闭了众多逼近问题中的这些差距。

10.4 Perspective and impact

无法高估本章中心思想的影响，即通过随机性和交互性丰富 *proof* 的基本概念。虽然形式证明的概念在确立数学真理（包括本章证明的真理）方面保持不变，但交互式和概率证明（确实留下了一些错误的机会）在许多方面都有回报：它们具有形式证明无法拥有的特性，在实际情况中极其有用，并引发了一系列新的理论发展。在这里，我们简要回顾了这一现象的一些主要例子，再次强调这些源于密码学和群论动机的原始提议在其他地方是如何至关重要的。当然，一个主要的影响领域 *hardness of approximation* 在前一小节中已讨论过，我们列出了其他领域。

Verifying computations, and the power of the prover in interactive proofs 证明系统，在所有出现它们的设置和应用中，都坚持验证者必须是高效的；无论定理的证明者提出证明是多么优越或巧妙，一个普通人都可以检查它是正确的。但是，随着概率和交互式证明理论的发展，各种理论和实际考虑也质疑了证明者的能力。事实上，证明者可能需要比（复杂度理论上的）陈述的难度工作得更加努力。例如，根据定理10.3，可以证明co$\mathcal{NP}$陈述，但该定理中的证明者在$\mathcal{PSPACE}$ (中，我们怀疑它不可能比在#$\mathcal{P}$中更弱，而#$\mathcal{P}$是一个比co$\mathcal{NP}$)更高的类别。

这个问题在强用户（证明者）试图说服弱用户（验证者）其进行的计算正确时变得极其重要。这在 *delegation*中经常出现，其中弱用户（例如智能手机）让强用户（例如云）为其执行复杂的计算，但又不信任它。人们希望说服弱用户正确性并不比计算本身更困难。在第18.9.2节中讨论了已经取得很大进展的委托。随着量子计算的兴起，这种需求的出现也带来了一个截然不同的环境：如何让经典验证者相信一个所谓的量子设备执行的计算的正确性？同样，希望说服量子算法的正确性并不比计算本身更困难。量子验证，最近也取得了显著进展，在第11.3节中讨论。

Locality in error-correcting codes 由于PCP仅在几个坐标上进行查询，并且这种查询方式足以验证它们的正确性，因此它们对噪声具有极高的鲁棒性：在PCP的足够小的一部分坐标上更改有效的PCP，几乎肯定不会影响此类随机查询的答案，因此验证器会接受它。在这种对噪声的鲁棒性方面，PCPs类似于纠错码：

在某种意义上，一个有效的PCP可以从一个有噪声的PCP中导出，就像一个有效的码字可以从一个有噪声的码字中导出一样，如果噪声足够小。事实上，在PCP的设计中使用了编码理论技术。

然后影响发生了反转，大幅反转！PCPs的主要特征，即它们的局部可检查性，被引入到纠错理论中。在经典编码理论中，接收并处理的是整个被噪声损坏的消息。在这个新颖的*local*视角下，寻求并设计了新的纠错码，这些码允许在本地执行各种测试和正确信息提取。这导致了定义和构造（例如，参见所提供的参考文献及其指向的其他文献）以及局部可测试码[Gol11, GG16]、局部可解码码[Yek12]、局部可纠正码[DSW14a]、局部可恢复码[Maz18]、私有信息检索方案（PIRs）[CKGS98, DG16]以及许多其他方案。不用说，这些方案也找到了实际应用，以及与经典纠错码理论、计算复杂性和其他数学领域的理论联系，例如组合几何[KS09, DSW14b]。在这个广泛领域内，许多美丽的开放性问题仍然存在，其中最令人震惊的可能是有常数查询局部可测试码的线性块长和有常数噪声率的局部可解码码的多项式块长。

Property testing and sublinear-time algorithms 几个随机抽查的想法，以测试序列与全局属性（作为码字的属性）的邻近性，在[GGR98]中被大大推广，他们专注于图属性。这反过来又进一步扩展到测试集合、概率分布（与统计学相关）、超图、函数等的属性（邻近性）。关于这个主题的广泛文本是书籍[Gol17]。

并且，尽管上述研究关注的是在固定查询次数下可以做什么，但已经开发出技术，允许更普遍的概率算法采样远少于它们可用的所有输入数据，并使用这些信息来学习许多全局定义的性质。这与我们的数据爆炸时代（例如在生物学、物理学、互联网等）非常契合，如此之大以至于确实没有时间去全部覆盖。这个不断发展的领域被称为*sublinear-time algorithms*，例如参见调查[CS07, RS11]。

Black-box vs. white-box access 我们能否比仅仅访问它所计算的功能（即其输入输出行为）更有效地获取信息或利用一个 *given* 计算设备（比如一段软件代码或一块硬件）是一个理解计算的基本问题。第二种通常被称为 *black-box* 访问，因为我们无法窥视设备的内部工作。第一种通常被称为 *white-box* (或 *clear-box*) 访问，其中给出了计算设备的完整描述。本书的几个章节讨论了这一基本问题的影响，例如在第7.3.3节中关于伪随机性，以及在第18.9.3节中关于程序混淆。这对于所有有效的归约（参见第3.6节）都是基本的，这些归约本质上使用一个计算来创建另一个。黑盒与白盒访问的相对能力已在许多其他环境中得到研究。

巴拉克在密码学领域发现了白盒访问相对于黑盒访问的惊人力量，这本质上依赖于PCP定理。巴拉克的博士论文[Bar04]给出了几个巧妙而有力的演示，其中对计算*given in robust PCP form*的白盒访问可以极大地改变我们的效率，甚至影响我们实现某些密码学原语和任务的能力。这篇论文开启了“非黑盒密码学”领域，这超出了我们的范围。我们只是推测这个想法可能很好地应用于计算复杂性本身，甚至可能是下界证明。

Practical probabilistic proofs and their applications 虽然许多类型的概率证明源于潜在应用，尤其是在密码学和委托方面，但原始构造以及随后的构造过于复杂，甚至难以想象在现实系统中使用。令人惊讶的是，在过去十年中，这种情况开始改变！部分动机来自新的潜在应用，这些应用没有可信方，并且匿名至关重要。这包括数字货币（例如比特币、Zcash和以太坊）和公共账本，更重要的是作为其基础设施的底层（热门词汇）*blockchain*。这些新协议需要大量的理论研究工作，我们仅提供一些读者可以用来开始探索的参考文献：例如，参见[BSCTV17]关于零知识证明，

并且[BSBC+17]在PCPs上。不用说，当协议确实变得足够高效时，第18章中讨论的许多通用应用也可能成为现实！

但是，在其他意想不到的实用方向上发展起来的概率证明。我们限制本段仅涉及零知识证明，其种类令人震惊。事实上，也许这并不令人惊讶；那种无需透露信息就能令人信服的反直觉能力具有普遍的吸引力，读者应该很容易从他们的日常生活中找到这样的能力不仅有用，甚至至关重要的例子。确实如此！但是，与本章中我们讨论的协议不同，这些协议是数字的（即，由交换比特的计算机执行），在现实世界中，代理人可能是人，试图就物理对象的属性进行争论，并在他们的论点中使用物理对象和自然法则。一个引人注目的例子是核弹头减少验证协议，该协议在不可信的各方之间进行[GBG14]。另一个例子是用于私有比较DNA序列（例如，用于刑事调查或遗传检测）；[FFN14]中给出了这个例子以及许多其他例子，以及“物理”零知识证明的高级视角。

11 Quantum computing

本章描述了计算复杂性与物理学之间独特而令人兴奋的互动，探讨了现实的本质，并为研究带来了新的视角。我们将讨论这一互动的许多方面。关于这个广泛的主题有许多优秀的文献，包括[KSV02, NC10, Aar13a]，每本都有不同的视角和风格。

让我们回到最基本的问题：哪些问题可以高效地解决？在本书的开头，我们将它们定义为属于类别$\mathcal{P}$，那些可以在*deterministic*多项式时间内解决的问题。计算机技术的快速发展使它们运行得更快，但在所有模型下，类别$\mathcal{P}$仍然稳健。第一次潜在的变化发生在人们意识到我们在算法中使用*randomness*时。似乎自然界为我们提供了无限的免费随机比特，我们可以将它们纳入图灵机的计算中。这允许我们（如果我们愿意容忍小概率的错误）将可解问题的类别扩展到具有多项式时间*probabilistic*算法的类别，即类别$\mathcal{BPP}$。虽然我们不知道随机性是否真的带来了额外的力量（并猜测它没有——见第7.2节），但许多概率算法目前仍在使用，仅仅是因为我们目前拥有的最佳确定性算法要慢得多（通常是指数级慢）。

有理由认为我们应该将自然界提供的一切似乎能增强计算机和算法能力的东西添加到我们的计算机和算法中。确实，如果我们的计算机不能有效地模拟某些自然现象，我们应该在它们中集成使自然界更有效的底层机制。费曼[Fey82]指出，模拟n粒子量子系统演化的明显经典算法（甚至使用随机性）需要n指数时间。因此，他建议计算机算法应该配备“量子力学”门，以实现这种高效的模拟（并可能使它们比经典计算机更强大）。在俄罗斯，曼宁[Man80]提出了类似的想法。随后的一系列论文[Ben80, Deu85, Yao93, BV97, AKN98]完全形式化了量子力学图灵机的概念，我下面将非正式地描述它。将其限制在多项式时间内运行，我们得到由这类算法有效计算的功能类$\mathcal{BQP}$。观察表明，像$\mathcal{BPP}$一样，$\mathcal{BQP}$也允许小的误差（这些误差可以通过重复任意减小）。

Definition 11.1 (该类 $\mathcal{BQP}$ [BV97])**.** 函数 $f : \mathbf{I} \to \mathbf{I}$ 在 $\mathcal{BQP}$ 中，如果存在一个量子多项式时间算法 A，使得对于每个输入 x，$\Pr[A(x) \neq f(x)] \leq 1/3$。

我们将回到讨论这个类的能力。首先，让我解释一下量子算法是什么。

量子算法是如何工作的？结果是，要理解这一点，并不需要了解任何量子力学。[1] 让我们用我们已经遇到的确定性算法和概率算法来类比解释它们。从这些类型算法的整个“状态”随时间演化的角度来看，这样做是有用的。每个算法都会通过一系列局部操作（每个操作作用于几个比特）来演化状态——即其（比如说）n比特内存的内容。不同类型的算法在状态的性质、允许的局部操作以及该过程的输出定义上有所不同。在所有这些算法中，初始状态都包含在内存中指定位置的问题输入（所有其他比特设置为默认值，比如说，0）。

在确定性算法中，状态仅仅是这些位值的总和，一个向量 $x \in \{0,1\}^n$。单个操作可以选择其中的三个，并使用它们的当前值根据任何函数 $g : I_3 \to I_3$ 替换它们。例如，一个这样的 3 位函数，称为 *Toffoli gate*，将一个位位置写入其当前值与另外两个位 AND 的异或（此单个函数可以模拟标准的布尔门 $\vee, \wedge, \neg$）。当机器停止时，此计算的输出是某些指定位子集的内容。总结来说，确定性算法的演变发生在离散空间 $x \in \{0,1\}^n$ 中。

在一个概率算法中，局部操作可以是概率性的。例如，在一步中，它可能以概率应用Toffoli门到三个特定的比特位置，以概率 $\frac{3}{4}$ 保持比特不变。因此，算法随时间推移的状态是一个随机变量，可以看作是 *convex combination*[2] $\sum_{x\in\{0,1\}^n} p_x x$ 在 n-比特序列 x 上的一个。因此，局部操作演化的概率

[1] Indeed, quantum algorithms may be viewed as a language to explain many of the principles of quantum mechanics.

[2] Namely, $p_x \geq 0$ for all x, and $\sum_{x\in\{0,1\}^n} p_x = 1$. In other words, p is a nonnegative vector of unit norm in L_1.

向量 $p = (p_x)$ 随时间变化。输出再次位于某些指定的位位置，但现在是一个关于布尔向量的随机变量（在这些位置的分布边缘分布）。因此，概率算法的演化发生在 $\mathbb{R}^{2^n}$ 中，向量 $x \in \{0,1\}^n$ 作为这个空间的自然基。

在量子算法中，状态再次被视为一个线性组合（称为*superposition*或*wave function*）$\sum_{x\in\{0,1\}^n} \alpha_x x$，但现在系数可以取复数值，并且系数向量（称为*amplitudes*）$\alpha = (\alpha_x)$必须在L_2中具有单位范数.[3]局部操作可以像以前一样，取一些常数数量的比特（在此设置中称为*qubits*）并对其执行一个（保持范数的）*unitary*线性操作（这要正式成为对整个状态的行动，需要与剩余量子比特上的恒等算符张量积）。一个重要例子是Hadamard门，作用于一个量子比特。作为一个线性操作，我们可以通过其作用在基上的方式来描述它：它将状态**0**发送到 $(\frac{1}{\sqrt{2}}\mathbf{0} + \frac{1}{\sqrt{2}}\mathbf{1})$ 并将状态**1**发送到 $(\frac{1}{\sqrt{2}}\mathbf{0} - \frac{1}{\sqrt{2}}\mathbf{1})$。接下来需要定义算法的输出，它应该像输入一样再次是一个布尔向量。这是通过*measurement*获得的，我们现在定义它。为了简单起见，假设输出是内存的全部内容，最终状态是$\sum_{x\in\{0,1\}^n} \alpha_x x$在$\mathbb{C}^{2^n}$中。由于$\alpha$是一个单位向量，向量$p_x = |\alpha_x|^2$是一个概率向量，输出被定义为具有概率$p_x$的$x$.[4]总结来说，量子算法的演化发生在（单位球体）$\mathbb{C}^{2^n}$上，之后测量将最终状态转换为概率输出。

一些评论是必要的。首先，似乎可能门的数量是无限的，但实际上，正如经典计算一样，只需要有限的一组基本操作。确实，(经典) Toffoli门和(量子) Hadamard门一起构成一个通用门集。其次，[5] 结果表明，复数并不是真正必要的；实数就足够了，只要它们也可以 *negative*——正如我们将看到的，这似乎是量子算法相对于概率算法的力量的来源。最后，量子算法可以掷硬币：请注意，将Hadamard门应用于单个固定的量子比特（比如说，0），然后测量结果将得到一个完美的硬币掷出，其结果为0的概率为 $\frac{1}{2}$，为1的概率为 $\frac{1}{2}$。因此，量子计算可以模拟概率计算.[6] 这最后一点导致以下定理。

Theorem 11.2. $\mathcal{BPP} \subseteq \mathcal{BQP}$.

所以，$\mathcal{BQP}$中还有什么？Shor的突破性论文[Sho94]提供了最令人震惊的例子，整数分解和离散对数，在4.5节中讨论。

Theorem 11.3 [Sho94]. *Factoring integers and computing discrete logarithms are in* $\mathcal{BQP}$.

这些算法是如何工作的？它们做了哪些经典算法做不到的事情？量子算法和概率算法似乎“同时”作用于所有2^n二进制序列，所以这本身并不是力量的来源。时髦的答案是*interference*，这应该以你在高中学习的那种直观方式来理解，即（海洋、电磁、声音等）波相互干涉（建设性或破坏性）。具有负系数意味着量子算法可以在这种指数级上产生抵消，从而降低或消除不良结果的可能性，并通过单位性增加期望结果的可能性。这种干涉是概率算法中不可能发生的，因为这些算法的状态向量的所有系数都是非负的。因此，在实际应用中，概率算法只是演化概率分布的一个*sample*（而不是物理上维持指数级长的状态向量），而对于量子算法来说，整个叠加态必须“存在”并演化。

让我们尝试探索一下干涉的这种神奇力量，讨论Shor算法的一个关键方面。该算法的一个重要子程序计算了状态在指数大的阿贝尔群$\mathbb{Z}_N$上的*discrete Fourier transform* (DFT)。一个人如何计算一个指数大的线性变换

[3]Namely, $\sum_{x\in\{0,1\}^n} |\alpha_x|^2 = 1$.

[4]If the output is designated to only be a subset of the bits, we similarly give each Boolean output a probability that is the total square length in α of full sequences containing it.

[5]Which follows from the first point but can also be seen directly.

[6]While we only allowed a measurement at the very last step of quantum algorithms, one can define them alternatively to allow measurements at any step—this does not change their power—and so they can toss coins at any step.

在多项式时间内？让我通过一个更简单的子程序来演示它，这个子程序用于Simon算法[Sim97]在布尔立方($\mathbb{Z}_2$)^n（上计算DFT。事实上，这项工作启发了Shor)。该算法只需一行：简单地按顺序在每个n位上应用Hadamard门。请检查描述Hadamard门的n 2 $\times$ 2矩阵的张量积是描述($\mathbb{Z}_2$)^{n}上DFT的$2^n \times 2^n$矩阵。为了看到干扰的实际作用，考虑将DFT应用于所有状态具有相同振幅（这是$2^{-n/2}$）的状态向量。不难看出，结果是所有0向量具有非零（实际上，1）振幅，而其他所有向量的振幅变为零。因此，这个全0向量的振幅通过构造性干扰呈指数增长，而所有其他向量的振幅通过破坏性干扰（当然，这只是一个对线性代数中简单事实的直观描述）减小到零。以类似但远更复杂的方式，Shor在$\mathbb{Z}_N$（上应用DFT，其中N取决于要分解的整数)，以演化某个状态向量，干扰导致输出（集中在）该整数的非平凡因子（如果存在）具有大振幅，同时减小所有其他（指数级）替代状态。

什么是其他可以通过量子算法高效解决而经典算法未知的问题？Grover [Gro96] 的一般技术为抽象搜索问题提供了相对于经典算法的二次量子加速。但像Shor算法可能提供的可能超多项式差距，只有相对较少的额外例子，其中大部分也具有类似数论或代数风味（例如，参见调查[CvD10]和更近期的[EHKS14]）。在许多情况下，算法的本质是使用某些适当的群中的 *fast Fourier transform* (FFT) 算法找到某种周期结构，其中“快速”意味着（经典）时间 $N(\log N)^c$，而 N 是群的大小。上述量子并行性消除了通常是指数级的输入大小 n) 的因子 N (，从而得到一个时间复杂度为 (log N)^c = poly(n) 的量子算法。所有阿贝尔群都有FFT，并且自然地尝试将此类FFT推广到非阿贝尔群（这些群不是计算特征而是计算不可约表示）以解决其他问题。这类问题的关键挑战是图同构问题（本书已提及几次，例如，参见第4.2节）。虽然对于所涉及群（对称群 S_n）的（似乎必需的）量子FFT版本是已知的 [Bea97]，但我们不知道如何将其用于图同构的高效量子算法。

另一个具有挑战性的任务是发明新的方法来使用量子干涉，可能用于不同性质的问题。一种有趣的技术，尚未提出一种可能的量子指数加速超过经典算法，是在图上的*quantum walk*，由Farhi和Gutmann [FG98] 发起，随后有许多后续工作（参见[CCD$^+$03]，其中在黑盒模型中证明了指数差距），Aaronson和Arkhipov [AA11]提出了一个完全不同的建议，即利用线性光学中非相互作用玻色子的行为，有效地解决与永真函数相关的一定采样问题（这将在第12章中扮演重要角色），在自然假设下，没有有效的经典算法（甚至超出了$\mathcal{NP}$）。

总结来说，与经典算法相比，量子算法的力量远未得到理解。一方面，几乎没有人相信它们可以解决$\mathcal{NP}$-完全问题。用符号表示：

Conjecture 11.4. $\mathcal{NP} \not\subseteq \mathcal{BQP}$.

另一方面，几乎每个人都认为量子算法比概率算法更强，即：

Conjecture 11.5. $\mathcal{BPP} \subsetneq \mathcal{BQP}$.

我们只知道关于图10中定义在4.1节中的经典复杂度类$\mathcal{P}$和$\mathcal{PSPACE}$之间，高效量子算法的复杂度类$\mathcal{BQP}$的更精细位置的一点点信息。一方面，$\mathcal{BQP}$可以通过访问计数查询#$\mathcal{P}$ (有效地模拟，因为计数允许在量子态中添加指数级多的振幅)。另一方面，Raz和Tal [RT18]的最近突破给出了一种$oracle$[7]，使得$\mathcal{BQP}$能够解决超出多项式层次$\mathcal{PH}$的问题。

我们转向讨论$\mathcal{BQP}$的实用性。

[7]See Section 5.1 and this paper for the meaning of oracle separation.

11.1 Building a quantum computer

除了是一个伟大的理论成就外，Shor算法对量子计算产生了巨大的实际影响。回想一下，分解和离散对数是当今所有加密和电子商务系统的基本原理，因此每个人都想要一台量子计算机[8]。然而，这种力量也可以引发新的加密方案，抵御更强的攻击。这是量子密码学的主题，我们在这里不讨论。因此，在纯粹学术兴趣的量子计算领域经过二十年后，Shor的论文突然激励政府和行业投资数十亿美元来开发一台工作量子计算机，已经开发出一些显著的新技术，这些技术有助于克服重大障碍。衡量不断改进的技术和设计质量的一个具体（尽管有偏见）的方法是它们可以使用Shor算法分解的最大整数。今天，在Shor论文发表二十年后，迄今为止通过这种方式分解的最大数字是21。

在构建量子计算机时遇到什么问题？图灵定义了他的图灵机后，其设计的局部性和简单性立即暗示了实现方法。尽管当时技术庞大且存在故障（这是晶体管之前的时期——每个比特都需要自己的真空管！），但大规模的计算机很快就被构建出来了。如今，技术不可思议的快速发展导致了速度和内存的改进，使得去年的智能手机看起来像考古发现。但这种进步都是经典的。经典计算机的状态始终只是一系列比特。对于概率算法也是如此，它们只是简单地演化描述当前概率状态的分布的 *sample*。然而，如果我们想要 *interference*，就没有办法在不破坏信息的情况下（例如通过测量）采样量子计算机的状态。因此，困难的是在复杂的 *entangled* 状态中保持许多比特。当这个量子状态在计算过程中演化时，这变得更加困难。另一个需要克服的重大问题是 *decoherence* 噪声——在量子力学中，一切总是不可避免地与其他一切纠缠在一起，量子计算机的状态受到外部量子世界的影响。[Sho95, ABO97, Kit03] 和其他人发明的用于对抗退相干的量子纠错码将被使用。但尚不清楚它们在实践中是否足够，甚至它们所假设的噪声模型是否反映了物理退相干。世界各地许多项目中的各种巧妙技术正在竞争以实现这一目标，尽管在带来其他好处的技术突破中，向大规模通用量子计算机的进展仍然似乎缓慢。

存在一个完全不同的阻碍进步的问题吗？那会是什么？毕竟，量子计算机的存在与量子力学理论完全一致（实际上只使用了该理论的基本部分）。嗯，也许理论是错误的，或者至少是不完整的！也许量子力学需要修订以处理大量纠缠粒子，就像牛顿力学需要修订以处理以非常快的速度运动的物体，或在非常高的能量和非常小的距离（例如普朗克尺度）上相互作用的物体一样？这样的修订可能会限制量子计算机能做什么，并可能解释上述提到的缓慢进展。这是一个非常有趣的情况，而且似乎无论结果如何，我们都会更多地了解量子物理和计算机科学，因为这两个领域的科学家们联合起来解决这些问题。

但是为了激发你对即将到来的内容的兴趣，请考虑以下内容。当某个公司试图向你推销“量子计算机”时，你如何知道它是呢？更确切地说，一个经典算法如何验证由*computationally superior*量子算法执行的计算？这一基本科学问题在11.3节中进行了讨论。

尽管通用量子计算机尚未到来，但这些实际项目在量子计算技术方面取得的科技进步为进一步推进费曼的……

[8] We note that while a quantum computer can destroy some classical hardness assumptions for cryptography, it is not known to destroy them all (e.g., the ones based on lattice problems, such as those mentioned at the end of Section 13.8).

[9] Entanglement is the quantum analog of probabilistic correlation. Informally, just as the correlation among the bits in a probability distribution on bit sequences captures how far this distribution is from a product of independent distributions on individual bits, the entanglement among the qubits in a quantum state captures how far this state is from a tensor product of quantum states on the individual qubits. It should be stressed that entanglement can be far more complex, and is far less understood, than classical correlation.

[10] Such results often go by the name “quantum threshold theorem” to indicate that noise below a certain constant threshold per bit can be tolerated in arbitrarily long quantum computations.

原始动机，高效模拟 *some* 量子系统。此外，此类进步还产生了针对 *some* 密码问题的量子协议的实现，其性质源于量子力学，因此可以抵抗计算上无界的对手。这与针对相同问题的经典协议形成对比，后者需要依赖于计算假设，并且只能抵御计算上有限的对手。

11.2 Quantum proofs, quantum Hamiltonian complexity, and dynamics

我们回到数学领域，并以在以下方面取得显著理论进展的几个方向来结束本节。我不会详细阐述量子信息理论和量子密码学在现在非常发达的学科中的许多进展，这些学科还有实际应用（例如，量子隐形传态[BBC+93]和量子密钥分发[BB84,Eke91]的理论思想正在成为现实）。相反，我希望关注量子复杂性理论，即我们在前几章中看到的应用于泛化、简化和完备性的方法。这项研究通常与物理学家有美丽的互动，并且对物理学问题以及科学哲学也有直接影响。

一个在经典世界中研究的中心概念是 *proof*，并且拥有一个高效的计算新模型，将证明（以及其他概念）扩展到这个环境是自然的。量子世界中的证明是什么？最基本的概念是 $\mathcal{NP}$ 到量子环境的自然推广，其中验证者是 $\mathcal{BQP}$ 机器，而证据允许是一个（短的）量子状态。所有具有此类验证系统的所有问题的类被称为 $\mathcal{QMA}$。[11] 类比于 Cook 和 Levin 发现一个自然的 $\mathcal{NP}$-完全问题，SAT，Kitaev [Kit03, KSV02] 发现了一个 $\mathcal{QMA}$-完全问题，从计算机科学和物理学的角度来看都是自然的。它可以自然地看作是一个量子 *constraint satisfaction problem (CSP)*，类似于 3-SAT，而且，局部约束自然地来自量子哈密顿量物理。我们注意到，尽管在句法上相似，但 Kitaev 的量子简化比 Cook-Levin 的经典简化要复杂得多。原因是对于经典状态，局部一致性（计算的演化）意味着全局一致性，而在量子设置中，两个全局非常不同的叠加态在局部看起来可能相同（例如，当投影到每个三个量子比特时）。尽管如此，局部一致性检查最终证明是足够的，我们现在将讨论 Kitaev 的工作的一些后果和延续（详细信息可以在例如调查 [GHL14] 中找到）。

我们首先定义量子哈密顿量和量子 *local* 哈密顿量的概念。首先，一个 *Hamiltonian* (on n 量子比特)[12] 简单地是一个厄米 $2^n \times 2^n$ 矩阵 H。在量子力学中，哈密顿量通过薛定谔方程 $i\hbar\frac{\partial\Psi}{\partial t} = H\Psi$ 定义了 n-体量子系统状态 Ψ 随时间 t 的动力学：H 是 *local*，如果它是 $H = \sum_i H_i$ 的和，其中每个 H_i 是作用在固定数量量子比特上的厄米矩阵（就像局部门在量子算法中的作用方式）。请注意，每个这样的 H_i 是一个“局部约束”——它本质上是一个固定大小的矩阵[13]，因此可以简洁地描述：命名它作用的量子比特的索引以及它如何作用在这些量子比特上。许多局部哈密顿量源于量化局部相互作用统计力学模型，如二维晶格上的伊辛自旋玻璃。凝聚态物理学中的一个中心问题是确定其最低本征值（*ground state energy*）以及更一般地，理解其本征向量（*ground state* 本身）。我们依次讨论这两个问题，以及计算视角如何影响它们在物理学中的研究。

11.2.1 The complexity of ground state energy

Kitaev上述关于量子CSP对$\mathcal{QMA}$完备性的结果，正是关于计算（或者更确切地说，近似）局部哈密顿量的基态能量。一类广泛量子CSP的重要参数包括*geometry* (哪些变量子集相互作用)，*locality* (此类子集的最大大小)，以及*dimension* (每个量子比特在H_i的)中可以取的值的数量。例如，Kitaev的是维度为2的5局部CSP。几十年来，物理学家们研究了在以下出现的众多量子CSP：

[11] It would seem that $\mathcal{QNP}$ is a better name, but there is a reason for this notation, which I do not go into here.
[12] Or more generally *qudits*, which can take more than 2 values.
[13] Which is formally tensored with a huge identity matrix trivially acting on the remaining qubits.

各种自然设置。但计算透镜对它们的研究方式产生了重大影响。为了理解为什么以及如何，首先观察任何经典CSP（如3-SAT）都可以被视为量子CSP。这表明，我们在经典设置中拥有的归约和完备性结果可以扩展到量子世界，从而理解各种量子CSP的相对复杂性（并且像经典设置中的优化问题一样，揭示了许多令人惊讶的联系）。这种归约，通常在物理上非常不同的量子系统的哈密顿量之间，催生了*quantum simulations*[CZ12]这一领域，它允许通过这种归约使用一个量子系统研究另一个。大量论文精确地确定了寻找许多物理重要哈密顿量的基态能量的复杂性以及它如何依赖于哈密顿量的参数。这项研究导致了对量子CSP在许多情况下算法难度的相当完整的描述。这类一个非常一般的结果（参见[CM13]，由[BH14]完成）确定了所有二维CSP（即，在量子比特上）的复杂性。事实证明，这些可以是$\mathcal{P}$，$\mathcal{NP}$-完备，$\mathcal{QMA}$-完备或“伊辛完备”。这为经典CSP在第4章中讨论的*dichotomy theorem*定理4.3提供了量子对应。自然地，也研究了近似算法和困难性。

11.2.2 Ground states, entanglement, area law, and tensor networks

现在我们来理解并计算局部哈密顿量的基态本身。首先让我们思考：这个问题究竟意味着什么？毕竟，如前所述，对于一个包含 n 个量子比特的系统，这实际上是一个 2^n 维的向量（更不用说系数是复数了）。因此，为了高效，我们只能希望计算基态（或其近似）的某种简洁表示，假设它存在。物理学家们提出了这样的简洁表示，特别是 *tensor networks*，它们允许计算局部可观测量（例如，状态的能量）。不正式描述它们（参见这篇友好的物理调查 [Orú14]，以及这篇以计算为中心的调查 [AKK17]），它们可以被视为计算设备（如电路），它们“计算”基态就像电路计算布尔函数一样。在这两种情况下，这些对象的明显描述长度是指数级的 n，但对于某些基态，就像某些函数一样，计算描述可能要简洁得多（例如，多项式级的 n）。就像计算复杂性一样，理解哪些局部哈密顿量具有这样的描述，以及进一步从哈密顿量的描述中有效地找到这些描述，是核心问题。

一个高效的张量网络必然限制了它所表示的状态中的纠缠；其几何学意味着某些粒子的子集不能与其他子集纠缠得太厉害。因此，一个更基本的问题是：哪些哈密顿量具有有限纠缠结构的基态？从逻辑上讲，哈密顿量的局部相互作用的几何学会影响这些量子相关性的结构（并可能为张量网络的建设提供信息）。一个被称为*area law*猜想的主要猜想断言，对于任何*gapped*系统，[14]系统内任意两组粒子之间的基态纠缠与*area*（成比例，或者从图论的角度来看，是局部相互作用图部分之间的切割大小)。因此，在1维系统（其中量子比特位于一条线上，所有相互作用都是在这条粒子路径上的相邻点之间）中，纠缠应该被一个常数所限制。在2维系统（例如，量子比特位于平面晶格的顶点或更一般地，位于任何平面图的顶点）中，它应该像$\sqrt{n}$那样扩展。如果我们有一个任意的相互作用图，面积是通过两个部分之间的相关切割中交叉的边数来衡量的，根据面积定律猜想，这限制了它们之间的纠缠。一般的面积定律猜想是开放的，但有一些令人兴奋的发展和相互作用，我们现在简要讨论。

首先，让我们讨论小张量网络的存在性，然后转向寻找它们。Hastings [Has07] 的重要成果是证明了1维系统的面积定律。进一步的工作 [AAVL11]，使用为PCP定理的量子类似物开发的某些计算方法（见第10.3节），

[14] “Gapped” means that there is a constant spectral gap between the ground and the second-lowest energy levels of the Hamiltonian $H = \sum_i H_i$, when each local term H_i in it is normalized to have at most unit norm. This is a different regime than the $\mathcal{QMA}$-complete problems, where the gap is typically inverse polynomially small.

提供了对Hastings界的一个指数级改进.[15] 结果表明，Hastings的结果暗示了存在一个小（多项式大小）张量网络（对于一维系统称为*matrix product state*）。这引发了如何有效地找到这种描述的问题。事实上，这个重要问题被考虑得较早，并开发了几个启发式算法，其中最流行的是White [Whi92] 开发的密度矩阵重整化群。DMRG及其变体是许多自然发生和研究的物理多体系统的极其有用的启发式方法，但它们的性能没有理论解释或可证明的界限.[16] 一系列使用各种计算技术的论文最终导致了[LVV13]，该论文开发了一种完全不同的、可证明的多项式时间算法，该算法构建了一个矩阵乘积态，该态逼近每个有隙一维哈密顿量的基态。

11.2.3 Hamiltonian dynamics and adiabatic computation

让我们以对量子哈密顿量 *dynamics* 的简要讨论来结束这个概述，正如在薛定谔方程中一样。结果发现，这种动力学暗示了一种新的量子计算方法，由 [FGGS00] 提出，称为“绝热计算”。它基于Born和Fock的绝热定理 [BF28]，这是量子力学中早期和基本的结果之一。我给出绝热计算的高级描述，特别是为了与我们对量子图灵机计算方式的描述进行对比。根据基塔耶夫的完备性结果，本质上任何我们想要解决的问题都可以编码为寻找给定局部哈密顿量的基态能量，例如，在 n 位上（为了简单起见，你可以考虑哈密顿量编码了 SAT 的一个实例）。称这个哈密顿量为 $H(1)$。接下来，“准备”一些简单的 n 位局部哈密顿量，并将其初始化为其基态。称其为 $H(0)$。最后，让 $H(t)$ 描述 $H(0)$ 到 $H(1)$ 的演化，这可以是两者之间的任何连续插值。绝热定理确保，如果这种变形足够慢，$H(t)$ 将在整个过程中保持在它的基态，因此我们将得到 $H(1)$ 的基态！这将解决我们的原始问题（例如，它将给出编码 SAT 公式的满意赋值）。这种设计的巧妙之处在于选择初始 $H(0)$ 和演化路径 $H(t)$，使得所有哈密顿量 $H(t)$ 的基态能量 $e(t)$ 都比下一个更高能级的能量 *well separated* 低，例如，通过某个 $\gamma > 0$。根据绝热定理，只要演化速度足够慢，t 就能在时间 $1/\gamma^2$ 内从 0 移动到 1。这个模型能否在多项式时间内解决 SAT？能否解决整数分解？

第一个值得注意的现象是量子力学提供了许多计算方法，这些方法彼此之间看起来非常不同。下一个值得注意的发现是一个定理[AvDK+08]，它表明这两种计算模型是等价的！

Theorem 11.6 [AvDK+08]。*Adiabatic computation and quantum circuits can simulate each other efficiently.*

与经典计算理论早期一样，当时发现了许多不同类型的计算模型（1磁带图灵机、多磁带图灵机、λ演算、随机存取机、细胞自动机等）被认为是等价的，定理11.6使我们在量子设置中拥有正确的计算模型也充满信心。确实，另一个非常不同（但多项式等价）的通用模型是Freedman、Kitaev、Larsen和Wang [FKLW03] 提出的*topological quantum computer*，其中由*quantum braids*构成的逻辑门作用于称为*anyons*的二维准粒子。[17]这个拓扑模型具有强大的容错特性，并成为构建量子计算机的一些实际项目的基石。

[15] The conjecture is wide open for 2-dimensional lattices. A very nice, seemingly much simpler challenge is to extend this constant upper bound on entanglement from paths to trees, where all cuts are still of size 1.

[16] This may be likened to the success of the simplex method for linear programming on many practically arising systems of linear inequalities. And like that story, where eventually a completely different algorithm (the ellipsoid method) was found to solve linear programming on all instances in polynomial time, here too there was a (theoretical) happy ending.

[17] Fear not—I too don't understand the last sentence.

11.3 Quantum interactive proofs, and testing quantum mechanics

正如可以推广 *written* $\mathcal{NP}$-证明一样，可以尝试推广第10.1节中讨论的 *interactive* 证明，并研究它们的效力。已经定义了交互式证明系统的各种类似物，从而以经典复杂度类和经典定理的类似定理为术语，对这些类进行了新的特征描述（例如，参见[BFK10, JJUW10, IV12, BJSW16]）。[18] 但关于交互式证明的最引人入胜的发展之一是在研究以下“混合”证明系统时发现的。在这个系统中，证明者不是任意强大的（例如，在 $\mathcal{IP}$ 中），而是一个有效的量子算法（即在 $\mathcal{BQP}$ 中）。这个系统中的验证者是一个经典算法（在 $\mathcal{BPP}$ 中）。让我们讨论这种混合证明系统的基本动机。

Is quantum mechanics a falsifiable scientific theory? 这个基本问题应该对任何理论提出，研究它的通用科学范例有时被称为“预测和实验”。任何理论都预测某些实验或观察的结果。当观察结果与预测相符时，我们进一步验证该理论。如果它们不相符，则该理论是错误的，需要修正。量子力学，由于其悖论性和反直觉性，已经产生了一系列建议的实验，并且实际上经受住了许多。爱因斯坦，他一生都拒绝相信它，设计了众多这样的实验，其中最著名的是1935年由爱因斯坦、波多尔斯基和罗森提出的EPR“悖论”[EPR35]。他们挑战了量子力学提出的可能性，即量子信息似乎被维持并瞬间传递，速度超过光速。在20世纪60年代，贝尔的著名不等式[Bel64]提出了一种具体的实验（由[CHSH69]简化）来反驳量子力学的“局域隐变量”解释。从20世纪80年代开始，这些实验得到了成功实施（参见[Asp99]），在某种程度上证实了量子力学确实是悖论性和反直觉的，或者至少违反了简单的经典概率解释。尽管量子力学已被完全接受，并产生了巨大的技术影响，但关于其解释的哲学争论至今仍在继续。如上所述，大规模通用量子计算的可能性是对这一理论完整性的真正挑战。一个人如何测试这一点?

好吧，如果确实存在某些问题，量子算法比经典算法快指数级，那么每个这样的算法都建议进行一项新的实验，并且随之而来的是一个真正的困境。取任何具有快速量子算法但没有经典算法的函数。要测试的预测很简单，即给定的量子算法在某个输入上计算出正确答案。困境是如何通过有效的经典手段来测试这一事实，而对于这个正确答案，根据假设，是不可能获得的。这不仅仅是一个哲学问题！如果多个试图构建量子计算机的项目之一声称成功（就像一些声称使用量子算法的计算机公司所做的那样），这可能会成为一个真正的问题。我们如何测试这些说法? 当然，对于某些问题，所谓的量子算法可以通过经典手段有效地进行测试。例如，Shor的分解算法很容易验证，因为它产生了可以由经典验证器相乘并检查的因子。更普遍地说，如果量子算法产生了必要的证据，这种经典测试可以用于任何在$\mathcal{NP} \cap \text{co}\mathcal{NP}$中的函数。但是$\mathcal{BQP}$可能包含许多更难的问题（例如，在$\mathcal{NP}$之外）。我们如何测试声称解决这些问题的量子算法（或设备）?

一个新想法，在[BFK09, ABOE10]（参见[ABOEM17]中的完整证明和历史调查）中独立提出，即在交互式证明的精神下允许{v*}实验。也就是说，假设一个$\mathcal{BQP}$算法可以解决某个问题。是否（可能另一个）$\mathcal{BQP}$算法可以与一个经典$\mathcal{BPP}$算法交互，并使其（以高概率）相信它正确地解决了这个问题? 对于*all*问题在$\mathcal{BQP}$中是否可以进行此类交互验证? Aharonov，Ben-Or和Eban[ABOE10]证明，答案基本上是“是”！（这些证明系统中的验证者并非完全经典，但几乎是——已知最好的一个只使用包含单个量子比特的寄存器[Bro15]）。Mahadev[Mah18]最近的一项突破表明，在自然的密码学假设下，一个完全经典的验证者可以检查任意量子计算!

[18] We do not have as yet a satisfactory analog of the PCP Theorem 10.6 (see [AAVL11] for the (subtle) formal definitions and some initial results—far more has happened since).

我们强调，通过交互式程序测试科学理论这一基本、自然的思想，在其他环境中也具有潜在的强大作用。

11.4 Quantum randomness: Certification and expansion

一个人工智能翻译引擎，专业、真实。以下为翻译结果：One of the most personally satisfying developments in the interaction between computational complexity and quantum mechanics has been the recent flurry of papers on certification and expansion of randomness, which connect several important notions discussed in this book. Let me elaborate. I have given many lectures on pseudo-random generators and on randomness extractors (the topics of Chapters 7 and 9) to varied audiences. Recall that (on top of many theoretical side benefits) these two theories explain how we can salvage the amazing utility of perfect randomness (namely, of independent, unbiased coin flips, which we hypothesize for probabilistic algorithms) in a world that may not have it.[19] In these lectures, the most typical question I get from physics-minded audience members is: "Why bother worrying about such hypothetical worlds? In our world, quantum mechanics suggests simple devices producing a stream of perfect random bits. Don't you trust quantum mechanics?" This is indeed an excellent question, with an excellent answer, related to what we have just discussed above: "Even if I do trust the theory, why should I trust the actual devices?" The basic question here is: How can you test that a given distribution (from which you can only sample) is perfectly random, or has entropy at all? 以下为翻译结果： 个人在计算复杂性与量子力学相互作用中最令人满意的进展之一，是最近关于随机性认证和扩展的论文热潮，这些论文将本书中讨论的几个重要概念联系起来。让我详细说明一下。我向不同听众讲授了许多关于伪随机生成器和随机提取器（第7章和第9章的主题）的讲座。回想一下（除了许多理论上的好处之外），这两个理论解释了我们在可能没有完美随机性的世界中如何挽救完美随机性的惊人效用（即独立、无偏的硬币抛掷，这是我们为概率算法假设的）。这些讲座中最常见的问题是：“为什么要担心这样的假设世界？在我们的世界中，量子力学表明简单的设备可以产生完美的随机比特流。你不信任量子力学吗？”这确实是一个非常好的问题，有一个非常好的答案，与上面我们刚刚讨论的内容相关：“即使我信任这个理论，为什么我应该信任实际的设备？”这里的基本问题是：你如何测试一个给定的分布（你只能从中采样）是否是完美的随机分布，或者是否具有熵？

以下为翻译结果： 个人在计算复杂性与量子力学相互作用中最令人满意的进展之一，是最近关于随机性认证和扩展的论文热潮，这些论文将本书中讨论的几个重要概念联系起来。让我详细说明一下。我向不同听众讲授了许多关于伪随机生成器和随机提取器（第7章和第9章的主题）的讲座。回想一下（除了许多理论上的好处之外），这两个理论解释了我们在可能没有完美随机性的世界中如何挽救完美随机性的惊人效用（即独立、无偏的硬币抛掷，这是我们为概率算法假设的）。这些讲座中最常见的问题是：“为什么要担心这样的假设世界？在我们的世界中，量子力学表明简单的设备可以产生完美的随机比特流。你不信任量子力学吗？”这确实是一个非常好的问题，有一个非常好的答案，与上面我们刚刚讨论的内容相关：“即使我信任这个理论，为什么我应该信任实际的设备？”这里的基本问题是：你如何测试一个给定的分布（你只能从中采样）是否是完美的随机分布，或者是否具有熵？

如果我们的测试更复杂（例如，我们被允许输入一些（可能是随机的）输入，B 的输出可能依赖于这些输入），这个论点仍然成立。

因此，我们需要对B的操作方式做出一些假设。[20]在这里我们相关的新发现是，一个最自然的物理假设，实际上比完整的量子力学要弱得多，就足够了。这个（经典！）假设被称为*no-signaling*。它假设B实际上由两个盒子B_1和B_2组成，这两个盒子以下列强烈的意义不进行通信：每个的输出分布独立于另一个的输入[21]。通过适当的空间分离，在原则上可以实现设备之间的无信号传输，我们将假设这一点。

因此，我们将尝试通过游戏从（比如说两个）无信号盒中提取 *certified* 随机性。关键将是这些无信号盒可以实施的策略集。在无信号设置中，量子策略相对于经典策略的优势是贝尔不等式及其通过 [CHSH69] 简化的关键，这关乎 EPR 疑难和隐变量理论。事实上，这种优势通过原始 CHSH 游戏 [CHSH69] 得到了完美的展示，我们现在来描述它。想象一个验证者分别向 B_1 和 B_2 发送独立的非偏置比特 x_1 和 x_2，他们分别以比特 y_1 和 y_2 响应。如果 $x_1 \wedge x_2 = y_1 \oplus y_2$（验证者可以轻松检查的事件）发生，我们说盒子“赢得”游戏。如果他们的策略必须是无信号的，盒子赢得游戏的最大概率是多少？这是一个简单的练习

[19] It is quite possible that our world is completely deterministic (providing no randomness at all), or that it has only weak randomness (some entropy is there somewhere, but the "coin flips" we can measure are arbitrarily biased and correlated). But these two respective theories "deal with" such possibilities!

[20] The so-called "no free lunch theorem" says that we must always pay (with assumptions) for what we get. Indeed, for the two theories of randomness above, we postulated (respectively) computational assumptions for pseudo-randomness in a world with no randomness, and postulated the availability of (very few) truly random bits for randomness extractors in a world with "weak" randomness.

[21] A similar definition exists for more than two boxes, which will be needed later. Basically, the joint output distribution of any subset of boxes is independent of the joint inputs to the rest. As an aside, we note that this notion was used in a recent classical result, providing another demonstration of the power of "quantum ideas" in the classical setting. The reader may recall that no communication between *provers* was essential to the multi-prover system $\mathcal{MIP}$ of [BOGKW89] mentioned briefly in Chapter 10 as a precursor to PCPs. The new powerful PCP theorem of [KRR14] for no-signaling provers allows ultra-fast, trusted *delegation* of computation to powerful entities. Other examples of the power of "quantum ideas" in classical settings can be found in the survey [DdW09].

要检查它们的策略是否是经典的（即，每个输出一个依赖于其输入的概率分布），那么最优策略以概率 .75 获胜。相比之下，无信号是一个如此弱的要求，以至于它为盒子提供了一个简单的联合策略（读者被邀请找到），以概率 1 获胜。令人印象深刻的是，设计量子策略（即，当盒子共享一个纠缠的量子比特对时）并不太难，这些策略可以通过非常简单的量子设备轻松实现，允许盒子以概率 $\cos^2(\pi/8) \approx .853$ 获胜（这碰巧是 Tsirelson 界限 [Tsi93] 下量子策略的最优值）。这种由贝尔不等式（特别是这个游戏所揭示的）产生的经典与量子能力之间的差距，几十年来一直是展示量子力学如何反直觉（以及可能是不完整的）的一个例子，如何局部隐变量理论无法解释它，如何量子力学产生的非交换概率理论与经典力学的交换概率理论不同，等等，等等——所有这些都是我们理解我们实际生活的这个迷人世界的核心。

然后，Colbeck在他的2006年博士论文[Col06]中发现了这样一个经典-量子差距，如这个游戏所展示的，另一个基本方面。他做出了以下观察。如果你反复与无信号盒子玩CHSH游戏，并且它们以超过75%的概率持续获胜，那么它们的输出$must$包含熵（超出它们输入提供的熵）。为了看到这一点，请注意，否则，它们正在使用确定性策略，这在特定情况下是经典的。因此，不需要信任盒子：我们可以在大量实验中测试获胜的统计数据，如果（比如说）它超过80%，我们就接受它们产生随机性（否则宣布它们有故障）。简而言之，随机性可以被认证。这是一个了不起的洞察！

现在你可能开始抱怨我们想要一串独立、无偏见的比特，而我们最多只能得到分数熵。你也可能抱怨我们不得不投入比我们得到的更多（且完美）的随机性，并质疑整个练习的意义。但然后，对于这两个问题，你回想起第9章关于随机提取器的内容，并看到了光明。事实上，所有这些都是在一系列快速发表的论文中完成的，从Pironio等人[PAM+10]开始，表明一个few完美的随机比特可以被扩展成$many$，几乎均匀的比特。最新的可用结果[MS14a, CY14, MS16]实现了人们所能期望的最好结果：无界扩展，误差最小。更确切地说，一定数量的无信号盒可以$certifiably$生成，输入种子为k真正随机的比特，一个在任何数量n的输出比特上的分布，该分布与2^{-k}指数接近于n比特上的均匀分布。

这些构造和证明相当复杂，我在这里只做了一些评论。一点是种子不同部分将具有不同的功能。一是隐藏在许多CHSH游戏中将用于生成输出的子集。二是用于将当前熵转换为均匀分布的随机提取器的正常种子。另一个重要点是，在这个过程中，盒子被反复重复使用以进一步扩展随机性，使用旧输出作为新输入。因此，似乎盒子可以使用它们的内存和共享纠缠来在最终输出中生成相关性。防止这一点很微妙，需要量子信息理论的新精致技术。最后，这种经过认证的随机性的新能力在量子密码学中至少有一个重要的用途，即用于量子密钥分发[MS14a]的设备无关安全性。

12 Arithmetic complexity

我们现在离开布尔域，讨论在任意域上计算 *polynomials* 的问题。多项式由于其基本性和实用性，在数学的各个领域都得到了研究。在这里，我们研究它们的复杂性：计算自然多项式（例如，基本对称多项式、行列式、永序、矩阵乘法、卷积）需要多少算术运算？这种研究从数学和实用角度来看显然是自然的。但是，一旦应用了计算复杂性中的归约和完备性机制，它就导致了 $\mathcal{P}$ 与 $\mathcal{NP}$ 以及其他在这里似乎比布尔世界更容易解决的问题的类似问题。这有几个原因。计算形式多项式比计算它们定义的函数要严格得多；与布尔世界相比，关系更少（例如，$x^2 = x$），这限制了算术计算的能力；最后，有更多的数学工具可用，主要来自代数。事实上，与布尔电路复杂性相比，算术电路复杂性的进展速度更快，有令人兴奋的新发展。因此，即使是最近的综述[SY10, CKW11]也不是完全最新的（但确实提供了这里大部分材料以及更多内容的详细说明和证明）。范围比我们在这里讨论的更广泛的广泛书籍有[BCS10, vzGG13]。

在这一章中，我们使用相同的[1]符号$\mathsf{S}(f)$来表示一个计算*polynomial* f的最小大小*arithmetic*。我们将在下面一般性地对多元多项式形式给出正式定义。但首先，让我们发现这类问题，即使是对于一元多项式，其难度、深度和意外联系。

12.1 Motivation: Univariate polynomials

考虑 $f(x) \in \mathbb{F}[x]$ 的次数 d。从 x 和任何常数 $\mathbb{F}$) 开始计算 f (需要进行多少次加法和乘法? 即使这个问题很简单，也是非平凡的。一个巧妙的上界是 *Horner's rule*，它给出 $\mathsf{S}(f) = O(d)$ (尝试证明它，或者查看脚注)。[2] 另一个非平凡的事实是存在性下界，表明 *some* 多项式需要 $\mathsf{S}(f) = \Omega(\sqrt{d})$。

但是，一些多项式计算得更快。例如，考虑 $g(x) = x^d$。显然，$\mathsf{S}(g) = O(\log d)$，因为我们可以计算连续的幂 $x, x^2, x^4, x^8, \ldots$，然后乘以必要的子集以获得 x^d。这也显然是 $\mathsf{S}(g) = \Omega(\log d)$；确实，至少需要 $\log d$ 次乘法。令人惊讶的是，以下问题是开放的。

Open Problem 12.1. 描述一个度数为 d 的多项式 f，使得 $\mathsf{S}(f) \neq O(\log d)$.

一个自然的猜测是，上面的 g 很容易，因为它有多个根。考虑一下多项式 $h(x) = (x-1)(x-2)\cdots(x-d)$，它有 d 个不同的根。它在有理数 $\mathbb{Q}$ 上的复杂度是什么? 除了明显的对数 $d \leq \mathsf{S}(h) \leq d$，没有其他已知信息。但是，强大的上界有惊人的后果——分解整数是容易的！提示：这种联系被称为“阶乘与分解”。Lipton [Lip94] 描述了这些以及更一般的结果，其中一些想法可以追溯到Shamir [Sha79a]。

Theorem 12.2 [Sha79a, Lip94]. *If* $\mathsf{S}(h) \leq (\log d)^{O(1)}$, *then integer factoring is in* $\mathcal{P}/\mathrm{poly}$.

12.2 Basic definitions, questions, and results

我们将考虑 $\mathbb{F}[x_1, x_2, \ldots]$ 中的多项式。定义和大多数问题在任何域 $\mathbb{F}$ 上都是有趣的。然而，一些结果仅对某些“足够大”的域成立，或者更具体地说，具有零特征或代数封闭。另一方面，我们将讨论的大多数多项式都将具有 0/1 系数，因此它们在任何域上都有意义。为了具体化，读者可以假设 $\mathbb{F} = \mathbb{C}$ 或 $\mathbb{F} = \mathbb{Q}$。

[1] As for Boolean circuits.

[2] Hint: Any degree d polynomial can be written as $a_0 + x(a_1 + x(a_2 + x(\cdots + x(a_d))))$, involving d additions and d multiplications.

布尔代数电路复杂度之间存在许多相似之处和差异，你可能想回顾第5.2节。我们首先定义大小。

就像在布尔电路中一样，一个 *arithmetic circuit* (over $\mathbb{F}$) 是一个有向无环图，其中非输入门用算术运算 $+$ 或 $\times$ 标记。也就是说，每个门输出多项式，即其两个输入多项式的和或积。输入节点可以用变量 x_i 标记，也可以用 $\mathbb{F}$ 中的任何常数标记。因此，例如，可以使用具有 3 个输入、一个乘法门和一个加法门的电路计算多项式 $\pi x + y$。电路的 *size* 简单地是其图中导线的数量。一个 *arithmetic formula* 是其图是树的电路。允许使用常数解释了为什么我们不需要一个特殊的减法门。我们很快还会讨论除法。

显然，每个多项式 $f \in \mathbb{F}[x_1, x_2, \dots]$ 都可以通过一个电路（和一个公式）来计算。我们用 $\mathsf{S}(f)$ 表示 f 的最小电路大小，用 $\mathsf{L}(f)$ 表示最小公式大小。我们显然有 $\mathsf{S}(f) \leq \mathsf{L}(f)$。注意，我们将算术电路视为计算 *formal* 多项式，而不是它们定义的函数。这是有限域上的一个区别。例如，形式多项式 x 和 x^p 是不同的，尽管它们在 $\mathbb{F}_p$ 上的函数是等价的，因此第一个的大小复杂度是常数，而第二个需要大小 $\geq \log p$。

与布尔电路复杂性一样，算术电路复杂性是一种渐近理论。我们通常会有一个由 n 参数化的多项式序列 $f = \{f_n\}$，其中 n 通常表示变量的数量或与之多项式相关的量。例如，行列式多项式将是一个序列 $DET = \{DET_n\}$，其中 $DET_n(X)$ 是 n^2 个变量的 $x_{i,j}$ 矩阵 $n \times n$ 的行列式多项式。需要注意的是，与布尔设置不同，一个多项式还有一个输入参数，其复杂性可能依赖于它，即它的 *degree*。几乎整个理论都处理多元多项式，为了集中关注一个参数，我们将坚持认为 f_n 的（总）次数最多是一个关于 n 的固定多项式。实际上，这几乎不会是一个限制，因为我们的多项式中的大多数将是 *multilinear* (，即每个变量在每个次数最多为 1 的单项式中都出现，因此总次数将自动被变量的数量所限制。

算术下界难以证明。确实，即使考虑香农定理5.6的算术类似物，它也证明了存在需要指数级大小电路的布尔函数。回想一下，它使用了计数论证——函数的数量比小电路多。然而，在无限域$\mathbb{F}$上的算术电路中，尽管算术电路的“骨骼”数量是有限的，但它们使用$\mathbb{F}$中的任意常数的 ability 使得它们的数量是无限的。当然，我们潜在困难多项式的单项式系数的数量也可以以无限多种方式选择，但我们承诺只考虑0/1系数的多项式，其中（限制变量数量和次数）只有有限多个。尽管如此，Hrubeš和Yehudayoff [HY11]证明了以下定理。

Theorem 12.3 [HY11]. *For every field $\mathbb{F}$, almost all multilinear polynomials f on n variables*[3] *with* $0/1$ *coefficients require* $\mathsf{S}(f) \geq 2^{n/10}$.

证明（给出了更一般的结果）用维度论证取代了计数论证，并诉诸于基本的代数几何。事实上，不出所料，诸如贝祖定理和代数超越论证等工具在此代数设置中至关重要，并且它们也用于我们所知的少数显式下界。我们将在下一节中讨论这些内容，但亮点是显式的多线性多项式 f, g 在 n 变量上，需要 $\mathsf{S}(f) \geq n \log n$ 和 $\mathsf{L}(g) \geq n^2/\log n$。因此，特别是在算术设置中，我们确实有（略微）超线性的多项式电路大小下界，其次数随着变量数量的增长而增长。这里有一个具有挑战性的开放问题，对于这个问题，上述代数几何工具似乎不足够。

Open Problem 12.4. 查找显式 *constant-degree* n-变量的多项式 f，使得 $\mathsf{S}(f) \neq O(n)$。

我们最后涉及的基本方面是电路和公式的相对能力。回想一下，在布尔设置中，我们认为电路比公式指数级强大。在算术电路中，它们是

[3] And so of degree $\leq n$.

与功率更接近。Valiant、Skyum、Berkowitz和Rackoff [VSBR83] 的重要结果表明，算术电路可以接受所谓的深度降低。也就是说，每个计算度-d多项式的电路可以被“压缩”为只有$O(\log d)$次加法和乘法门之间的交替，而不会显著增加其大小。对于我们关心的多项式（即，度数为 $d = n^{O(1)}$），公式大小最多是电路大小的准多项式。我们只陈述关于公式大小的推论，这是Hyafil [Hya79] 之前证明的。

Theorem 12.5 [Hya79]. *Let f be a polynomial of degree d. Then $\mathsf{L}(f) \leq \mathsf{S}(f)^{O(\log d)}$.*

12.3 The complexity of basic polynomials

让我们讨论一些多项式的基本例子以及我们对其复杂性的了解。

12.3.1 Symmetric polynomials

一个重要的类别 多项式是 *elementary symmetric* 多项式，d 由...定义

$$\mathrm{SYM}_n^k(x_1, x_2, \ldots, x_n) = \sum_{S \subset [n]:|S|=k} \prod_{i \in S} x_i,$$

对于所有 $0 \leq k \leq n$。计算它们最好的方法是什么？将它们写成“乘积之和”将占用指数级大小（例如，对于 $k = n/2$）。结果发现，允许“和的乘积之和”，这是 Ben-Or [BO85] 关于多项式插值能力的美丽观察，即使在公式中也能实现指数级节省。

Theorem 12.6 [BO85]. *For all n and k, $\mathsf{L}(SYM_n^k) \leq O(n^2)$.*

简单证明源于注意到在 n 变量中的 (多变量) 对称多项式是 (一元) 多项式 $g(t) = \prod_{i=1}^n (t + x_i)$ 在新变量 t 中的 *coefficients*。因此，该公式在 $n+$ 任意 1 个不同的 t (上评估 g，每个 t (是和的乘积)，然后使用插值（将多项式的值转换为系数的线性组合）从这些评估中计算所需的系数。请注意，这仅在足够大的域上有效；实际上，我们知道在固定大小域上；例如，深度-3（或实际上任何有界深度）的中对称多项式电路需要指数大小。

此类电路（或公式），在求和与乘积之间有三次交替，被称为 $\Sigma\Pi\Sigma$ 电路。碰巧的是，对于这种深度为 3 的公式，这个二次上界在所有域上都是紧的 [SW01]。

假设我们移除深度限制，允许使用通用公式或电路。我们是否可以期望在线性大小内计算对称多项式？这种可能性在Strassen和Baur-Strassen的两篇论文的精彩组合中被排除[Str73a, BS83]，再次使用了我们下面概述的基本代数几何。它提供了我们所知的最佳显式电路下界，并且40年来没有被击败！

Theorem 12.7 [Str73a, BS83]. *For all n, $\mathsf{S}(SYM_n^{n/2}) = \Omega(n \log n)$.*

这个证明的想法对于一组更简单的对称多项式来说更容易解释，即*traces* (或*power sums*)，所以我们转向讨论它们。让

$$T_n^d(x_1, x_2, \ldots, x_n) = \sum_{i=1}^n x_i^d.$$

同一位作者证明了以下内容。

Theorem 12.8 [Str73a, BS83]. *For all n, d, $\mathsf{S}(T_n^{d+1}) = \Omega(n \log d)$.*

[4]The usual representation of a polynomial as a sum of monomials is a $\Sigma\Pi$-circuit, and we will discuss the importance of $\Sigma\Pi\Sigma\Pi$-circuits in Section 12.5.3 below.

当我们详细解释这个证明的想法时，一些读者可能希望在第一次阅读时跳过，并继续到下一个主题——矩阵乘法（第12.3.2节）。

这个定理的证明可以通过结合下面的定理12.9和12.10立即得出，这两个定理突出了算术电路以比预期更有效的方式计算的不同非平凡方法，揭示了可能解释为什么下界难以证明的强大能力。我们需要以明显的方式扩展我们的电路大小度量符号，以适应具有多个输出的电路，这些电路计算多个多项式，用$\mathsf{S}(f_1, f_2, \ldots, f_m)$表示。定理12.9表明，计算幂之和的任务并不比单独计算每个幂的任务容易多少，而定理12.10证明了后者的下界。

Theorem 12.9. *For all n, d, $\mathsf{S}(x_1^d, x_2^d, \ldots, x_n^d) \leq O(\mathsf{S}(T^{d+1}))$.*

Theorem 12.10. *For all n, d, $\mathsf{S}(x_1^d, x_2^d, \ldots, x_n^d) = \Omega(n \log d)$.*

每个这些定理都是更一般定理的一个特例，我们依次解释它们。对于第一个，Baur 和 Strassen [BS83] 提供了一般化简（在 [Mor85] 中由 Moregenstern 大大简化），从计算多项式 f 的 *gradient*[5] ∇f 到计算多项式本身。这是发明一个重要算法来证明下界的一个非常好的例子。不用说，计算多变量函数的梯度是优化中的基本子例程，当执行 *gradient descent* 算法的许多变体时。事实上，统计学习社区在 Werbos [Wer74, Wer94] 之前发现了一个更一般的定理。该算法，在该文献中被称为 *back propagation*，是 *Deep Learning* 近期革命的核心，我们在第 20.6.3 节中简要讨论。

Theorem 12.11 [BS83, Wer74]. *For every polynomial f on any number n of variables, $\mathsf{S}(\nabla f) \leq O(\mathsf{S}(f))$.*

这是一个相当令人惊讶的定理，因为计算一阶偏导数最明显的方法是逐个计算，这会导致大小损失一个因子 n。但结果证明，这些计算可以巧妙地组合，只损失一个常数因子的大小。在概述证明之前，请注意，它通过注意到$\frac{\partial}{\partial x_i} T^{d+1} = T^d$蕴含了定理12.9。

为了对定理12.11$_n$（遵循Morgenstern的美丽论证[Mor85]）的证明提供一个高层次提示，想象电路f的门（按顺序）计算多项式$g_1, g_2, \ldots, g_s$，其中前n g_i是变量x_i，最后一个$g_s = f$。将这个电路的“镜像”附加到电路中，即门$h_s, \ldots, h_2, h_1$将（使用多元函数偏导数的链式法则）按此顺序计算$h_i = \partial f / \partial g_i$，使用原始电路中$g_i$的孩子。

现在让我们转向定理12.10。这似乎很明显。毕竟，正如我们在第12.2节中看到的，计算每个输出x_i^d需要log$d$次乘法，而且这些*surely*计算不能合并，因为它们涉及不同的变量。因此，我们必须支付单次任务成本的n倍，从而得到所需的$n\log d$下界。信服了吗？像许多包含“当然”（或“很容易看出”，等等）等词语的论点一样，上述论点是错误的，错误恰恰在于这个傲慢的词语。我们将很快看到一个例子，其中n对完全不相交变量*can*的不同计算可以非平凡地合并，其子线性大小仅增加一个比单个任务多n^c的因子，且$c <$为1。换句话说，在算术计算中，有时可以实现令人惊讶的规模经济，这排除了这种类型的论点。[6]

因此，必须采取另一条途径来证明对于当前的问题，不可能实现规模经济。Strassen [Str73b] 证明了以下一般 *degree* 下界。

Theorem 12.12 [Str73b]. *For any set of polynomials $f_1, f_2, \ldots, f_m$ on a set variables $x_1, x_2, \ldots, x_n$, we have*

$$\mathsf{S}(f_1, f_2, \ldots, f_m) \geq \log \deg(f_1, f_2, \ldots, f_m).$$

这里，deg$(f_1, f_2, \ldots, f_m)$ 扩展了单个多项式度数的通常概念。它表示由多项式 $f_1 - z_1, f_2 - z_2, \ldots, f_m - z_m)$（带有 z_i 新变量）定义的 *algebraic variety* 的次数。

[5] Namely, the vector of first partial derivatives of f: $\nabla f = (\frac{\partial}{\partial x_1} f, \frac{\partial}{\partial x_2} f, \ldots, \frac{\partial}{\partial x_n} f)$.

[6] This question—of achieving economy of scale by combining computations of many independent instances of the same problem—is relevant to any computational model. It is called the *direct sum* question, and is understood only for precious few models.

与 x_j 不相交。我们在此处不正式定义它。但在将此定理与平凡的单变量情况类比时，它在证明中所起的作用很容易理解，这已经给我们提供了 $\mathsf{S}(g) \geq \log \deg(g)$ 的下界。这个单变量下界是从关于次数的基本事实中得出的；即，（a）加法不会增加它所加的两个多项式的最大次数，以及（b）乘法最多将那个最大次数翻倍。碰巧的是，（a）和（b）在具有那种次数概念的多变量情况下也得到满足；这是由一个称为“Bézout定理”的基本代数几何事实保证的。利用这一点，Strassen的多变量定理就像单变量定理一样得出。要从中得出定理12.10成立，只需检查 $\deg(f_1, f_2, \ldots, f_m) \geq d^n$ 是否确实成立，这最终是从以下简单事实中得出的：多项式方程组 $\{f_i = 1\}$ 在 $\mathbb{C}$ 中有 d^n 个解。

12.3.2 Matrix multiplication

考虑下一个 *matrix multiplication* MM，其中 $\mathrm{MM}_n(X, Y)$ 接受两个 $n \times n$ 矩阵 X, Y 并输出它们的乘积 XY。形式上，这是一个多项式 *map*，因为在这里我们计算乘积的 n^2 个条目，并考虑具有如此多输出的电路。显然的算法，分别执行每个内积，给出 $\mathsf{S}(\mathrm{MM}) = O(n^3)$，这被认为是几个世纪以来最好的可能。Strassen [Str69] 通过证明一个次立方界限，$\mathsf{S}(\mathrm{MM}) = O(n^{\log_2 7}) = O(n^{2.8074\cdots})$，震惊了数学界。事后看来，你可以自己想出来：尝试设计一种方法，使用仅7次乘法（和任意数量的加法）来乘以两个 2×2 矩阵。如果你成功了，而且你没有使用矩阵条目的交换律，那么你可以使用递归来乘以更大的矩阵（并检查乘法是否主导总复杂性）。

一个次立方算法的后果是解决前一小节提到的直接和（规模经济）问题。它展示了算术电路 {v*} 在联合执行独立任务时有时能实现非平凡节省。为了看到这一点，请注意，两个 {v*} 矩阵的乘积 {v*} 可以看作是 {v*} 个矩阵-向量乘法的实例，其中我们固定矩阵 {v*}，并让 {v*} 的列向量作为独立变量。对于典型的固定矩阵 {v*}，乘以一个向量的任务需要 {v*} 次操作。斯特拉斯的快速矩阵乘法算法神奇地组合了 {v*} 个独立任务的计算，其规模因子远小于 {v*}。

这个算法突破产生了一系列非常长的指数改进，当前记录是 $\mathsf{S}(\mathrm{MM}) = O(n^{2.3728639\cdots})$。这一历史中的丰富思想在Stothers的博士论文[Sto10]中得到了概述。显然的问题如下：指数会下降多少？它能下降到2吗?

Open Problem 12.13. 证明或反驳：对于每个 $\epsilon > 0$，$\mathsf{S}(\mathrm{MM}) = O(n^{2+\epsilon})$。

主要工作线，负责大多数进展并导致当前记录，包括Strassen的*laser method*的变体和扩展。但正如你可能从记录指数中数字的数量中猜到的，最近取得的进展（使用了大量的计算机计算）回报递减。Ambainis等人[AFG14]正式封装了围绕激光方法的这一系列技术，并证明它们在$n^{2.3078}$处陷入困境。最近在[AW18, CVZ18]中提出了更一般技术的障碍。

一个完全不同、巧妙的矩阵乘法方法由Cohn和Umans [CU03]（并在[CKSU05, CU13, BCC+17]中进一步发展）提出。它展示了如何从（适当的）有限群简单性质直接得出矩阵乘法指数的上界（可能接近2），从而向群论者提出了一个具体的挑战：这样的适当群是否存在？在此，我不会陈述对群所需的条件，但仅指出它涉及某些子群的尺寸及其复不可约表示的最大维度。正如原始论文中精彩解释的那样，这种矩阵乘法方法可以看作是算术计算另一颗宝石的非交换类比：通过循环群Z_n上的FFT对两个n-向量卷积的$n \log n$大小电路。在这两种情况下，所需的产品被简化为在适当群的群代数中乘以两个元素。在阿贝尔情况（卷积）中，傅里叶变换立即将其简化为几个常数的乘法，因为所有不可约表示的维度都是1。在非阿贝尔情况（矩阵乘法）中，傅里叶变换将其简化为一系列较小的矩阵

乘法；大小取决于不可约表示的维度，这些维度递归处理。

12.3.3 The determinant

接下来，我们考虑可能是数学中最重要的多项式，即*determinant*多项式DET，它由以下熟悉的公式定义（其中S_n表示在$[n]$上的排列对称群）：

$$DET_n(X) = \sum_{\sigma \in S_n} sign(\sigma) \prod_{i=1}^{n} x_{i,\sigma(i)}$$

其中$sign(\sigma) = (-1)^{\#\text{of inversions in } \sigma}$。

我们将在下一节中看到，这个多项式可以在多项式时间内计算。这是一个非常令人惊讶的事实，因为它的定义公式包含$n!$个单项式。

我们在这里介绍它的原因是，它的计算算法与我们在本章中看到的其他算法有很大不同。特别是，它允许在我们的算术电路中使用减法门，而且减法在帮助我们计算多项式方面起着至关重要的作用。

Theorem 12.14. *$DET \in VP$.*

我们将在下一节中证明这个定理。这里的问题是：如何？以下是一些可能的方法：

- 使用高斯消元法（Gaussian elimination）。这是一个众所周知的多项式时间算法，但它使用除法，而且我们不知道如何在不使用除法的情况下模拟它。
- 使用Berkowitz算法[Ber84]。这是一个非常巧妙的算法，但它的电路深度为$\Omega(\log^2 n)$。
- 使用Valiant的降维定理[Val79]。这将DET表示为$PERM$的投影，但$PERM$的维度为$2^{\Omega(n)}$。
- 使用Kaltofen和Villard的算法[KV04]。这是一个非常复杂的算法，其电路深度为$O(\log^2 n)$，但其大小为$n^{O(1)}$。

Theorem 12.15.

- $VP_{DET} = VP$.
- $VP_{DET} = VNC^2$.

12.3.4 The permanent

所有至今讨论的多项式都可以高效计算，我们以臭名昭著的*permanent*多项式$PERM$结束我们的讨论，它是行列式的单调兄弟，定义为

$$PERM_n(X) = \sum_{\sigma \in S_n} \prod_{i=1}^{n} x_{i,\sigma(i)}$$

其中S_n表示在$[n]$上的排列对称群。

尽管它与行列式相似，但目前已知的最佳算法是指数级的[Ryser63]，这比我们已知的最佳算法要差得多。

Theorem 12.16 [Ryser63]. *$PERM \in VE$.*

我们很快就会看到，$PERM$是一个非常重要的多项式，即所谓的*VP完全*多项式。

Conjecture 12.17. *$PERM \notin VP$.*

12.4 Reductions and completeness, $\mathcal{VP}$ and $\mathcal{VNP}$

Valiant的论文[Val79a]将算术复杂性转化为复杂性理论。在其中，他提供了布尔计算复杂性的所有基本基础的类似物：

- 提供了一个关于多项式之间有效可约性的数学优雅概念：*projection*。
- 定义了$\mathcal{P}$和$\mathcal{NP}$的算术类似物，现在分别称为“$\mathcal{VP}$”和“$\mathcal{VNP}$”。
- 赋予这些类在如下缩减下的自然完备多项式：*permanent*对$\mathcal{VNP}$是完备的，*determinant*对$\mathcal{VP}$（几乎是）完备的。

让我们逐一考虑这些，然后讨论一些后果。请注意，所有定义都是*nonuniform*关于布尔电路。

我们现在定义两种几乎等价的概念化简，Valiant的*projection*和稍微更一般的、数学上标准的*affine projection*。[9]虽然我们主要使用后者，但本质上所有结果都适用于两者。

Definition 12.18 (投影和仿射投影)**.** 设 $f \in \mathbb{F}[x_1, x_2, \ldots, x_n]$ 和 $g \in \mathbb{F}[y_1, y_2, \ldots, y_m]$。我们称 f 是 g 的 *affine projection*，记作 $f \leq g$，如果存在 m 个仿射函数 $\ell_i : \mathbb{F}^n \to \mathbb{F}$ 使得 $f(x) = g(\ell_1(x), \ell_2(x), \ldots, \ell_m(x))$。我们称 f 是 g 的 *projection*，如果它是所有仿射函数 ℓ_i 至多依赖于一个变量的仿射投影。

显然，这些减少在电路大小方面是高效的；如果 $f \leq g$，那么 $\mathsf{S}(f) \leq \mathsf{S}(g) + O(mn)$，因为给定 g 的电路，我们可以为其输入提供仿射函数 ℓ_i 以获得 f 的电路。这种关系显然是传递的，因此给出了多项式相对复杂性的偏序，就像在布尔世界中一样。

$\mathcal{VP}$ 类与 $\mathcal{P}$/poly 完全类比，简单地就是所有可以通过多项式大小算术电路计算的多项式。记住，对于所讨论的所有多项式，其次数都是多项式地受变量数量的限制。因此，例如，上一节中的多项式 SYM, MM, DET 都在 $\mathcal{VP}$ 中。

Definition 12.19 (类 $\mathcal{VP}$)**.** 我们说 $f = \{f_n\}$ 在 $\mathcal{VP}$ 中，如果 $\mathsf{S}(f) \leq n^{O(1)}$。

定义 $\mathcal{VNP}$ 的模拟 $\mathcal{NP}$ 稍微复杂，但仍然很自然。在 $\mathcal{NP}$ 中使用了存在量词，这可以看作是所有可能的布尔值到潜在证人或“证书”的布尔 *disjunction*，在一个多项式大小的布尔电路中。在算术 $\mathcal{VNP}$ 中，这个析取被替换为在一个多项式大小的算术电路中可能证人的 *summation*（从而有效地将计算某些对象的存在转换为计数它们）。仍然只对 *Boolean* 值取这个和，而不考虑底层域，这是一个非平凡且重要的选择。

Definition 12.20 (类 $\mathcal{VNP}$)**.** 我们说 $f = \{f_n\} \in \mathbb{F}[x_1, x_2, \ldots, x_n]$ 在 $\mathcal{VNP}$ 中，如果存在 $g = \{g_n\} \in \mathbb{F}[x_1, x_2, \ldots, x_n, y_1, y_2, \ldots, y_n] \in \mathcal{VP}$ 使得 $f_n(x)$ 从 $g_n(x, y)$ 通过 $f_n(x) = \sum_{\alpha \in \{0,1\}^n} g_n(x, \alpha)$ 定义。

我们显然有 $\mathcal{VP} \subseteq \mathcal{VNP}$，算术复杂度理论的主要问题是证明以下猜想。

Conjecture 12.21. $\mathcal{VP} \neq \mathcal{VNP}$.

最后，我们到达了完整的难题，行列式和恒等式。注意到当域 $\mathbb{F}$ 的特征为 2 时，这两个多项式是相同的，我们现在只考虑特征不同于 2 的域。Valiant 证明了两个完备性定理。其中一个关于 $\mathcal{VNP}$ 是完全干净的。

[9]Note that most polynomial-time reductions discussed for Boolean computation are actually projections, or a very simple Boolean function of a few projections.

Theorem 12.22 [Val79a]. *PER is $\mathcal{VNP}$-hard.*
More precisely, for every $f = \{f_n\} \in \mathcal{VNP}$ and every n, $f_n \leq PER_{n^{O(1)}}$.

读者被邀请验证，同样 $PER \in \mathcal{VNP}$，因此定理意味着 PER 实际上是对 $\mathcal{VNP}$ 的一个完整多项式。行列式的完备性表述起来有点困难。首先，我们需要定义一个重要的子类 $\mathcal{VP}$，称为 $\mathcal{VL}$，即所有具有多项式大小公式的多项式。

Theorem 12.23 [Val79a]. *DET is $\mathcal{VL}$-hard. More precisely, for every $f = \{f_n\} \in \mathcal{VL}$ and every n, $f_n \leq DET_{n^{O(1)}}$.*

由于 DET 在 $\mathcal{VP}$ 中，对 $\mathcal{VL}$ 来说很困难，并且根据定理12.5，这两个类几乎相等（直到拟多项式因子），我们得到了 DET-完备性在 $\mathcal{VP}$ 中的精确含义。也就是说，每个多项式 $f \in \mathcal{VP}$ 都是 $n^{O(\log n)}$ 行列式的投影。

这些完备性结果，除了为许多其他缩减提供起点（正如 SAT 在布尔复杂性中所扮演的角色），还突出了这两个多项式——恒等式和行列式的重要性，并可能部分解释了它们在数学中扮演的重要角色。首先，行列式在数学中无处不在——许多自然出现的有用多项式都可以表示为行列式（例如，*Jacobians* 在微积分中，*Alexander polynomials* 在结理论中，*Wronskians* 在微分方程中，*characteristic polynomials* 和 *resultants* 在代数中，*volumes* 在几何中的平行六面体，以及其他许多）。现在我们知道，每个可以用小公式描述的多项式都有一个同样小的行列式表示，这种现象可能就不会那么令人惊讶了。同样，恒等式也相当频繁地出现。它似乎捕捉了图论中的 *Tutte* 和 *chromatic polynomials*，结理论中的 *Jones polynomials*，以及在统计力学模型中出现的许多 *partition functions*；它还计算凸集中的整数点，计算偏序集的扩展等等。与行列式的例子不同，这些似乎很难计算。

这些完备性结果的重要贡献之一是，将 $\mathcal{VP}$ 与 $\mathcal{VNP}$ 分离的主要问题可以转化为从 PER 到 DET 的最佳投影问题。更精确地说，设 $m(n)$ 为满足 $PER_n \leq DET_{m(n)}$ 的最小整数。那么完备性结果意味着以下内容。

Corollary 12.24. *If $\mathcal{VP} \neq \mathcal{VNP}$ then $m(n) \neq n^{O(1)}$. And almost conversely, if $m(n) \neq n^{O(\log n)}$, then $\mathcal{VP} \neq \mathcal{VNP}$.*

尝试研究 $m(n)$ 的工作始于Polya，他注意到[10] $m(2) = 2$。甚至可能不清楚 $m(n)$ 是否总是有限的，但当然结合定理12.16和12.23，我们得到 $m(n) \leq \exp(n)$。由Mignon和Ressayre [MR04]（由[CCL10]扩展到所有领域）给出的最佳下界是二次的。

Theorem 12.25. $m(n) \geq \Omega(n^2)$.

证明本质上是线性的，并且它只使用了线性投影和行列式多项式的Hessian的简单性质。改进这个二次界限是一个主要挑战。

Separating $\mathcal{VP}$ from $\mathcal{VNP}$

一个旨在证明 $m(n)$ 上的超多项式甚至指数下界，并将 $\mathcal{VP}$ 与 $\mathcal{VNP}$ 区分开来的雄心勃勃的计划由 Mulmuley 和 Sohoni 提出（参见综述[BLMW11, Mul11, Mul12a, Gro15]，以及我们在第 13.9.1 节中的讨论）。这一计划关键地利用了永久多项式和行列式多项式都由它们的对称性决定的这一事实。这些对称性是线性群的子群，仿射投影降维也是线性的。这允许将 $\mathcal{VP}$ 与 $\mathcal{VNP}$ 的问题形式化为关于由行列式和永久多项式的轨道闭包（在其自然对称群下）定义的代数簇交集的问题。这种表述自然地建议使用不变量理论、表示理论和代数几何中的工具（更多细节见第 13.9 节）。到目前为止，这个计划似乎存在严重障碍（例如，参见[BIP16]），但这一重点

[10] Via the simple projection $\mathrm{PER}\left(\begin{smallmatrix} x & y \\ z & w \end{smallmatrix}\right) = \mathrm{DET}\left(\begin{smallmatrix} x & -y \\ z & w \end{smallmatrix}\right)$.

使用对称性和开发出的工具可能有助于阐明（也许还能证明算术复杂性中其他问题的新的下界）。

另一种完全不同且反直觉的分离 $\mathcal{VP}$ 与 $\mathcal{VNP}$ 的途径是通过设计一个针对布尔复杂性问题的高效确定性算法，关于算术复杂性。这个方向是这两个领域之间的重要桥梁，并将它们与伪随机性和去随机化联系起来。当前的问题是 *polynomial identity testing* 问题（或简称为 PIT）。它询问一个给定的算术公式是否计算了恒等于零的多项式。几乎等价地（由于定理 12.23），在 Edmonds [Edm67b] 提出的原始形式中，同样的问题询问给定一个 *symbolic matrix* (的行列式，其条目是某些变量) 的线性形式，是否恒等于零。这个问题有一个简单的有效概率算法（在第 7.1 节开头讨论），但已知的最优确定性算法需要指数时间。令人惊讶的是，Kabanets 和 Impagliazzo [KI04] 证明了确定性复杂性的显著改进（即非平凡的去随机化）将涉及算术或布尔复杂性的显式下界。关于 PIT 问题及其相关问题的更多内容，以及在该问题及其相关问题上取得的局部进展，可以在综述 [SY10] 的第 4 节中找到。更近期的进展，来自一个非常不同的方向，可以在 [GGOW15] 中找到。

最后，正如在布尔复杂性（见第5章）中，人们可以也应该对*why*感到好奇，我们在算术设置中也陷入了困境，几十年来无法证明强下界并将$\mathcal{VP}$与$\mathcal{VNP}$区分开来。事实上，似乎算术设置比布尔设置更有限（或更结构化），我们也有可能从数学中获得更多工具来应用于下界。为了解释这一点，正如在布尔设置中，我们需要*barrier*结果，说明为什么当前技术到目前为止取得了有限的成果。这些技术最近才得到发展。条件障碍，在本质上类似于自然证明（见第5.2.4节），在[FSV17, GKSS17]中得到了发展。对于捕获算术复杂性中大多数现有下界的线性代数方法，一个无条件的障碍在[EGOW17]中得到了发展。

12.5 Restricted models

关于布尔电路，我们无法证明通用模型强下界的能力促使我们研究受限模型，这些模型本身通常很有趣。我们描述了其中一些已知超多项式下界的模型，尽管它们的性质受限，但仍然存在一些重大挑战。在之前的章节中描述得较为简略的类似努力，正在布尔电路、证明复杂性和各种其他计算环境中进行。总的来说，我们（希望）通过开发新技术，正在逐步接近真正通用的下界，并通过在多种受限计算模型上证明下界来检验我们的能力。

12.5.1 Monotone circuits

单调电路对于有序域如 $\mathbb{R}$ 或 $\mathbb{Q}$ 有意义。它们就像一般电路一样定义，只是它们只能使用域中的 *positive* 系数。单调电路可以明显地计算任何具有正系数的多项式。就像在布尔情况下一样，我们既有指数下界，也有单调与非单调电路之间的自然分离。实际上，这些算术界限早于布尔界限，并且显著更简单。

Theorem 12.26 [SS79, TT94]. *PER requires monotone circuits of size* $\exp(n)$.

Theorem 12.27 [Val80]. *There is a positive polynomial $f \in \mathcal{VP}$ that requires* $\exp(n)$ *monotone circuits.*

12.5.2 Multilinear circuits

在多线性电路和公式中，每个门必须计算一个多线性函数。显然，这样的电路可以计算每个多线性多项式。

Theorem 12.28 [Raz04a]. *DET requires multilinear formulas of size $n^{\Omega(\log n)}$.*

我们知道 $\mathsf{L}(DET) \leq n^{O(\log n)}$，但提供这个上界的公式是 *not* 多线性。实际上，人们认为 DET 实际上需要指数大小多线性电路。证明这一点，以及证明

任何超多项式 多线性电路的l电路大小下界是重要的 重要开放问题。

Tensors 自然子族的多线性函数由 *tensors* 定义。张量以自然的方式扩展矩阵——一个 d-张量由一个与相关 d-形式（与矩阵定义双线性形式的方式相同）的系数的 d-维数组描述。理解张量对于数学、物理和计算机科学的许多领域至关重要。Lansberg 的书 [Lan12] 对张量的几何、复杂性和应用进行了广泛的论述。

张量最基本的复杂度度量是其*rank*；再次，从矩阵扩展这个概念，对于给定的d-张量T，其秩rk(T)是相加得到T的最小*rank-1*张量数量。由于秩为1的张量仅仅是d线性形式的乘积，因此由秩定义的计算模型极其简单：一个算术ΣΠΣ-电路（乘积之和的和），在下一小节中将有更多讨论。那么，我们能否证明张量秩的下界呢？这不可能容易。遗憾的是，与矩阵的情况（d=2）不同，矩阵的秩完全被理解并且可以高效计算，对于d >2，结构要复杂得多，计算秩成为$\mathcal{NP}$-完全[Hås90]。

让我们将已知的最简单非平凡情况联系起来，$d = 3$. 确定三阶张量的秩在许多应用中是必不可少的；例如，上面讨论的矩阵乘法指数是由自然的3维矩阵乘法张量确定的。计数论证表明（在任意域上）大多数$n \times n \times n$张量的秩为$\Omega(n^2)$。但任何显式3阶张量最佳下界[AFT11]仅为3n。困难部分可由最近的障碍结果[EGOW17]解释。它们表明，迄今为止用于此类下界的大多数方法（包括代数几何学家使用的*flattenings*）都无法得到优于6n的秩下界。这提出了以下自然挑战，即获得显式的超线性下界。

Open Problem 12.29. 描述一个显式的3-张量族T_n（大小为n），其rk(T_n) $\neq O(n)$。

12.5.3 Bounded-depth circuits

这个模型听起来可疑地受限。让我们明确其重要性，以及通过研究它最近在算术复杂性方面取得的令人兴奋的进展。

操作操作交替次数的界限是布尔复杂性和逻辑中的标准做法（例如，一阶理论允许存在量和全称量词之间有限次的交替）。我们已经在第12.2节中看到了对布尔电路加法和乘法之间交替次数的类似限制。例如，ΣΠ电路捕捉了多项式标准写法，即单式之和。在第12.6定理中，我们看到允许更多一次交替，ΣΠΣ电路（和的积的和），可以给出指数级的优势（例如，在计算对称多项式时）。显然，允许更多一次交替将更强大，以此类推。然而，长期以来，人们认为对这种受限电路的研究主要是为了缓慢地发展“真正的东西”，即通用电路下界。这种观点一夜之间随着Agrawal和Vinay [AV08]的一篇论文而改变。他们意识到，深度降低（第12.5定理）的想法可以被用来将电路压缩到只有四次交替，即到ΣΠΣΠ电路。他们的结果被Koiran和Tavenas [Koi12, Tav13]进一步锐化，产生了以下结果。

Theorem 12.30. *If $f \in \mathcal{VP}$, then f has a ΣΠΣΠ-circuit of size $n^{O(\sqrt{n})}$. Moreover, if f is homogeneous, the resulting circuit is homogeneous.*[11]

因此，为了证明一般下界，“我们”所需要的只是同质 ΣΠΣΠ-电路的下界。这为攻击此类电路提供了巨大的能量提升，Gupta、Kamath、Kayal 和 Saptharishi [GKKS13] 的突破性成果几乎证明了此类下界，这一系列改进最终导致了 Saraf 和 Kumar [SK14] 的匹配下界。

Theorem 12.31. *There exists an explicit homogeneous polynomial $f \in \mathcal{VP}$, which requires ΣΠΣΠ-circuits of size $n^{\Omega(\sqrt{n})}$.*

[11] Namely, every gate in the circuit computes a homogeneous polynomial.

注意，根据前面的定理，{v*}对指数的这种非常数改进（例如，对于永续量）将分离$\mathcal{VP}$和$\mathcal{VNP}$！本定理的下界是通过使用由[GKKS13]发起的*projected, shifted partial derivatives*实现的。我将不解释这种技术，但将简要解释其祖先，即*partial derivatives*技术。它是由Nisan和Wigderson[NW96]引入的，他们用它来证明较弱ΣΠΣ电路和其他限制模型的下界。这些以及更多关于偏导数方法的应用在[SY10, CKW11]中得到了综述。主要思想是在多项式上定义以下*complexity measure*。对于一个多元多项式g，考虑集合$PD(g)$是所有g (的各阶偏导数的多项式，并让dim(g)表示$PD(g)$的线性空间的维度。这个度量对于输入变量来说很小，并且发现它在加法和乘法下“缓慢增长”。因此，具有大dim(f)的多项式需要大的电路！

12.5.4 Non-commutative circuits

非交换性不仅存在于数学中，也存在于生活中。事实上，我们遇到或考虑的大多数动作对都不交换。当变量取非交换环中的值时，如矩阵环或非交换群的群代数，自然会出现非交换多项式。

到目前为止，在本节中，我们隐含地假设了交换律（即，我们所有的变量 {v*} 两两交换）。现在让我们放弃这个假设，并讨论非交换多项式、以及它们的电路和公式。对于非交换多项式，必须指定每个单项式中变量出现的顺序（例如，xy 和 yx 是不同的单项式）。同样，在电路和公式中，我们必须指定乘积门中乘法的顺序。这个模型弱点的良好证明是，在交换律设置中，我们可以仅使用 *one* 乘法来计算 x^2-y^2，即 $(x-y)(x+y)$，但在非交换律设置中这是不可能的。

Nisan [Nis91a] 证明了行列式[12] (和永续项)的指数公式下界，以及公式和电路的指数差距。

Theorem 12.32 [Nis91a]. *PER and DET require non-commutative formulas of size* exp(n).

Theorem 12.33 [Nis91a]. *There is a non-commutative polynomial with a linear size non-commutative circuit, which requires* exp(n) *non-commutative formulas.*

注意，最后一个定理意味着定理12.5的深度降低，表明在交换情况下公式和电路具有几乎相等的功率，在非交换情况下是错误的。事实上，这两个世界之间还有许多其他差异。其中一个令人惊讶的是，由于Arvind和Srinivasan[AS10]，在非交换情况下，永权和行列式（凯莱版本）的难度相等：$DET \leq PER$和$PER \leq DET$！我们讨论的另一个问题是，Strassen在计算多项式时高效消除除法门，在非交换情况下并不为人所知。这在[HW14]中进行了研究，并且与第13.9.3节中讨论的某些不变量理论问题非常密切相关。

该领域的核心问题是证明超多项式电路下界。一种攻击[HWY10]展示了如何从某些超线性 *commutative* 电路下界推导出这样的（甚至指数）下界。

[12] As discussed, to formally define this polynomial, one has to order the variables in each monomial. A natural ordering we pick is row-order, called the *Cayley determinant*.

13 *Interlude*: Concrete interactions between math and computational complexity

介绍讨论了数学与计算之间各种交互作用的高级概述。在本章中，我们将遇到计算复杂性理论与不同数学领域的具体交互实例。我的目标是多样性——本节展示了现代数学的几乎任何领域都未受到这种计算联系的触及，这在某些情况下相当令人惊讶。

我已经选择在每个数学领域集中关注一个基本问题或发展。通常，这仅触及一个小子领域，无法公正地体现丰富的联系。因此，每个部分都应被视为更大工作量甚至更大潜力的展示。确实，在某些领域，合作已经相当稳固，而在其他领域，它们才刚刚萌芽，有许多激动人心的问题等待解决，有许多理论有待发展。描述相对简短，但它们包括背景和直觉，以及进一步阅读材料。本章中的小节有望诱使读者深入探索。

字段和焦点的选择受我个人的品味和有限的知识所影响。组合数学、优化、逻辑、拓扑和信息理论等领域与其他领域的联系已在本文本的部分中出现。毫无疑问，还可以添加其他领域。以下是涵盖的领域和每个领域选择的主题列表；这些部分可以按任何顺序阅读：

- 数论：*Primality testing*
- 组合几何：*Point-line incidence*
- 算子理论：*The Kadison-Singer problem*
- 度量几何：*Distortion of embeddings*
- 群论：*Generation and random generation*
- 统计物理：*Monte-Carlo Markov chains*
- 分析和概率：*Noise stability*
- 格论：*Short vectors*
- 不变量理论：*Actions on matrix tuples*

13.1 Number theory

提到，高效计算数学对象的需求一直是数学家和科学家们历史上的核心问题，当然，最早的主题是算术。这种关注的最激进体现可能是我们用来表示整数的位值系统，它之所以被使用了数千年，正是因为它支持对算术运算进行极其高效的操纵。自从古代以来，算术中的下一个计算挑战就是访问以这种方式表示的整数的乘法结构。

这里摘录了C.F.高斯对他的时代数学界的一次呼吁[1]（在1801年的第329篇文章
Disquisitiones Arithmeticae（中），关于测试素数和将整数分解为其素数因子的计算复杂性。高斯对这一计算挑战的重要性，他对当时技术状态的挫败感，以及他恳请数学界解决这一问题的呼吁，都显露无遗：

[1]Which is of course in Latin. I copied this English translation from a wonderful survey by Granville [Gra05] on the subject matter of this section.

> 区分质数和合数的问题，以及将后者分解为其质因数的问题，在算术中被认为是最重要的和最有用的之一。这个问题吸引了古代和现代几何学家们的行业和智慧，以至于详细讨论这个问题似乎是多余的。尽管如此，我们必须承认，迄今为止所提出的所有方法要么局限于非常特殊的情况，要么是如此繁琐和困难，以至于即使是对于不超过可敬之人构建的表格极限的数字，它们也考验着熟练计算者的耐心。而且，这些方法根本不适用于更大的数字……科学本身的尊严似乎要求探索一切可能的手段来解决这样一个优雅且备受赞誉的问题。

我简要回顾了这两个基本数论问题的最新进展。2002年，Agrawal、Kayal和Saxena[AKS04]找到了高斯第一个问题*efficiently deciding primality*的一个显著回应。为此问题使用符号多项式是完全新颖的。以下是他们对素数的优雅描述。

Theorem 13.1 [AKS04]. *An integer $N \geq$^2 is prime if and only if*

- *N is not a perfect power,*
- *N does not have any prime factor $\leq$ (对数 $N)^4$, and*
- *For every $r, a <$ (对数 $N)^4$, we have the following equivalence of polynomials over $\mathbb{Z}_N[X]$:*

$$(X + a)^N \equiv X^N + a \mod (X^r - 1).$$

可以看出，这种特征化产生了一种简单的测试素性的算法，该算法是确定性的，并且运行时间与N的二进制描述长度是多项式级别的。之前的确定性算法要么假设了广义黎曼猜想[Mil76]，要么需要略高于多项式级别的时间[APR83]。AKS确定性算法是在一系列有效的概率算法[SS77, Rab80, GK86, AH92]之后出现的，其中一些是基本的，而另一些则需要复杂的数论技术和使用的发展。这些概率性和确定性算法部分受到密码学领域的启发，并且对该领域具有重要意义。

不为人所熟知的是，即使是那些阅读了[AKS04]中美丽、巧妙的证明的人，也并不知道AKS通过仔细去随机化[AB03]（使用多项式）的先前概率算法来开发他们的确定性算法。请注意，*de-randomization*将概率算法转换为确定性算法，现在已成为计算复杂性中的一个主要领域，拥有丰富的理论和许多类似的成功以及挑战。*every*高效概率算法存在确定性对应物的惊人可能性是计算复杂性中的一个主要问题，并且有强有力的证据支持它（参见[IW97]）。关于这一点，可以在处理随机性的章节中找到更多内容，特别是第7章。

高斯第二个挑战，即是否能够高效分解整数，仍然悬而未决。但正是这个挑战在多个重要方面丰富了计算机科学，包括实践和理论。事实上，分解的假设难度几乎成为全球几乎所有加密和电子商务系统安全性的主要保障（表明困难问题可能是有用的！）更普遍地说，密码学是数论概念的狂热消费者，包括椭圆曲线、Weil配对等，这些对于各种密码学原语和应用至关重要。这些发展打破了哈代对数论完全无用的学术追求的看法。

存在一些关于整数的难题，其自然定义依赖于分解，但仍然可以有效地解决，从而绕过分解的看似必要性。或许最早被正式描述的算法是欧几里得算法，用于计算两个给定整数的最大公约数[2] m和n。另一个著名的此类算法是利用高斯二次互反律计算勒让德-雅可比符号（$\frac{m}{n}$）。

[2]It extends to polynomials and allows an efficient way of computing multiplicative inverses in quotient rings of $\mathbb{Z}$ and $\mathbb{F}[x]$.

一个快速算法可能在量子计算的新发展中出现，量子计算是基于量子力学原理的计算机研究，我们在第11章中讨论了这一点。Shor在[Sho94]中表明，这样的计算机能够在多项式时间内分解整数。这一结果导致政府、公司和学术界投资数十亿美元开发能够构建大规模量子计算机的技术，而该项目是否可行仍存在争议。目前没有已知的理论障碍，但该项目至今未能成功的一个可能原因是量子力学中尚未发现的原理。

其他中心计算问题包括在有限域中求解多项式方程，其中最早的高效（概率性）算法之一是由Berlekamp [Ber67] 开发的（使此算法去随机化仍然是一个巨大的挑战）。在算法数论书籍[BS97]中可以找到许多其他例子。

13.2 Combinatorial geometry

什么是包含单位长度线段在 *every* 方向上的平面区域的最小面积？这是Kakeya针问题（这样的集合被称为 *Kakeya sets*），Besicavitch [Bes19] 意外地解决了这个问题，他表明这个面积可以任意接近于零！对他方法的一点点变化产生了一个勒贝格测度为零的Kakeya集。用更稳健的度量，如豪斯多夫和闵可夫斯基维度来代替“面积”（即，勒贝格测度）是有意义的。这改变了局面：Davies [Dav71] 证明了平面的Kakeya集在这两个度量中都必须具有全维度 (=2)，尽管在勒贝格测度中如此稀疏。

自然地，将这个问题扩展到更高维度是合理的。然而，获得类似的结果（即，Hausdorff和Minkowski维度是满的）却变得极其困难。尽管这个问题似乎具有娱乐性质，但它对许多数学领域（傅里叶分析、波动方程、解析数论和随机提取）具有重要意义，并且已经通过相当多样的数学思想进行了攻击（参见[Tao09]）。

以下欧几里得问题的有限域类似由沃尔夫[Wol99]提出。令{v*}表示一个大小为{v*}的有限域。如果一个集合{v*}包含每个方向上的直线，则称为Kakeya集合。更确切地说，对于每个方向{v*}，都存在一个点{v*}，使得直线{v*}：{v*}包含在{v*}中。正如上面所述，我们希望证明任何这样的{v*}都必须很大（将维度{v*}视为一个很大的常数，将域大小{v*}视为趋于无穷大）。

Conjecture 13.2. 设 $K \subseteq \mathbb{F}^n$ 为一个Kakeya集。那么 $|K| \geq C_n q^n$，其中 C_n 是仅依赖于维度 n 的常数。

最佳指数 q 在这样的下界中直观上对应于欧几里得空间中的Hausdorff和Minkowski维度。利用算术组合学的先进技术，Bourgain、Tao等人将 $n/2$ 的平凡界改进到大约 $4n/7$。

奇怪的是，这个猜想完全独立地出现在ToC中，源于对*randomness extractors*的研究工作[LRVW03]，这是一个研究弱随机源纯化的领域，我们在第9章中讨论过（例如，参见综述[Vad11]）。在这种动机下，Dvir [Dvi09] 优雅地证明了Wolff猜想（有时称为“有限域Kakeya猜想”），使用了（代数几何）多项式方法（该方法受到解码代数纠错码技术的启发）。随后，这种技术很快被应用于其他几何问题，包括Guth-Katz [GK10] 解决著名的Erdős距离问题，以及最优随机提取等（一些在Dvir的综述[Dvi10]中列出）。

后续工作确定了上述常数 C_n 的确切值（至多乘以2）[DKSS13]，这是基于 Dvir 的突破性成果 [Dvi09]。

Theorem 13.3 [DKSS13]. *Let $K \subseteq \mathbb{F}^n$ be a Kakeya set. Then $|K| \geq (q/2)^n$. However, there exist Kakeya sets of size $\leq 2 \cdot (q/2)^n$.*

许多其他关于点和线（以及更高维度的几何对象）的关联问题一直是几何学家、代数学家和组合学家之间活动和合作的热点。

计算机科学家。这些问题在计算机科学方面的动机来自各种来源，例如局部纠错问题[BDWY13]和去随机化[DS07, KS09]。其他发生定理（例如，Szemerédi-Trotter [STJ83]及其Bourgain-Katz-Tao [BKT04]的有限域版本）已被用于随机性提取[BIW06]、压缩感知[GLR10]和扩张构造（见第8.7节）。

13.3 Operator theory

以下1959年Kadison和Singer的基本数学问题[KS59]旨在形式化Dirac关于量子力学中测量的“普遍性”的基本问题。我们需要几个定义。考虑$B(\mathcal{H})$，希尔伯特空间$\mathcal{H}$上的连续线性算子代数$\mathcal{H}$。定义一个$state$为$B(\mathcal{H})$上的线性泛函f，归一化到$f(I) = 1$，它在正半定算子上取非负值。状态形成一个凸集，如果一个状态不是其他状态的凸组合，则称为$pure$。最后，让D是$B(\mathcal{H})$的子代数，由所有$diagonal$算子（在固定某些基之后）组成。

Kadison和Singer询问D上的每个纯态是否都有$unique$扩展到$B(\mathcal{H})$。这个问题在无限维算子中在有限维度找到了许多等价的表达形式，这些表达形式来源于算子理论、差异理论、Banach空间理论、信号处理和概率的动机和直觉。所有这些问题都在Marcus、Spielman和Srivastava最近的工作中得到了肯定的解决[MSS13b]（该工作还概述了许多相关的猜想）。以下是他们证明的一个陈述，它暗示了其他陈述。

Theorem 13.4 [MSS13b]. *For every $\epsilon > 0$, there is an integer $k = k(\epsilon)$ such that the following holds. Fix any n and any $n \times n$ matrix A with zeros on the diagonal and of spectral norm 1. Then there is a partition of $\{1, 2, \cdots, n\}$ into k subsets, $S_1, S_2, \cdots, S_k$, so that each of the principal minors A_i (namely, A restricted to rows and columns in S_i) has spectral norm at most ϵ.*

此陈述明确暗示，其中一个次级矩阵至少具有线性尺寸 n/k。这个结果被称为Bourgain和Tzafriri的 *restricted invertibility* 定理 [BT91]，它本身是算子理论中的一个重要结果。

计算机科学家是如何对这个问题产生兴趣的？不深入太多细节，这里简要描述了通往这一辉煌成果的曲折历程。

一个核心的计算问题，众多应用的核心，是求解线性方程组。虽然高斯消元法相当高效（算术运算的数量约为n^3对于$n \times n$矩阵），但对于大型n来说，这仍然效率低下，需要寻求更快的方法，希望接近线性地依赖于给定矩阵的非零项数量。对于拉普拉斯[3]线性系统（出现在许多图论应用中，如计算电流量和随机游走），Spielman和Teng[ST11]实现了这一点。他们引入的一个主要概念是矩阵（或加权图）的 *spectral sparsifiers*。

一个给定矩阵的稀疏化矩阵是另一个矩阵，其非零项（实际上，是线性的）要少得多，但仍然具有与原始矩阵基本相同（归一化）的谱（甚至不明显存在这样的稀疏矩阵）。请注意，完全图的稀疏化器的一个非常特殊的情况是按定义 *expander graphs*[4]（，关于这个扩展子中心概念的更多内容请参阅[HLW06]和第8.7节）。算法应用导致了对任意拉普拉斯矩阵的稀疏化器的最优构造的追求（在稀疏性和逼近之间的权衡），这在[BSS14]中得到了美丽的实现，他们还提供了一种确定性的多项式时间算法来构造这样的稀疏化器。这反过来又导致[SS12]对上述限制可逆性定理的新证明，并对其与Kadison-Singer问题的联系进行了分析。

然而，解决 Kadison-Singer 的问题似乎需要另辟蹊径。同一团队 [MSS13a]

[3] Simply, symmetric PSD matrices with zero row sum.

[4] All nontrivial eigenvalues of the complete graph (or constant matrix) are 0, and an expander is a sparse graph in which all nontrivial eigenvalues are tiny.

首先解决了Bilu和Linial [BL06] 关于矩阵“签名”谱的 bold conjecture。[5] 这个猜想是构建拉马努金图（最佳可能的扩张图）的简单迭代计划的一部分。拉马努金图在 [LPS88, Mar88] 中被引入和构建，但依赖于数论和代数几何的深刻结果（一些人认为这对于任何此类构建都是必不可少的）。Bilu和Linial寻求一种基本的构建方法，并在他们的猜想上取得了进展，展示了他们的迭代方法如何为构建“接近”拉马努金扩张器提供另一种方式。

为了证明Bilu-Linial猜想（并且实际上产生每个可能度数的拉马努金图——这是代数构造无法提供的），[MSS13a]发展了一种关于*interlacing poly- nomials*的理论，这最终成为解决[MSS13b]中Kadison-Singer问题的关键技术工具。在这两种情况下，新颖的观点是将这些猜想概率化，并通过分析平均特征多项式来分析随机算子的范数。这种方法是有意义且实际上有效的，这是深刻而神秘的。此外，它提供了一种新的存在证明，对于寻找所需对象没有已知的有效算法（甚至概率算法）。分析大量使用了*real stable*多项式理论，其背后的归纳过程让人联想到（并受到）Gurvits的[Gur08]关于van der Waerden猜想及其推广的非凡证明。[7]

13.4 Metric geometry

一个度量空间与另一个度量空间有多接近，由*distortion*这一概念来捕捉，它衡量一个空间的距离嵌入到另一个空间时发生了多少扭曲。

Definition 13.5. 设 (X, d) 和 (X', d') 是两个度量空间。如果对于每对点 $x, y \in X$，都有$f : X \to X'$ 的保形度为 $\leq c$，则称 f 为嵌入

$$d(x, y) \leq d'(f(x), f(y)) \leq c \cdot d(x, y).$$

当 X 是有限且大小为 n 时，我们允许 $c = c(n)$ 依赖于 n。

理解各种度量空间和赋范空间之间最佳嵌入的方法一直是泛函分析理论和度量几何学中的一个长期努力。该领域的一个主要结果是Bourgain嵌入定理[Bou85]。

Theorem 13.6 [Bou85]. *Every metric space of size n can be embedded into Euclidean space L_2 with distortion $O(\log n)$.*

第一个将这些结构问题与计算复杂性联系起来的重要论文是Linial、London和Rabinovich的论文[LLR95]。他们寻求寻找低失真嵌入的有效算法，并注意到对于某些这样的问题，使用半定规划是自然的。他们将这种几何联系应用于获取图上的算法问题（特别是我们很快将要讨论的稀疏切问题）的旧结果和新结果。他们讨论的另一个动机（这很快发展成为近似算法的主要主题）是，某些计算（例如寻找最近邻）在某些空间中比在其他空间中更有效，因此有效的低失真嵌入可能从困难空间到简单空间提供了有用的降低。他们描述了一种实现Bourgain定理13.6的有效算法，并证明他的界限是最好的（证明这个界限的度量仅仅是任何常数度*expander*图中点之间的距离；参见第8.7节）。

下一个该领域的演变阶段，以及几何学家和ToC研究人员之间互动水平的转变，源于试图证明“近似难度”结果。一个例子是Goemans-Linial猜想 [Goe97, Lin02]，研究稀疏割问题，关于L_1与“负类型”度量空间 L_2^2 (的关系，这是一种在多个情境下自然出现的度量空间的一般类别)。

[5] This beautiful conjecture states that for every d-regular graph, there exist $\{-1, 1\}$ signs of the edges, which make all eigenvalues of the resulting signed adjacency matrix lie in the “Ramanujan interval” $[-2\sqrt{d-1}, 2\sqrt{d-1}]$.

[6] With respect to the spectral gap. This is one of a few important expansion parameters to optimize.

[7] This is yet another example of a structural result (on doubly stochastic matrices) whose proof was partly motivated by algorithmic ideas. The connection is the use of hyperbolic polynomials in optimization (more specifically, as barrier functions in interior point methods.)

大致上，这些是在 $\mathbb{R}^n$ 上的度量，其中欧几里得距离被平方。更精确地说，一个度量 (X, d) 是负类型的（即在 L_2^2 中），如果 $(X, \sqrt{d})$ 与 L_2 的一个子集同构（没有扭曲）。

Conjecture 13.7. $L22$ 可以嵌入到 L_1 中，距离保持不变 rtion.

这个猜想被Khot和Vishnoi [KV05]证明是错误的，他们证明了以下内容。

Theorem 13.8 [KV05]. *For every n there are n-point subsets of L_2^2 for which every embedding to L_1 requires distortion $\Omega(\log \log n)^{1/6}$.*

比结果本身更有趣的是它的起源。Khot 和 Vishnoi 正在试图证明（加权）“最稀疏割”问题是难以近似的。他们通过一个被称为 Khot 的 *unique games* 假设的计算假设 [Kho02]（参见[Kho10]和第 4.3 节），通过所谓的 *PCP*-约简（参见第 10.3 节）做到了这一点。消除这个计算假设是展示计算问题之间约简的强大功能和灵活性的神奇部分。他们将他们的 PCP 约简应用于一个特定、精心选择的唯一游戏实例，该实例不能被某个半定程序很好地近似。结果是稀疏割问题的一个实例，同样的约简确保它难以被半定程序近似。如上所述，该结果实例可以理解为度量空间，近似困难转化为所需的失真界限。

精确地确定了将嵌入 L_2^2 到 L_1 中的精确扭曲为 $\sqrt{\log n}$ (，直到低阶因子)，在两篇发展新算法和几何工具的美丽论文中得到了精确确定。我只提到每个的最后一个词，因为这些论文包含详细的历史。在上界方面，Arora、Lee 和 Naor [ALN08] 通过所谓的测量下降法获得了逼近非均匀稀疏切到因子 $\sqrt{\log n} \log \log n$ 的有效算法，这给出了相同的扭曲界限。最近，Naor 和 Young [NY17] 使用海森堡群上的新等周不等式证明了扭曲的下界为 $\sqrt{\log n}$。

另一个此类问题与目录表（ToC）之间强大的联系是通过（再次）展开图。一个基本例子是任何常度展开图的图度量证明了上述Bourgain嵌入定理是最佳的。更复杂的例子来自于试图理解（也许反驳）Novikov和Baum-Connes猜想（参见[KY06]）。此项目依赖于另一个，远比*coarse*嵌入弱的概念。

Definition 13.9. (X, d) 具有 (X', d') 的粗糙嵌入，如果存在映射 $f : X \to X'$ 和两个递增、无界的实值函数 α, β，使得对于任意两点 $x, y \in X$，

$$\alpha(d(x, y)) \leq d'(f(x), f(y)) \leq \beta(d(x, y)).$$

Gromov [Gro03] 首先证明了无限多个扩张器不能粗略地嵌入到希尔伯特空间中。这个结果被 Lafforgue [Laf08] 和 Mendel-Naor [MN14] 大大推广，他们构造了不能粗略地嵌入到任何 *uniformly convex* 空间的图度量。有趣的是，虽然 Lafforgue 的方法是代数的，但 Mendel-Naor 的构造遵循了来自计算复杂性的扩张器的组合 *zig-zag* 构造 [RVW02]。

许多其他关于度量嵌入和扭曲的交互项目，我们没有涉及，包括它们在诸如聚类、距离预言机和k-服务器问题等众多算法和数据结构问题中的应用，以及扭曲与*dimension reduction*之间的基本相互作用，这对几何和计算机科学都相关，其中许多基本问题仍然是开放的。

13.5 Group theory

群论学家，就像数论学家一样，自从该领域起源以来就内在地对计算问题感兴趣。例如，给定某个群的生成元中的一个词*word problem* (，它是否评估为平凡元素？)对于理解任何研究的群都是如此基本，以至于一旦创造了语言来正式讨论这个问题的计算复杂性，就有一系列结果随之而来，试图确定这种复杂性。这包括可判定性和不可判定性

结果一旦图灵建立了目录并提供了第一个不可判定问题；随后是 $\mathcal{NP}$ 完备性结果和一旦在 1970 年左右引入 $\mathcal{P}$ 和 $\mathcal{NP}$ 后的效率算法。不用说，这些 *algorithmic results* 通知我们手头群的 *structural* 复杂性。而词问题只是第一个例子。另一个演示是几十年来算法和结构进步之间的美妙互动，在 *graph isomorphism problem* 上，最近导致了 Babai [Bab15] 的突破！[8] 大量工作致力于寻找计算交换子群、Sylow 子群、中心化子、基、表示、特征以及一群其他重要子结构的有效算法，这些子结构可以从对群的某种自然描述中找到。优秀的教科书包括 [HEO05, Ser03]。

这里我们关注两个相关的问题，即 *generation* 和 *random generation* 问题，以及从计算复杂性中借用的新概念，这些概念对于研究它们至关重要。在正式定义它们之前，让我们考虑一个例子。假设我给你10个可逆矩阵，比如说100 $\times$ 100大小的矩阵，在大小为3的域上。你能告诉我它们是否生成另一个这样的给定矩阵吗？在我们都死去之前，你能提供令人信服的证据吗？关于生成由这些生成器生成的子群中的随机矩阵怎么样？当然，问题是这个子群的大小将远远大于已知宇宙中的原子数量，因此其元素无法列出，并且生成群中元素的典型单词可能需要过长。事实上，即使在极其特殊的情况下，对于 $\mathbb{Z}_p^*$ (中的元素，即一个，1×1矩阵)，第一个问题与 *discrete logarithm* 问题相关，对于 $\mathbb{Z}_{p \cdot q}^*$，它与 *integer factoring* 问题相关，目前都需要指数时间来解决（作为描述长度的函数）。

让我们考虑任何有限群 G，并让 $n \approx \log |G|$ 大约是 G 中一个元素的描述的长度。假设我们给出了 k 个元素在 G，$S = \{s_1, s_2, \ldots, s_k\}$ 中。如果我们描述的算法能在 n 和 k (的时间复杂度内工作将是理想的，这阻止了枚举 G 的元素，其大小是指数级的 n)。

*generation problem*询问一个给定的元素$g \in G$是否由S生成。如何证明这样一个事实？对于肯定答案的标准证书是*word*在S (的元素及其逆元)中求值得到g。然而，即使G是循环的，这样的最短词可能呈指数级增长于n。一种替代的、受计算动机驱动的描述是给出一个*program*来描述g。它的定义表明，“程序”这个词非常适合它，因为它具有与典型计算机程序相同的结构，只是我们使用的是乘法和逆元这样的群运算，而不是应用一些标准的布尔或算术运算。

Definition 13.10. 一个 *program* (超越 S) 是一个由元素 $g_1, g_2, \ldots, g_m$ 组成的有限序列，其中每个元素 g_i 要么在 S 中，要么是之前 g_j 的逆，要么是之前 g_j, g_ℓ 的乘积。如果 $g = g_m$，我们称它“简单计算 g”。

在循环情况下，程序在描述长度上提供了指数级的节省，因为一个程序允许我们通过重复平方元素来写出大幂。值得注意的是，这种节省对于*every*群是可能的。Babai和Szemerédi [BS84]的这项发现表明，每个群的每个元素都可以用生成它的任何一组元素以极其简洁的方式描述。

Theorem 13.11 [BS84]. *For every group G, if a subset of elements S generates another element g, then there is a program of length at most $n^2 \approx (\log |G|)^2$ that computes g from S.*

它很有趣地注意到，这个证明使用了一个对群论家来说非常组合和反直觉的结构：即一个 *cube*，我们稍后会再次看到。对于从 G 中选取的元素序列 $(h_1, h_2, \ldots, h_t)$，立方 $C(h_1, h_2, \ldots, h_t)$ 是由 2^t 个元素 $\{h_1^{\epsilon_1}, h_2^{\epsilon_2}, \ldots, h_t^{\epsilon_t}\}$ 组成的（多重）集合，其中 $\epsilon_i \in \{0, 1\}$。证明的另一个重要特征是它在非常一般的“黑盒”群设置中工作——它永远不需要显式描述宿主群，只需要能够乘以元素并取它们的逆。这对于论证群是一个非常重要的范例，将在下面再次使用。

如何证明一个元素 g 是由 S 生成的 *not*? 可能根本不存在简短的“经典”证明！这个问题激励了Babai定义Arthur-Merlin游戏——一种新的概率性交互证明的概念（同时与Goldwasser、Micali和Rackoff [GMR89]提出类似）

[8] See also Helfgott's exposition of this result in [HBD17].

密码学原因的构想）。Babai展示了如何在新的框架中证明非成员资格。交互式证明定义对计算理论的影响巨大，并在第10.1节中讨论。

返回到生成问题，现在让我们考虑更具挑战性的问题 *random generation*。这里我们给定 S，并希望有一个随机化过程能够快速输出在由 S 生成的 G 的子群 H 上的（几乎）均匀分布。这个问题，除了其自然吸引力外，通常被计算群论学家遇到，是许多群论算法的子例程。在实践中，使用启发式方法，如著名的“乘积替换算法”及其变体，这些方法通常效果良好（例如，参见最近的 [BLG12] 和参考文献）。在这里，我们讨论可证明的界限。

显然，S及其逆元中的足够长的随机词将完成这项工作，但就像证书一样，足够长往往是过于长的。在一份漂亮的论文中，Babai [Bab91] 描述了一种生成随机程序的过程，该程序计算H的几乎均匀元素，并在 $n^5 \approx (\log |G|)^5$步内运行。此算法在黑盒群的完全一般性下工作。该论文随后由Cooperman以及Dixon [Coo02, Dix08] 提出了更快且分析更简单的算法，而最先进的算法其步数与上述提到的生成证明的长度惊人地相同——换句话说，随机性实现了对此问题非确定性的效率。

Theorem 13.12 [Bab91, Dix08, Coo02]. *For every group G, there is a* 概率 *program of length* 多项式(n) ≈ 多项式(log $|G|$) *that, given* 任何 *generating set S for G, produces with high probability a (nearly) uniformly random element of G.*

13.6 Statistical physics

统计物理领域非常广泛，我们在此主要关注统计力学与ToC的联系。存在大量各种物理和化学系统的数学模型，旨在理解不同材料的基本性质和基本过程动力学。这些包括诸如Ising、Potts、单体-二聚体、自旋玻璃和渗流等熟悉的模型。一个典型的例子是解释此类数学模型与物理学和化学的联系以及研究的基本问题，即Heilmann和Lieb [HL72] 的开创性论文。

许多研究的问题都可以在以下一般设置中观察。我们有一个巨大的（指数）对象空间称为Ω（这些对象可以被视为系统的不同配置）。每个对象分配一个非负权重（这可以被视为该状态的“能量”）。缩放这些权重会产生Ω上的概率分布（通常称为“吉布斯分布”），为了研究其性质（相变、临界温度、自由能等），人们试图从这个分布中生成样本（如果状态描述需要n位，那么列出所有相关概率将是指数级不切实际的）。

由于Ω可能高度无结构，解决此采样问题的最常见方法被称为蒙特卡洛马尔可夫链（或MCMC）方法。其思路是在Ω的对象上构建一个图，如果一对对象在某些意义上相似（例如，仅在少数坐标上不同的序列），则通过边连接。接下来，从任何对象开始，在这个图上执行一段时间有偏随机游走：达到的对象就是产生的样本。在许多情况下，设置随机游走（通常称为“格劳伯动力学”或“梅特罗波利斯算法”）并不困难，以便马尔可夫链的*limiting*分布确实是所需的分布。这种方法中的主要问题是*when*停止游走并输出一个样本；*when*我们是否足够接近极限？换句话说，链需要多长时间才能收敛到极限[11]？在许多情况下，这些决策是基于直觉和启发式方法做出的，而没有对收敛时间进行严格分析。例外情况，其中已知严格的界限，通常是结构化的，例如，当马尔可夫链是某些群（例如，[Ald83, Dia88]）的凯莱图时。

[9] The reader familiar with Section 8.7 on expanders will realize that, if the Cayley graph of G with generators S is a sufficiently good expander, then short random words suffice.

[10] In a sense, the random straight-line program “adds generators” which shortcut long words and creates expansion.

[11] Yet again notice that this is essentially an issue of graph expansion (see Section 8.7) which we just encountered in the previous Section 13.5. Work described in this section has considerably enriched our understanding of expanders.

自过去几十年与计算理论互动以来，这种状况已经发生了相当大的变化。在描述它之前，让我们看看计算问题在这个领域中甚至是如何出现的。两个主要来源是 *optimization* 和 *counting*。上述设置适合许多优化问题实例是很容易看到的。将Ω视为给定优化问题的解集（例如，满足一组约束的一组参数值）和权重表示解的质量（例如，满足的约束数量）。因此，从相关分布中随机选择有利于高质量解。计数联系更为微妙。在这里，Ω代表一个想要计数或近似其大小的组合对象集（例如，图中完美匹配的集合，或满足一组约束的赋值）。[JVV86] 的一个基本结果表明，对于许多这样的设置，随机采样一个对象（近似地）允许递归过程产生集合大小的近似值。此外，将有限集视为连续对象（例如，凸集中的格点）的精细离散化，允许计算体积，更普遍地，在这样一些域上积分函数。

大约在1990年，引入了严格的技巧[Bro89, SJ89, Ald90, DFK91]来分析由不同的近似算法产生的这种一般马尔可夫链的收敛速度。他们通过*canonical paths*或*coupling*论证在马尔可夫链上建立了*conductance*界限（这项早期工作的综述见[Jer96]）。协作工作很快就能正式证明许多模型中建议的一些启发式算法背后的物理直觉，并且还吸引了物理学家为优化问题提出这样的巧妙链。该领域也吸引了概率论家和几何学家，现在非常活跃且多样化。我们提到两个结果来说明这类重要问题的严格收敛界限。

Theorem 13.13 [JSV04]. *The permanent of any nonnegative $n \times n$ matrix can be approximated, to any multiplicative factor $(1+\epsilon)$, in polynomial time in n/ϵ.*

这个近似算法的重要性源于Valiant [Val79b] 的开创性结果，即对于几乎所有计数问题（特别是上述统计物理模型、优化和计数问题中出现的那些），永久[12]多项式是*universal*。因此，与行列式不同，精确计算它极其困难。

Theorem 13.14 [DFK91]. *The volume of any convex set in n dimensions can be approximated, to any multiplicative factor $(1+\epsilon)$, in polynomial time in n/ϵ.*

体积，除了其内在兴趣外，还捕捉了自然计数问题，例如给定偏序集的扩展数。该算法的分析以及其许多后续改进，导致了凸几何中具有独立兴趣的纯结构结果，以及诸如从任何对数凹分布中高效采样的推广（参见综述[Vem05]）。

另一个合作的后果是对*spacial*属性（如相变和吉布斯分布中遥远站点之间的长程相关性）与*temporal*属性（如采样或近似计数算法（如Glauber动力学）的收敛速度）之间关系的更深入理解。这种联系（例如在[DSVW04]中调查）自20世纪70年代以来已被物理学家用于自旋系统建立。Weitz [Wei06] 在*hard core*模型上的突破性工作给出了一种在相变前有效的*deterministic*算法，并由Sly [Sly10] 在相变后的硬度结果补充。这些计算复杂性的相变，与吉布斯分布的相变同时发生，是引人注目的，而这种现象的普遍性仍在研究中。

更普遍地，统计物理模型与优化问题（尤其是在随机实例中）的紧密相似性，对双方都有益。让我提几个令人兴奋的发展。这种联系揭示了相变时解空间精细的几何结构，例如，对于k-SAT [ACORT11]。基于重整化、退火和副本对称破缺等想法的物理直觉导致了优化问题的新的算法，其中一些现在得到了严格的解析（例如[JS93]）。其他，如布尔可满足性问题（在一般情况下是$\mathcal{NP}$-完备的）中最快（但未证明）的启发式方法之一，基于物理方法“调查传播”[MPZ02, Het16]。另一个联系是*Lovász local lemma* (LLL)，它使人们能够建立罕见“全局”事件的*existence*。LLL的有效算法版本由Beck [Bec91]启动，从Moser [Mos09]（然后[MT10]）的工作开始，导致了罕见事件的近似计数和均匀采样版本（例如，参见[GJL16]）。这些分析*directed, non-reversible*马尔可夫链的新技术是许多更多应用的有力新工具。Moitra [Moi16]在LLL区域中的完全不同的*deterministic*算法承诺了更多的应用；它甚至在解空间（以及因此自然马尔可夫链）不连通的情况下也能工作。回到MCMCs，我们注意到[ALGV18]最近在近似随机簇模型的配分函数和基的数量方面的突破性工作；在另一个令人惊叹的互联演示中，它特别使用了关于*high-dimensional*扩张器上的随机游走的结果（在我们扩张器部分的末尾提到，第8.7节）。

最后，我们注意到MCMC算法是最古老的概率算法之一，其诞生和使用都早于计算复杂性理论。然而，随着对随机性和其在算法中力量的研究（第7章和第9章对此进行了阐述），我们对这类算法的理解有了显著增长。MCMC算法实际上只需要很少的随机位这一特别有趣且令人惊讶的后果，在[NZ96]中给出。

13.7 Analysis and probability

本节展示了越来越多的不等式家族——大偏差不等式、等周不等式等——由于计算理论和离散数学中的各种动机而超越了其经典起源。此外，应用有时需要这些不等式的 *stability* 版本，即理解使不等式几乎尖锐的结构。这些动机还推动了经典结果的推广和许多新的

[12]This notorious sibling of the determinant, in which no signs appear, was defined and discussed in Chapter 12.

ones. 大部分材料在此处，以及更多关于分析布尔函数这一激动人心的领域的动机和发展，可以在O'Donnell的书中找到[O'D14]。

以下故事可以从多个角度来讲述。一个角度是函数的 {v*}。让我们将注意力限制在具有均匀概率测度的布尔立方上，但许多问题和结果也扩展到任意乘积概率空间。设 $f:\{-1,1\}^n \to \mathbb{R}$，我们假设它是平衡的，即 $E[f]=0$。当 f 的像为 $\{-1,1\}$ 时，我们可以将 f 视为一个投票方案，将 n 个个体的二进制投票转换为二进制结果。从这样的投票方案中，一个自然的愿望可能是 *noise stability*——即通常非常相似的输入（投票向量）将产生相同的结果。虽然在这个社会科学环境中很自然，但此类问题也出现在统计物理环境中，其中自然函数，如键渗流，对噪声表现出极高的敏感性 [BKS99]。让我们正式定义噪声稳定性。

Definition 13.15. $\rho \in [0,1]$ 是一个相关参数。我们说两个向量 $x,y \in \{-1,1\}^n$ 是 ρ- 相关的，如果它们分布如下。向量 x 是随机均匀抽取的，y 通过独立翻转 x 的每个比特 x_i 并以概率 $(1-\rho)/2$ 获得从 x。注意，对于每个 i，相关 $E[x_i y_i] = \rho$。*noise sensitivity* 在 ρ，$S_\rho(f)$ 的 f 定义为输出之间的相关，$E[f(x)f(y)]$。

可以看出，使噪声稳定性最大化的函数是任何 *dictatorship* 函数（例如，$f(x)=x_1$），其中 $S_\rho(f)=\rho$。但另一个自然的社会科学关注点是投票方案中玩家的 *influence* [BOL85]，这禁止了这样的解决方案（在民主环境中）。选民的影响是在所有其他选票都是均匀随机的情况下，它可以改变结果的概率（因此，在独裁制度下，独裁者为1，其他人为0）。一个公平的投票方案应该没有选民具有高影响力。正如我们在实值函数中定义影响力一样，我们将使用（条件）*variance* 来衡量一个玩家在所有其他（随机）选票的情况下可能产生的影响。

Definition 13.16. 一个函数 $f:\{-1,1\}^n \to \mathbb{R}$ 对 τ 有影响，如果对于每个 i，$\mathrm{Var}[x_i|x_{-i}] \le \tau$ 对于所有 i（，其中 x_{-i} 表示不含 i 个坐标的向量 x。

例如，多数函数有影响 $O(1/\sqrt{n})$。一个平衡函数的影响可以多小的问题非常有趣，并且引出对我们故事（内容和技巧）高度相关的不等式。事实证明，最终公平（每个玩家的$1/n$ 影响力）是不可能的：[KKL88] 表明每个函数都有一个具有非成比例影响力的玩家，至少 $\Omega(\log n/n)$。无论如何，人们可以问具有 *small* 影响力的函数中哪一个最稳定，并且自然地猜测多数应该是最好的。[13]

这个情况被称为*majority is stablest*猜想，其来源完全不同且令人惊讶——优化领域，特别是“近似难度”。一篇杰出的论文[KKMO07]表明，它意味着[14]一种自然算法在近似图的最大割（将顶点分割以最大化它们之间的边数——一个基本优化问题，其精确复杂度为$\mathcal{NP}$-完全）的优化性。这种联系高度非平凡，但到目前为止，我们已经有很多例子表明，对各种优化问题的某些（基于半定规划的）近似算法的分析提出了许多新的等周问题，丰富了这一领域。

大多数稳定的猜想由Mossel、O'Donnell和Oleszkiewicz在[MOO10]中在提出后不久以强形式证明。这里是一个正式的陈述（实际上适用于有界函数）。

Theorem 13.17 [MOO10]. *For every (positive correlation parameter) $\rho \ge 0$ and $\epsilon > 0$, there exists (an influence bound) $\tau = \tau(\rho,\epsilon)$ such that for every n and every $f:\{-1,1\}^n \to [-1,1]$ of influence at most τ, $S_\rho(f) \le S_\rho(Majority_n) + \epsilon$.*

证明揭示了故事另一个角度——大偏差不等式和不变性原理。为了看到联系，回忆Berry-Esseen定理[Fel71]，它推广了标准中心极限定理

[13] This noise sensitivity tends, as n grows, to $S_\rho(Majority_n) = \frac{2}{\pi}\arcsin\rho$.

[14] Assuming another, complexity-theoretic, conjecture called the "unique games" conjecture, discussed in Section 4.3.

到 *weighted* 个独立随机符号的和。在这个定理中，影响非常自然地出现。考虑 $\sum_{i=1}^n c_i x_i$。如果我们把权重 c_i 归一化以满足 $\sum_i c_i^2 = 1$，那么 c_i 是第 i 个投票者的影响，且 $\tau = \max_i |c_i|$。这个中心极限定理的质量随着影响 τ 线性下降。Lindeberg 对 Berry-Esseen 定理的证明使用了一个不变性原理，表明对于线性函数，累积概率分布 $Pr[\sum_{i=1}^n c_i x_i \leq t]$（对于每个 t）是不变的（直到 τ），*regardless* 变量的分布 x_i，只要它们是独立的，期望为 0 且方差为 1。因此，特别是，它们可以被取为标准高斯分布，这使问题变得简单，因为加权求和也是一个高斯分布！

要证明定理13.17，首先观察到在噪声稳定性问题中，高斯情况是简单的。如果x_i, y_i是具有相关系数ρ的标准高斯，则稳定性问题简化为Borell [Bor85] 的一个经典结果：噪声稳定性通过原点的任何超平面都是最大的。请注意，这里的n维高斯的旋转对称性，它也有助于证明，并不能区分“独裁者”函数和多数函数——两者都是这样的超平面。鉴于这一点，一个质量依赖于τ的不变性原理就能完成任务。下一步是证明只需要证明*low degree*多线性多项式的不变性原理就足够了（因为噪声的影响随着度数的增加而衰减）。最后，需要证明这类多项式的Berry-Esseen的非线性扩展，其形式如下。我注意到，不变性原理被用于证明[MOO10]中的其他猜想，并且自从这篇论文发表以来，已经找到了相当多的进一步推广和应用。

Theorem 13.18 [MOO10]. *Let x_i be any n independent random variables with mean 0, variance 1, and bounded 3rd moments. Let g_i be n independent standard Gaussians. Let Q be any degree-d multilinear n-variate polynomial of influence τ. Then for any t,*

$$|Pr[Q(x) \leq t] - Pr[Q(g) \leq t]| \leq O(d\tau^{1/d}).$$

为了结束本节，让我再给出一个关于噪声稳定性和等周问题的令人惊讶的问题（及其答案）的演示，这些问题源于优化近似难度相同的计算考虑。这里是问题：*What is the smallest surface area of a (volume-1) body that tiles $\mathbb{R}^d$ periodically along the integer lattice $\mathbb{Z}^d$?*换句话说，我们寻求一个 d-维体积-1 子集 $B \subseteq \mathbb{R}^d$，使得 $B + \mathbb{Z}^d = \mathbb{R}^d$ 及其边界具有最小 $(d-1)$-维体积.[15]让我们用 $s(d)$ 表示这个下确界。你可以在这里稍作停顿，测试你的直觉：你期望答案在 d 的渐近情况下是什么?

此类问题源于19世纪末汤姆森（后来的开尔文勋爵）对*foams*在三维空间中的研究[Tho87]，后来在数学、物理学、化学、材料科学甚至建筑学中得到进一步研究、推广和应用。然而，对于这个非常基本的问题，其中周期性由最简单的整数晶格定义，似乎对于大的d，$s(d)$的平凡上下界在超过一个世纪的时间里没有得到改善。$s(d)$的平凡上界由单位立方体提供，其表面积为$2d$。$s(d)$的平凡下界来自忽略镶嵌限制，仅考虑体积。在这种情况下，单位体积球具有最小的表面积，$\sqrt{2\pi e d}$。$s(d)$在这个二次范围内位于何处？特别是，是否存在具有$s(d) = O(\sqrt{d})$的“球体立方体”？

最后一个问题成为复杂性理论家们的一个核心问题，当[FKO07]将其直接与重要的唯一游戏猜想以及上述组合问题的最优不可近似性证明（特别是最大割问题）联系起来时。论文详细阐述并激励了这种非平凡的联系，通过寻找Raz的[Raz98a]著名的并行重复定理的最紧版本。[16]Raz [Raz11]再次提供了一个关于并行重复定理“多强”的限制。将他的技术扩展到几何、连续设置，[KORW08]解决了上述问题，证明了球体立方体确实存在！

Theorem 13.19 [KORW08]. *For all d, $s(d) \leq \sqrt{4\pi d}$.*

[15]Note that the volume of B ensures that the interiors of $B + v$ and $B + u$ are disjoint for any two distinct integer vectors $u, v \in \mathbb{Z}^d$, so this gives a tiling.

[16]A fundamental information-theoretic inequality of central importance to the “amplification” of probabilistically checkable proofs (PCPs).

一个简单的证明以及该结果的多种扩展随后在 [AK09] 中给出。所有已知的证明都是概率性的。给出一个可能更好地说明球形立方体（即使表面面积非常糟糕但非平凡）看起来像的具体构造似乎是一个具有挑战性的问题。

13.8 Lattice theory

欧几里得空间中的格是数学中最普遍的对象之一，因为它们不仅是自然的（例如，在晶体结构中出现）且本身就值得研究，而且它们捕捉了不同领域中的各种问题，如数论、分析、逼近理论、李代数和凸几何。正如我们将看到的，格论中的许多基本结果都是*existential* (，即不提供获得已证明存在对象的有效手段)，这在某些情况下限制了这些应用的发展。

本节讲述了Lenstra、Lenstra和Lovász [LLL82]的一个算法的故事，通常被称为“LLL算法”，以及它对这些经典应用以及密码学、优化、数论、符号代数等领域现代应用的某些影响。但我们最好先定义一个格[17]。

设 $B = \{b_1, b_2, \ldots, b_n\}$ 是 $\mathbb{R}^n$ 的一个基。那么 *lattice* $L(B)$ 表示由这些向量的所有整数线性组合构成的集合（实际上，阿贝尔群）；$L(B) = \{\sum_i z_i b_i : z_i \in \mathbb{Z}\}$。$B$ 也被称为晶格的 *basis*。自然地，给定的晶格可以有许多不同的基（例如，由 $\{(0,1),(1,0)\}$ 生成的平面标准整数晶格，同样也可以由 $\{(999,1),(1000,1)\}$ 生成）。与晶格 L 相关的一个基本不变量是其行列式 $d(L)$，它是任何 B 基的 $\det(B)$ 的绝对值，这也是晶格的基本平行六面体的体积)。为了简单起见，并且不失一般性，我们假设 B 已经归一化，因此我们只考虑 $d(L) = 1$ 的晶格 L。

最基本关于格的结果，即它们必须包含 *short* 个向量（在任何范数下），是由闵可夫斯基（他开创了格论，以及与之相关的数论几何）证明的 [Min10]。

Theorem 13.20 [Min10]. *Consider an arbitrary convex set K in $\mathbb{R}^n$ that is centrally symmetric*[18] *and has volume $> 2^n$. Then, every lattice L (of determinant 1) has a nonzero point in K.*

这个无辜的定理，它有一个简单但*existential* (鸽巢)证明，结果在几何学、代数和数论中有许多基本应用。在著名的例子中，这个定理（通过适当的范数和格的选择）产生了诸如狄利克雷的丢番图逼近定理和拉格朗日的四个平方定理等结果，并且（通过更多的工作）数域类数的有限性（参见，例如[PZ89]）。

从现在起，我们将关注（最自然的）欧几里得范数下的短向量。将Minkowski定理直接应用于立方体 $K = [-1,1]^n$ 的一个直接推论是以下内容。

Corollary 13.21. *Every lattice L of determinant 1 has a nonzero point of Euclidean norm at most $\sqrt{n}$.*

稍微偏离一下主题，请注意，在Minkowski一个世纪之后，Dadush（参见[DR16]）提出的关于 *computational*动机的上述推论[19]的强逆命题最近在[RSD16]中得到证明。这个逆命题对于格的覆盖半径、算术组合数学、布朗运动等有许多结构性的后果。我们在这里不会详细阐述计算复杂性和优化与格理论和凸几何之间这种新的相互作用。上述论文巧妙地说明了这些联系和应用，并讲述了为这个复杂证明所需的思想和技术工作的历史。

返回到闵可夫斯基关于欧几里得范数的推论，证明仍然是存在性的，找到这样一个短向量的明显算法在 n 中需要指数时间。突破性论文 [LLL82] 描述了 LLL 算法，这是一个有效、多项式时间的算法，通过一个 2^n 因子近似任何 n-维格的短向量长度。

[17] Only full-rank lattices are defined here, which suffice for this exposition.
[18] Namely, $x \in K$ implies that also $-x \in K$. Such sets are precisely balls of arbitrary norms.
[19] Which has to be precisely formulated.

Theorem 13.22 [LLL82]. *There is a polynomial-time algorithm, which given any lattice L, produces a vector in L of Euclidean length at most a factor of 2^n longer than the shortest vector in L.*

这个指数界限一开始可能看起来过高，但应用的数量和多样性令人震惊。首先，在许多问题中，维度 n 是一个小的常数（因此实际的输入长度来自给定基的位数）。这导致例如Lenstra算法在常维数下（精确求解）整数规划[Len83]。它还导致Odlyzko和Riele对Mertens关于Möbius函数中消去的猜想[OtR85]的驳斥，以及[Sim10]中的大量数论示例。但结果是，即使当 n 非常大时，许多问题也可以在 poly(n)-时间内解决。以下是一些代表这种多样性的新旧问题示例列表，其中一些可以追溯到原始论文[LLL82]。总的来说，只需要将实数输入在维度 n 中近似到 poly(n) 位即可。

- **Diophantine approximation**. 当使用有界分母的有理数来逼近一个实数时，其最佳可能逼近可以通过其（可高效计算的）连分数展开轻松解决，但对于 *simultaneous* 逼近，没有已知的方法。形式上，给定一个实数集（例如，$\{r_1, r_2, \ldots, r_n\}$），一个界限 Q 和 $\epsilon > 0$，找到整数 $q \leq Q$ 和 $p_1, \ldots, p_n$ 使得所有 $|r_i - p_i/q| \leq \epsilon$。存在性地（使用Minkowski），Dirichlet的“盒子原理”表明 $\epsilon < Q^{1/n}$ 是可能的。使用LLL，可以高效地获得 $\epsilon < 2^{n^2} Q^{1/n}$,，这对于由 poly($n$) 比特描述的 Q 是有意义的。

- **Minimal polynomials of algebraic numbers**. 在这里，我们给定一个单实数 r 和一个度数界限 n，并询问是否存在一个系数为整数的多项式 $g(x)$，其度数不超过 n，且 r 是其根（如果存在，还需要生成这样的多项式 g）。实际上，这是上述问题的一个特殊情况，其中包含 $r_i = r^i$。虽然该算法仅输出 g，对于 $g(r) \approx 0$，但通常很容易检查它实际上为零。注意，通过改变 n，我们可以找到这样的最小多项式。

- **Polynomial factorization over rationals**. 在这里，输入是一个整数多项式 h，其次数为 n，我们希望将其在 $\mathbb{Q}$ 上进行因式分解。高级思想是首先找到 h (的一个（近似）根 r，例如使用牛顿法)，将其输入到上述问题中，这将返回一个以 r 为根的最小 g，从而可以整除 h。我们强调，此算法产生的是精确的因式分解，而不是近似的!

- **Small integer relations between reals**. 给定实数 $r_1, r_2, \ldots, r_n$，以及一个界限 Q，确定是否存在整数 $|z_i| < Q$ 使得 $\sum_i z_i r_i = 0$（如果存在，则找出这些整数）。作为一个著名的例子，LLL 可以找到 arctan(1) ≈ 0.785398, arctan($1/5$) ≈ 0.197395 和 arctan($1/239$) ≈ 0.004184 之间的整数关系，从而得到 Machin 的公式

$$\arctan(1) - 4\arctan(1/5) + \arctan(1/239) = 0.$$

- **Cryptanalysis**. 注意上述问题的一个非常特殊的情况（其中系数 z_i 必须是布尔值）是“背包问题”，一个著名的 $\mathcal{NP}$-完全问题。这里的关键是，在密码学的早期，一些系统基于对背包问题的假设“平均情况”难度。许多这样的系统都通过使用 LLL（例如，[Lag84]）被破解。LLL 也被用来破解 RSA 密码系统的一些版本（具有“小的公钥指数”）。

它可能是对最后一项的合适结语，因为格不仅能够破坏密码系统，还能够创建它们（我们在第18章讨论密码学）。高效地近似短向量直到多项式（与LLL产生的指数相反）因子的问题被认为在计算上是困难的。以下是这个假设的一些主要后果。首先，Ajtai在一篇令人瞩目的论文[Ajt96]中表明，这种困难在平均意义上被保留，在巧妙选择的随机格分布上。这导致了Ajtai和Dwork[AJ97]基于这种困难的新公钥加密方案。基于这种困难假设的变体，目前是唯一已知的可以

可能承受 *quantum attacks*；我们在第11章讨论量子计算，特别是演示了Shor的高效量子算法如何高效地分解整数和计算离散对数[Sho94]。量子计算技术的最新进展推动了大量投资，以使这种基于格的密码系统实用化！在Gentry [Gen09b] 的另一项突破性工作中，这种困难假设被用来设计 *fully homomorphic* 加密，这是一种不仅允许加密数据，而且可以直接对加密数据进行任意计算的计划。更多内容请参阅Peikert [Pei16, Bra 18] 的优秀综述。

13.9 Invariant theory

不变量理论，诞生于1845年凯莱的论文[Cay45]，是代数学的一个重要分支，不仅与代数几何和表示论有重要的自然联系，而且与其他许多数学领域也有联系。我们在这里将看到一些，以及与计算复杂度的一些新联系，这将导致该领域的新问题和结果。请注意，计算效率在不变量理论中始终很重要，该理论充满了巧妙的算法（从凯莱的*Omega process*开始），这一点从书籍[CLO92, Stu08, DK15]中可以看出。

不变量足够熟悉，例如以下示例：

- 在高中物理中，我们学习到能量和动量在一般物理系统的动力学中是守恒的（即，是 *invariants*）。
- 在化学反应中，每种元素的原子数保持不变，因为一种分子混合物转化为另一种分子（例如，将氢氧化钠（NaOH）和盐酸（HCl）结合产生常见的食盐氯化钠（NaCl）和水（H_2O））。
- 在几何学中，一个经典的难题是：一个平面多边形能否沿着直线“剪切和粘贴”到另一个多边形上？这里明显的不变量，*area*，是唯一的。[20]然而，将这个难题推广到三维多面体时，发现除了明显的变量，*volume*，还有一个由德恩发现的变量。[21]

更普遍地，关于两个曲面（例如，结）在同胚下的等价性、两个群是否同构，或者两个点是否在动力系统的同一轨道上等问题，都会引发类似的问题和处理方式。对这类问题给出否定答案的一种典型方法是通过 *invariants*，即某些作用在底层空间上保持的量。

我们将关注在 *linear groups* 上作用在 *vector spaces* 上的不变量。让我们回顾一些符号。固定一个域 $\mathbb{F}$ (，虽然问题在每一个域中都有趣，但结果主要只适用于无限域，有时只是对于特征零或代数闭的域)。设 G 为一个群，V 为 G 的一个表示，即一个 $\mathbb{F}$-向量空间，其中 G 作用：对于每一个 $g, h \in G$ 和 $v \in V$，我们有 $gv \in V$ 和 $g(hv) = (gh)v$。

向量（或点）$v \in V$ 在 G 下的 *orbit*，记为 Gv，是所有可以通过此动作移动到 v 的点的集合，即 $Gv = \{gv : g \in G\}$。理解一组动作的轨道是这个领域的核心任务。一个基本问题捕捉了许多上述例子，即：给定两个点 $u, v \in V$，它们是否位于同一个 G 轨道中，即 $u \in Gv$ 是否成立？一个相关的基本问题，在代数几何中更为自然（当域 $\mathbb{F}$ 是代数闭的，且特征为零时），是两个轨道的 *closures*[22] 是否相交（即，Gu 中的某些点是否可以被 Gv 中的点任意良好地逼近）。我们将回到这些问题的具体体现。

当 G 作用于 V 时，它也作用于 $\mathbb{F}[V]$，即 V 上的多项式函数，也称为 V 的 *coordinate ring*。在我们的设定中，V 将具有有限维度（例如，m），因此 $\mathbb{F}[V]$ 简单地是 $\mathbb{F}[x_1, x_2, \ldots, x_m] = \mathbb{F}[X]$，即 $\mathbb{F}$ 上的 m 变量的多项式环。我们将用 gp 表示群元素 $g \in G$ 对多项式 $p \in \mathbb{F}[V]$ 的作用。

[20] And so, every two polygons of the same area can be cut to produce identical (multi)sets of triangles.
[21] So there are pairs of 3-dimensional polyhedra of the same volume, which cannot be cut to identical (multi)sets of tetrahedra.
[22] One can take closure in either the Euclidean or the Zariski topology (the equivalence in this setting has been proved by Mumford [Mum95]).

一个多项式 $p(X) \in \mathbb{F}[X]$ 是 *invariant*，如果它不受此操作的影响；也就是说，对于每个 $g \in G$，我们有 $gp = p$。所有不变多项式显然形成 $\mathbb{F}[X]$ 的一个子环，记为 $\mathbb{F}[X]^G$，并称为此操作的 *ring of invariants*。理解群操作的不变量是不变量理论的主要主题。希尔伯特 [Hil93] 的一个基本结果表明，在我们的线性设置中，[23] *all* 不变量环将作为一个代数 *finitely generated*。找到“最简单”的这种生成不变量的集合是我们这里的主要关注点。

两个熟悉的问题完美解决方案的例子如下。

- 在第一个，$G = S_m$，由m个字母构成的对称群作用于m个形式变量X (的集合，因此通过排列它们生成它们所生成的向量空间)。然后，一组生成不变量简单地是X中的第一个m *elementary*个对称多项式。
- 在第二个，$G = SL_n(\mathbb{F})$，行列式为1的简单线性矩阵群作用于$M_n(\mathbb{F})$向量空间中的$n \times n$矩阵（因此 $m = n^2$），通过简单的左矩阵乘法。在这种情况下，所有多项式不变量都是由一个多项式生成的，即这个m-变量矩阵X的行列式。

在这些两种情况下，它们真正提供了对不变环 $\mathbb{F}[X]^G$ 的完整理解，生成集在几个意义上都是 *good*。存在 *few* 个生成不变量，它们都具有 *low* 次数，并且它们是 *easy* 可计算的——所有这些量都受一个关于 m 的多项式的限制，即向量空间的维度.[26] 在这些良好情况下，对于关于群作用轨道的基本问题，存在有效的算法。例如，几何不变理论的一个基本对偶定理（参见[MFK82]，定理A.1.1），说明了如何使用不变环的生成集来解决轨道闭包交点问题。

Theorem 13.23 [MFK82]. *For an algebraically closed field $\mathbb{F}$ of characteristic 0, the following are equivalent for any two $u, v \in V$ and generating set P of the invariant ring $\mathbb{F}[X]^G$:*

- *The orbit closures of u and v intersect.*
- *For every polynomial $p \in P$, $p(v) = p(u)$.*

在接下来的小节中，我们将看到，即使对于非常具体的自然群作用，试图理解轨道闭包交集及相关问题，也与计算复杂性美妙地相互作用。

13.9.1 Geometric complexity theory

这里简要解释了尽管一般计算不需要有任何对称性，但上述对群作用的研究可以与其核心问题非常相关！这项工作已经产生了许多新问题以及学科间的合作。首先，简要介绍算术复杂度理论的主要问题（参见第12章以获取定义和更多讨论）。Valiant [Val79b] 定义了复杂度类 $\mathcal{VP}$ 和 $\mathcal{VNP}$ 的算术类似物，并猜想这两个算术类是不同的（参见猜想12.21）。他进一步证明（通过令人惊讶的完备性结果），为了分离这些类，只需证明对于任何 $m =$ 多项式 n，$n \times n$ 矩阵上的永续多项式不投影到 $m \times m$ 矩阵上的行列式多项式即可。请注意，这是一个关于中心计算猜想的纯粹而具体的代数表述。

在一系列论文中，Mulmuley和Sohoni引入了*geometric complexity theory* (GCT)来解决这个重大未解问题。[27] Mulmuley [Mul11, Mul12a]以及Landsberg的书中[lan17]对该项目进行了概述。非常简洁地，GCT项目开始如下。首先，对$n \times n$进行简单的填充

[23] The full generality under which this result holds are actions of *reductive* groups, which I do not define here, but includes all examples we discuss.

[24] This means that there is a finite set of polynomials $\{q_1, q_2, \ldots, q_t\}$ in $\mathbb{F}[X]^G$, and for every polynomial $p \in \mathbb{F}[X]^G$, there is a t-variate polynomial r over $\mathbb{F}$ so that $p = r(q_1, q_2, \ldots, q_t)$.

[25] For example, they have *small* arithmetic circuits or formulas.

[26] There are additional desirable structural qualities of generating sets that we will not discuss, such as completely understanding the algebraic relations between these polynomials (called *syzygies*).

[27] The origins of using invariant theory to argue computational difficulty via similar techniques go back to Strassen [Str87].

永久多项式使其具有度m并作用于一个$m \times m$矩阵的元素。考虑所有此类$m \times m$矩阵上线性群SL_{m^2}的作用。这种作用扩展到这些变量的多项式，特别是我们关心的两个：行列式和修改后的永久。*The main connection is that the permanent projects to the determinant (in Valiant's sense) if and only if the orbit closures of these two polynomials intersect*。证明它们不相交（对于$m = \text{poly}(n)$)自然导致关于寻找此类交点（以及因此所需的计算下界）的表示理论障碍的问题。这使事情变得非常复杂，描述它们超出了本综述的范围。迄今为止，代数几何和表示理论的工具甚至不足以改善定理12.25中m的二次界。事实上，一些最近的发展表明，原始GCT方法存在严重局限性（也许引导它走向更富有成效的方向）；参见[BIP16]及其历史记载。尽管如此，这条攻击路线（在计算复杂性中还有其他路线）导致了计算共轭代数中的许多新问题，并促进了代数学家和复杂性理论家之间的合作——我们现在将考虑描述其中的一些。

为了做到这一点，我们关注线性群在矩阵 *tuples* 上的两个自然动作：同时共轭和左右作用。这两个都是 *quiver representations* (的特殊情况，参见 [Gab72, DW06])。[28] 对于这两种群作用，我们将讨论关于不变量环的经典问题和结果，以及由计算考虑所激发的最近进展。

13.9.2 Simultaneous conjugation

考虑以下 $SL_n(\mathbb{F})$ 对 d-元组 $n \times n$ 矩阵的作用。我们有 $m = dn^2$ 个变量，排列成 d $n \times n$ 个矩阵 $X = (X_1, X_2, \ldots, X_d)$。矩阵 $Z \in SL_n(\mathbb{F})$ 对此元组的作用是通过同时共轭，将其转换为元组 ($Z^{-1}X_1Z, Z^{-1}X_2Z, \ldots, Z^{-1}X_dZ$)。现在，关于此作用的一般问题是：在变量 X 中的哪些多项式在此作用下是不变的?

拉兹myslov、Procesi、Formanek和Donkin的工作[拉74、Pro76、For84、Don92]为在特征为零的代数闭域上生成不变量提供了一个好的集合（在上述大多数方面讨论的）。生成元仅仅是给定矩阵{v*}乘积长度最多为n^2的迹，即集合

$$\{Tr(X_{i_1}X_{i_2}\cdots X_{i_t}) : t \leq n^2, i_j \in [d]\}.$$

这些多项式是显式的，度数小，易于计算。唯一的缺点是生成集的大小呈指数级增长。例如，使用它来决定轨道闭包的交集将只会导致指数级时间算法。

通过希尔伯特存在诺特定理的诺特定理规范化引理 [Hil93][30]，我们知道这个生成不变量集的大小原则上可以减少到 $dn^2 + 1$。实际上，当群作用在一个维度为 m 的向量空间上时，取任何有限生成集的 $m+1$ “随机”线性组合（概率为1）将导致一个小的生成集。然而，由于我们开始时有一个指数数量的生成器，这个程序既低效又不够明确（不清楚如何使其确定性）。可以使用Gröbner基算法（参见 [MR11] 以获取最佳已知复杂度界限）得到一个最小大小的显式生成集，但这将需要双指数时间 n。

以上工作 [Mul12b,FS13] 将此复杂性降低到多项式时间！这发生在两个阶段。首先，Mulmuley [Mul12b] 通过巧妙地使用指数级众多不变量的结构（使用这些不变量，只需多项式数量的随机比特和多项式时间即可获得足够随机的线性组合）给出了一种概率多项式时间算法。然后，他论证说，使用条件去随机化结果，类似于第7.2节中讨论的性质，可以推导出一个确定性的多项式时间

[28] I will not elaborate on the theory of quivers representation here, but only remark that reductions and completeness occur in this study as well. The left-right quiver is *complete* in a well-defined sense (see [DM15], Section 5). Informally, this means that understanding its (semi)-invariants implies the same understanding of the (semi)-invariants of all acyclic quivers.

[29] Convince yourself that such polynomials are indeed invariant.

[30] This is the same foundational paper that proved the *finite basis* and *Nullstellensatz* theorems. It is interesting that Hilbert's initial motivation to formulate and prove these cornerstones of commutative algebra was the search for invariants of linear actions.

算法在自然计算难度假设下。不久之后，福布斯和什皮尔卡[FS13]展示了如何在不任何未证明假设的情况下去随机化Mulmuley算法的一个变体，从而得到一个对该问题的无条件确定性多项式时间算法。他们的算法使用去随机化方法：大致来说，他们首先注意到Mulmuley的概率算法可以通过一个非常受限的计算模型（一种特定的单次读取分支程序）来实现，然后使用一个高效的伪随机生成器来为这个计算模型生成随机数。这里有一个重要的算法推论（可以扩展到其他箭头）。

Theorem 13.24 [Mul12b, FS13].
There is a deterministic polynomial-time algorithm for solving the following problem. Given two tuples of rational matrices $(A_1, A_2, \ldots, A_d)$, *and* $(B_1, B_2, \ldots, B_d)$, *determine whether the closures of their orbits under simultaneous conjugation intersect.*

有趣的是，如果我们只考虑轨道本身（而不是它们的闭包）——也就是说，询问是否存在 $Z \in SL_n(\mathbb{F})$ 使得对于所有 $i \in [d]$ 我们都有 $Z^{-1}A_iZ = B_i$——这成为 *module isomorphism* 问题在 $\mathbb{F}$ 上。对于这个重要问题，存在一个确定性算法（与上述算法非常不同，使用其他代数工具），它可以使用 $\mathbb{F}$ 上的多项式次数算术运算来解决任何域 $\mathbb{F}$ 上的问题 [BL08]。

13.9.3 Left-Right action

考虑现在两个副本 $SL_n(\mathbb{F}) \times SL_n(\mathbb{F})$ 在 d-元组 $n \times n$ 矩阵上的作用。我们仍然有 $m = dn^2$ 个变量，排列成 d $n \times n$ 个矩阵 $X = (X_1, X_2, \ldots, X_d)$。一对矩阵 $Z, W \in SL_n(\mathbb{F})$ 对这个元组的作用是通过左右作用，将其转换为元组 $(Z^{-1}X_1W, Z^{-1}X_2W, \ldots, Z^{-1}X_dW)$。在变量 X 中的哪些多项式在这种作用下是不变的？尽管表面上与同时共轭相似，但这里的守恒量具有非常不同的结构，并且限制它们的大小需要不同的论证。

作品 [DW00,DZ01,SVdB01,ANS10] 提供了一组生成不变量的无限集合。生成元（再次，在代数闭域上）是 d 矩阵的线性形式的行列式，具有任意维度的 *matrix* 系数：

$$\{\det(C_1 \otimes X_1 + C_2 \otimes X_2 + \cdots + C_d \otimes X_d) : C_i \in M_k(\mathbb{F}), k \in \mathbb{N}\}.$$

这些生成器虽然描述简洁，但在上述大多数良好特性方面存在不足，我们现在讨论改进。首先，根据希尔伯特有限生成性，我们知道矩阵系数的维度存在一个特定的有限界限 k。Derksen 和 Makam [DM15] 在那里描述了一系列长期改进后，得到了一个二次上界 $k \leq n^2$。尽管如此，仍然存在指数数量的可能矩阵系数，其大小可以明确描述，并且允许随机性，可以将这个数量进一步减少到多项式。因此，例如，我们有以下关于此左右作用的轨道闭包交集的定理13.24的较弱类似物。

Theorem 13.25. *There is a probabilistic polynomial-time algorithm to solve the following problem. Given two tuples of* 合理的 *matrices* $(A_1, A_2, \ldots, A_d)$ *and* $(B_1, B_2, \ldots, B_d)$, *determine whether the closures of their orbits under the left-right action intersect.*

在本文本剩余部分，我们讨论这个问题的关键特殊情况，即当所有 $B_i = 0$，对于这种情况已经找到了一个 *deterministic* 多项式时间算法。虽然这个问题属于交换代数，但这个算法出人意料地在分析、非交换代数以及计算复杂性和量子信息理论等领域产生了影响。我们将提及其中一些，但让我们首先定义这个问题。

对于一个线性群 {v*} 在向量空间 {v*} 上的作用，定义该作用的 {v*} 为包含 0 的轨道 {v*} 的闭集 {v*}。零锥中的点有时被称为 {v*}。零锥在不变量理论中具有基本的重要性。我们已讨论的作用的零锥的例子如下。对于 {v*} 在 {v*} 上通过左乘的作用，它

[31] Well, a possibly infinite number, but it can be reduced to be exponential.

是 *singular* 矩阵的集合。对于 $SL_n(\mathbb{C})$ 对 $M_n(\mathbb{C})$ 的共轭作用，它是 *nilpotent* 矩阵的集合。正如你所猜想的（一个方向是平凡的），零锥恰好是所有不变多项式都消失的点集。因此，如果我们有一个好的生成集，我们可以用它来有效地测试零锥中的成员资格。然而，我们无法对左右作用这样做。尽管如此，[GGOW15] 中在复数上获得了一个确定性多项式时间算法，然后在 [IQS15] 中获得了一个非常不同的算法，该算法适用于所有域。这两个算法具有不同的性质和属性，并以不同的方式使用了不变量中矩阵系数维度的上界。

Theorem 13.26 [GGOW15, IQS15]. *There is a deterministic polynomial-time algorithm, that on a given tuple of matrices $(A_1, A_2, \ldots, A_d)$ in $M_n(\mathbb{F})$ determines whether it is in the nullcone of the left-right action.*

我们总结了一些该算法的多样后果。以下概念的所有精确定义，以及证明、相互关联和导致这些算法的曲折故事，都可以在[GGOW15, GGOW16]中找到。

Theorem 13.27 [GGOW15, GGOW16]. *There are deterministic polynomial-time algorithms to solve the following problems.*

- *The feasibility problem for Brascamp-Lieb inequalities, and more generally, computing the optimal constant for each.*
- *The word problem over the (non-commutative) free skew field.*
- *Computing the non-commutative rank of a symbolic matrix.*[32]
- *Approximating the commutative rank of a symbolic matrix to within a factor of 2.*[33]
- *Testing whether a completely positive quantum operator is rank-decreasing.*

[32] That is, a matrix whose entries are linear forms in a set of variables.
[33] Computing this rank exactly is the PIT problem discussed at the end of Section 12.4.

14 Space complexity: Modeling limited memory

尽管在微型化计算机存储器方面取得了显著的技术进步（我们习惯于在口袋里携带几GB的电影、图片和音乐），空间是一种昂贵的资源，其最小化在众多应用中至关重要，尤其是在处理“大数据”的应用中。本章的一个重要信息是，即使使用很少的内存，也能做些令人惊讶的事情。

我们首先描述空间受限算法的基本计算模型、研究的主要复杂度类以及空间复杂度经典理论中的一些最基本的结果和开放问题。然后，为了展示小空间计算的强大能力，我们采用两个框架进行。在第一个框架中，我们讨论了更现代的*streaming*空间受限计算模型，其中大量数据飞驰而过，只能看到一次。尽管如此，"压缩"它的方法允许在较小的空间中计算重要的统计数据。在第二个框架中，我们描述了关于可能最小空间的两项较老的结果，并出人意料地表明，*fixed*数量的内存足以计算任意大的数字。

14.1 Basic space complexity

空间复杂度与时间复杂度（本书中讨论的主要资源）几乎一样被深入研究，为了研究由空间受限算法解决的问题类别及其与时间和其他资源限制的关系，已经发展了一套复杂的理论。事实上，本书中已经提到了其中的一些类别和联系，例如，在第4.1节中定义的$\mathcal{PSPACE}$战略问题类别，以及基本定理10.3，即这个类别中的每个问题都拥有一个交互式证明。本书迄今为止讨论的大多数问题都在多项式空间内，一个主要问题是其中哪些问题可以在更少的空间内解决——亚线性、输入大小的对数或甚至常数。

模型计算使用如此少的内存，小于输入长度，应谨慎定义，以便空间限制精确到算法的 *working space*。标准模型简单地区分输入访问和工作内存访问。因此，在空间受限的算法（例如，图灵机）中，输入位于一个 *read-only* 磁带上，该磁带 *cannot* 可以修改，空间限制仅适用于算法通过读写访问的单独磁带（或磁带）。空间 $s(n)$ 捕获了可以使用此类算法（例如，图灵机）解决的问题类别，即在长度为 n 的输入上仅使用 $s(n)$ 位工作内存。对于需要长输出（超过空间限制）的问题，提供另一个单独的 *write-only* 磁带。

迄今为止，最重要的、研究最深入的空间限制复杂度类是 $\mathcal{L}$，它包括在 $O(\log n)$ 空间内可解的问题。我们在此列出该类中的一些基本问题。读者可能希望找到这些问题的（大多数都很简单）小空间算法。

- **Arithmetic problems.** 给定两个整数，计算它们的和、积以及一个除以另一个的余数。
- **Comparison problems.** 比较两个整数，对一组整数进行排序。
- **2-coloring.** 给定一个图，确定它是否是二分图。
- **Word problem in the free group.** [LZ77] 给定来自字母表 $\{a, a^{-1}, b, b^{-1}\}$ 的一个序列，确定它们的乘积是否是恒等元。

关于时间限制的复杂性，自然地扩展空间限制模型以允许概率性和非确定性计算是合理的.[1] 再次，建模应谨慎进行。为了正确计算空间，模型应指定如何访问随机/猜测位；它必须确保每个这样的位只能访问一次（算法可能或可能不想明确存储在工作内存中）。这种有限的

[1] As well as alternating, interactive, quantum, and so forth, which we will not discuss.

随机访问或非确定性将在我们下面讨论的这些模型的弱点中发挥关键作用，与它们的时间限制类似物相比。

概率和非确定性类似物 $\mathcal{L}$ 分别表示为 $\mathcal{BPL}$ (概率对数空间，类似于 $\mathcal{BPP}$)，和 $\mathcal{NL}$ (非确定性对数空间，类似于 $\mathcal{NP}$)。空间上界时间，而时间永远不会超过空间指数的基本观察结果，导致以下时间与空间类之间的包含链：

$$\mathcal{L} \subseteq \mathcal{NL} \subseteq \mathcal{P} \subseteq \mathcal{NP} \subseteq \mathcal{PSPACE} \subseteq \mathcal{EXP}.$$

就像更多时间购买更多计算能力（例如，$\mathcal{P} \neq \mathcal{EXP}$）一样，更多空间购买更多计算能力（例如，$\mathcal{L} \neq \mathcal{PSPACE}$）。这些分离意味着上述一些直接包含是严格的，但我们不知道是哪一个。

在很大程度上，空间复杂度比时间复杂度更容易理解。让我们提及一些这样的基本结果及其直观含义（不正式定义相关类别）。其中大部分都可以简单地理解为以下两种变体的低空间算法，即有向和无向的 *graph connectivity* 问题。

问题 $DCONN$ 和 $UCONN$ 定义如下。在这两个问题中，输入是一个图 G，以及两个特殊节点标记为 s 和 t，问题是要确定是否存在从 s 到 t 的路径 G。唯一的区别在于，在 $DCONN$ 中，输入图是 *directed*，而在 $UCONN$ 中是 *undirected*。$DCONN$ 对 $\mathcal{NL}$ 扮演的角色与 SAT 对 $\mathcal{NP}$ 扮演的角色相同；它是此类问题的完全问题。当然，必须定义一个适当的归约概念；在这里，对数空间归约代替了多项式时间归约。$DCONN$ 的完备性简单地源于这样一个事实：在计算对数空间机器时，只有多项式数量的不同 *configurations*，并且给定输入之间的转换自然地由一个有向图描述。如果一个输入被接受，那么从初始配置可以到达一个接受配置。$DCONN$ 有多项式时间算法的简单事实解释了上述提到的包含 $\mathcal{NL} \subseteq \mathcal{P}$。

无向版本 $UCONN$ 也通过与上面类似的论证，在称为 $\mathcal{SL}$ 的一个类中作为一个完整问题发挥了重要作用。它是 $\mathcal{BPL}$ 中问题的重要例子之一；但正如我们下面将看到的，由于下面的定理 14.3，这个问题和这个类现在不那么特殊了。

我们以对非确定性（或猜测）在日志空间范围内的两个上界开始。回想一下，对于时间复杂度的类似结果尚不清楚，实际上，人们并不相信。第一个，是复杂性理论中最古老的结果之一，归功于Savitch [Sav70]，是$\mathcal{P} = \mathcal{NP}$的类似物。在这里，可以通过在空间中平方爆炸来消除非确定性。这是通过证明$DCONN$，即$\mathcal{NL}$的完备问题，可以使用仅（log n）^{2}空间确定性解决来实现的。以下我们使用$\mathcal{L}^t$表示可以使用空间（log n）^{t}解决的问题，对于任何有理数t。

Theorem 14.1 [Sav70]. $\mathcal{NL} \subseteq \mathcal{L}^2$.

下一个结果，独立于Immerman [Imm88] 和Szelepcsényi [Sze88]，是空间复杂度下 $\mathcal{NP} = \text{co}\mathcal{NP}$ 的强类比。它表明存在性和全称量词可以在仅 *linear* 的空间成本下交换。等价地，存在一个针对以下任务的确定性对数空间算法；它接收一个有向图作为输入，并产生另一个作为输出，使得一个有 $s-t$ 路径当且仅当另一个没有！对于那些接受寻找此类算法挑战的人的一个提示：它取决于 *counting* 从一个节点到另一个节点的路径数量。

Theorem 14.2 [Imm88, Sze88]. $\mathcal{NL} = \text{co}\mathcal{NL}$.

我们接下来转向无向连通性。Reingold [Rei08] 的一个突破性结果是 $UCONN$ 的确定性对数空间算法。这是现有最复杂的图算法之一，它以本质和令人惊讶的方式使用扩张图和伪随机性。更抽象地说，该定理表明 *symmetric* 非确定性不会为空间受限算法增加任何能力。

Theorem 14.3 [Rei08]. $\mathcal{SL} = \mathcal{L}$.

我们得出结论，从空间限制算法中去除随机性的（非常低）成本，即去随机化 $\mathcal{BPL}$。一切始于一个 *unconditional* 伪随机生成器的开创性构造

对于Nisan [Nis92]的概率空间有界计算。再次强调，与时间复杂度（比较第7.3节中的定理7.14、7.15）不同，这里没有硬度假设。然而，从伪随机性和硬度之间的联系来看，Nisan构造的核心是一个*provable*下界（更多内容请参见下一节14.2的结尾）。为了陈述定理，我们注意到，对于空间，伪随机性的定义与我们定义时间伪随机性的方式相同，但考虑到空间有界计算中随机性的单向访问。我们说，在n位上的分布*fools*一个对数空间机器，如果对其位的单向访问不能与对n位上的均匀分布的单向访问区分开来（例如，优势为1/poly(n)）。

Theorem 14.4 [Nis92]. *There is a log-space computable generator $G : \{0,1\}^m \to \{0,1\}^n$, with $m = O((\text{对数 } n)^2)$, that fools log-space computations.*

以下两个定理都通过以非常不同的方式利用上述Nisan生成器而得出。两者都可以被视为（不可比较的）$\mathcal{BPP} = \mathcal{P}$的类似物。在第一个结果中，类$\mathcal{SC}$包含所有可以通过多项式时间且多项式对数空间确定性算法解决的问题。

Theorem 14.5 [Nis94]. $\mathcal{BPL} \subseteq \mathcal{SC}$.
Theorem 14.6 [SZ99]. $\mathcal{BPL} \subseteq \mathcal{L}^{3/2}$.

从我们所知，让我们转向我们想要了解的内容。在基本空间复杂度中，最突出的问题可能是随机性和非确定性在此模型中的确切力量。以下猜想反映了普遍的信念，即随机性根本不增加任何力量，而非确定性则不然。

Conjecture 14.7. $\mathcal{BPL} = \mathcal{L}$.

一个优雅的“完整”图连通性问题变体，其中对数空间算法将证明这个猜想，在[RTV06]中给出。

Conjecture 14.8. $\mathcal{NL} \neq \mathcal{L}$.

算法上，这个猜想表明 $DCONN$ 有一个确定性对数空间算法，与它的无向兄弟 $UCONN$ 不同。

当前情况下，我们不知道任何自然函数（例如，在$\mathcal{NP}$中）需要超过对数空间来计算。回想一下，我们也不知道任何自然函数需要超过线性时间来计算。因此，当Fortnow [For00]给出了一个非常简单的证明，即 SAT，最基本$\mathcal{NP}$-完备问题，在上述空间或时间意义上至少对于空间或时间来说很难时，这就像一颗重磅炸弹。事实上，这个结果要强得多：使用近线性时间的算法必须使用几乎线性空间！

Theorem 14.9 [For00, FLvMV05]. *For every fixed $\epsilon > 0$, there exist $\delta > 0$, such that any algorithm solving SAT that uses $n^{1-\epsilon}$ space requires time at least $n^{1+\delta}$.*

此类结果被称为 *time-space trade-offs*，除了图灵机之外，在许多其他模型中都有所了解，包括均匀和非均匀模型。我们将在第15.2.2节中讨论另一个这样的权衡。另见[BSSV03]及其历史概述。

14.2 Streaming and sketching

一个令人兴奋且具有挑战性的空间有限模型，其重要性因大数据应用而不断增长，就是 *streaming* 模型（参见综述 [Mut05, Cor11, McG14]）。与允许对输入进行多次访问的经典空间有限模型不同，在这里，输入“飞逝”，而在有限的空间（远小于输入长度）中未存储的内容将永远消失。常被引用的一个例子来展示这种模型所激励的情况是瑞士大型强子对撞机（LHC）的实验，最近因确立希格斯玻色子的存在而闻名。几乎所有的LHC探测器从加速器中高能碰撞的碎片中记录的大量数据都因空间不足而瞬间丢弃，只有一小部分被保留；所使用的复杂算法试图仅保留对新现象发现至关重要的信息。不用说，在许多其他天文、生物学和

其他科学，以及在网络流量、财务信息、天气等方面的观察，都需要。在几乎所有情况下，只需要关于数据的基本统计或结构信息。

一个接地气的例子，是这个领域的第一个激励性例子，如下所示。假设通过输入的是 $x = (x_1, x_2, \ldots, x_n)$，其中每个 x_i 是范围 $[n] = \{1, 2, \ldots, n\}$ 中的一个元素。一个基本问题是 x 中有多少个 *distinct* 元素。当然，使用线性 $O(n)$ 空间，可以存储 x 并回答这个问题，但如果空间是次线性的呢？很明显（但需要证明），无法精确且确定地解决这个问题，实际上，这个领域的多数算法都允许近似和随机化。例如，对于上述问题，你认为需要多少最小空间，以至少 99% 的概率给出一个在正确数量的不同元素 1% 以内的估计？答案是惊人的小：$O(\log n)$ 比特足以进行如此高质量的估计，并且具有如此高的置信度。这样的算法是由 Alon、Matias 和 Szegedy [AMS96] 发现的，读者被邀请尝试找到一个次线性空间算法。

确实，[AMS96] 和后续论文研究了计算分布 x 的其他“频率矩”的算法。更确切地说，输入 x 定义了一个直方图 $h = (h_1, h_2, \ldots, h_n)$，其中 h_i 计算具有值 i 的 x_j 的数量。x 的不同元素的数量简单地是 h 的非零条目的数量，换句话说，h 的“0-范数”，表示为 $|h|_0$。关于此类数值数据的最有价值的统计信息是 h 的其他范数（或矩）$|h|_p$ 或该分布的熵等。碰巧的是，其中一些可以在小空间中准确估计，而另一些则不能，其中空间的上限和下限通常需要相当复杂的技术，并导致有趣的联系（例如，到稳定分布、度量嵌入、稀疏恢复等主题）。在流算法中最常用的方法是维护一个 *sketch* (或指纹)，这是一个小空间数据结构，它捕获了关于迄今为止看到的输入部分足够多的信息，并且在新数据项到达时易于更新。不用说，为给定问题找到一个合适的草图是非常不平凡的。关于流和草图以及这些和其他类型的问题（例如，在图上近似参数）的更多内容，可以在上述调查中找到。

让我给大家演示一下草图在实际中的应用（高级）。这个优雅的选择，用于逼近上述 L_2 范数，是由 Indyk [Ind00] 提出的。再次，读者可能会考虑在非常小的空间内完成这一挑战，在继续阅读之前，这似乎是不可能的任务。下面是这个算法。在看到输入之前，我们一次性在 $v \in \{-1, 1\}^n$ 中选择一个随机符号向量。草图将是一个整数 z，初始化为 0。它将随着每个输入 x_i 的到来进行更新：如果 $x_i = j$，我们只需将 v_j 添加到 z。通过线性性质很容易看出，在这个算法结束时，z 的值仅仅是 v 和 h 的内积。由于 v 是一个随机符号向量，z^2 的期望值是 $|h|_2^2$，并且它高度集中在均值附近。因此，为了以高概率获得准确的估计，我们应用了通常的技巧：维护几个独立的向量 v 和每个向量的草图 z，并在算法结束时报告它们的平均值。请注意，每个 z 计数器只需要 $O(\log n)$ 的空间。我们完成了吗？等等（我听到你这么说）——那么随机向量 v 呢？存储它们需要线性空间！实际上，并不需要——Indyk 在他的论文中展示了如何使用定理 14.4 中的 Nisan 空间伪随机生成器来生成和存储足够好的替代向量，仅使用 $O(\log n)^2$ 的空间，这为第 7 章中讨论的去随机化探索提供了另一个应用。

甚至从这个简单的例子中，关于流和草图“思维方式”的一些后果是显而易见的。一是草图不仅可以提供输入结束时的非常准确的估计，实际上在整个输入过程中，在添加每个新符号之后，都可以提供这样的估计。因此，输入长度不需要预先固定，可以视为无限。另一个是输入可能不同于科学观察的一些原始数据。例如，它可能是一个对象的更新序列（例如，随着时间的推移，某些链接和节点丢失或添加的网络），草图可以捕捉网络的多种连接属性。因此，草图产生了新的数据结构，用于处理输入随时间变化的动态问题。

让我们以一个简单的流任务结束，该任务在较小的空间内解决 *cannot*；实际上，这正是定理14.4中Nisan的伪随机生成器所基于的问题。两个长度为$10s$的随机序列，首先是 x，然后是 y，飞驰而过，任务是计算它们的内积模2。尝试证明以下陈述：任何空间 s 算法猜测它们的内积的概率，

无法超过平凡的 $\frac{1}{2}$ 超过 2^{-s}。这样的下界通常使用 *communication complexity*，第 15 章的主题，在那里我们进一步讨论 Nisan 的生成器。

14.3 Finite automata and counting

内存最严重的限制是将其通过一个与输入大小无关的绝对常数进行限制。本节说明即使这样的有限模型也具有惊人的能力，这反过来又证明了一个重要观点：某些下界可能难以证明，仅仅是因为它们是错误的。例如，以下两个算法是在试图证明它们不存在时发现的。我们还将看到，这种非常弱的计算模型的能力远未完全理解；事实上，关于它的某些未解问题与上面讨论的标准空间复杂度类问题密切相关。

“不可能”的任务展示了这种力量，这是 *counting* 的基本问题。证明使用固定数量的内存进行任意高度的计数是不可能的并不困难。这种直觉似乎也适用于可以掷随机硬币的有限内存设备，或者获得一些输入无关的“建议”（请检查你是否相信这些直观的陈述）。但我们将看到，在这两种情况下，只需要少量内存（比如说，10位）就可以确定长度为 *any* 的二进制序列中1的个数是否多于0的个数（在概率设置中，以高概率，比如说 2/3）。

让我们更方便地描述计算模型。当图灵机的工作空间 $s(n)$ 的大小是常数（即，与输入大小 n 无关）时，它可以简单地被视为机器的有限控制的一部分。换句话说，图灵机变成了一个 *finite automaton*。有限自动机在计算机科学的历史中发挥了重要作用。形式定义、中心结果以及它们在计算理论早期发展中的作用，可以在Sipser的优秀教科书中找到[Sip97]。

我们从双向确定性有限自动机（2DFA）开始，这恰好是一个具有常量空间的图灵机。“双向”强调输入带可以双向扫描（像图灵机一样，但与我们将很快遇到的其它自动机不同）。有限自动机有一个有限的控制（或程序），它被有限数量的状态所捕获。自动机最初处于某个起始状态，其读取头位于（比如说）左侧输入符号。在每一步，单元格内容和当前状态决定了自动机的下一个状态（如果它进入接受或拒绝状态，则它停止），并且如果头向左或向右移动（我们假设输入的左右端点是可检测的）。总结来说，2DFA只是一个没有写入能力的图灵机。

一个2DFA无法计数！考虑多数函数，它对于二进制序列计算它是否比0多1。尝试证明具有n状态的2DFA甚至不能计算长度为$2n$的序列的多数。因此，我们将根据上述讨论增强我们的2DFA，使其具有额外的功能。因此，让我们尝试给我们的有限自动机添加力量或额外功能：非确定性、交替、随机性和建议，并发现这一活动的起源，这是我们领域最基本的活动（正如我们在整本书中看到的那样）。

但在开始之前，让我们讨论1DFA，它是2DFA的较老版本。这个模型由Kleene [Kle51] 定义，以理解McCulloch和Pitts [MP43] 的“神经网”，这是神经元和大脑的第一个数学模型。Kleene证明了1DFAs精确计算 *regular* 语言：具有强周期性结构的序列集合.[2] 这种特征很容易推断出多数语言不是正则语言，也不能由1DFA计算。

第一个系统地探索不同能力有限自动机模型相对能力的学者是Rabin和Scott，他们在开创性论文[RS59]中设定了后来许多其他计算模型研究的范例。特别是，Rabin和Scott [RS59] 定义了2DFAs，他们的一项主要结果是，对于确定性计算，这两个模型在能力上是相等的.[3] 这证明了

[2] A *regular language* S (over an alphabet Σ) is either a finite set of sequences over Σ, or can be obtained from previously defined languages as the union of two, $S = S_1 \cup S_1$, the concatenation of two, $S = S_1 S_2$ (concatenating any sequence of S_1 to one in S_2), or the *Kleene star* of one, $S = S_1^*$ (namely, a concatenation of any finite number of sequences of S_1).

[3] This is far from obvious, as a 2DFA can access the input bits arbitrarily many times, whereas a 1DFA sees each input bit for one step, and then it is gone. Try proving it!

2DFAs，就像1DFAs一样，精确地计算正则语言。特别是，2DFA也无法计算多数。

在相同的论文中，Rabin 和 Scott 建议通过允许从每个给定状态和单元格内容进行多个可能的转换，将 *nondeterminism* 添加到模型中（这启发了后来将非确定性添加到图灵机中，例如，当将 $\mathcal{P}$ 扩展到 $\mathcal{NP}$ 时）。他们还证明了由此产生的模型，称为“2NFA”，不能计算比其确定性兄弟更多的集合。[4] 该模型进一步扩展，以允许 *alternation* (，就像在 [LLS84] 中定义的多项式时间层次结构 $\mathcal{PH}$) 中的单个存在量词被扩展到交替存在和全称量词一样。然而，他们证明，即使这个模型，称为 2AFA，仍然不能计算比 2DFA 更多的集合，即只有正则语言。总结来说，从最弱的决定性单向自动机 1DFA 开始，通过升级为双向输入访问、非确定性和甚至交替，并不会增加计算能力。

因此，我们将 *randomness* 添加到模型中，分别创建了一向和双向概率有限自动机（1PFA 和 2-PFA）。与图灵机一样，这些模型允许抛掷完美的随机硬币并在计算中使用它们（即，在状态之间进行随机转换），并且要求在每次输入上以高概率（例如，2/3）计算正确答案。这个模型能比上述所有模型做更多吗？提到的结果似乎暗示，如果空间是常数，那么很难利用额外的能力，如非确定性交替。此外，在单向模型中，Rabin [Rab63] 证明了 1PFA 不能比 1DFA 做更多，因此在该设置中添加随机性也没有帮助：1PFA 只能计算正则语言。因此，当 Freivalds [Fre81] 证明双向概率有限自动机 *are* 更强时，这令人惊讶。特别是，2PFAs 可以计算任意高的数。

Theorem 14.10 [Fre81]. *There is a 10-state 2PFA that computes Majority with probability $\geq 2/3$ on every input. Moreover, for every $\epsilon > 0$, there is an integer $c = c(\epsilon)$ and a c-state 2PFA that computes Majority with probability $\geq 1 - \epsilon$ for every input.*

我们将在本节末尾解释该算法背后的简单思想，留给你时间和空间去尝试自己弄明白。

我们继续向双向自动机添加一个不同的特性：*nonuniformity*。非均匀性在第5章中讨论。在那里，布尔电路的非均匀模型被证明在配备（输入无关的）*advice*时与均匀图灵机等价。更具体地说，在那里和这里，我们允许机器在给定长度为n的输入时，能够（只读）访问一个外部（建议）带，其长度为n中的某个固定多项式。这样的非均匀机器在对于每个n，都存在一个二进制（建议）序列α_n，如果它位于建议带上，就会使机器在所有长度为n的输入上给出正确答案（如果建议带上包含任何其他字符串，则没有要求）。请注意，建议序列α_n当然可以依赖于所计算的功能，但由于非常短，不能包含长度为n的每个可能输入的答案。

因此，如果你只有10（甚至一百万）位的内存，并且需要在一个随机的n位序列上计算多数，那么一个长度为n^{10}的建议序列能为你做什么好处呢？致力于这个问题的社区一致认为这样的简短建议是无用的，并且实际上，证明建议字符串长度的指数下界不应该很难（实际上，如果机器只有*1 bit of memory*，这样的下界已经被证明了）。因此，当Barrington（他试图证明这样的下界）宣布这项任务可行[Bar86]时，这让人感到震惊。事实上，在这个模型中，只需要*3 bits of memory*就可以计算多数！还有许多其他函数也是如此！

Theorem 14.11 [Bar86]. *There is a 5-state nonuniform 2DFA[5] that computes the Majority function. In-*

[4] They actually proved it for 1NFA, simulating it by a 1DFA, but the same proof works for 2NFA. Note that this simulation of nondeterministic automata by deterministic ones incurs exponential blow-up in the number of states. This is known to be tight for 1NFA, but it is completely open for 2NFA. Indeed, proving a super-polynomial lower bound on the number of states in such a simulation will prove Conjecture 14.8. See [Pig13] for a survey of this approach.

[5] In accordance with our notation in Definition 5.2 for (nonuniform) circuits, we might denote this nonuniform automata by 2DFA/poly, to stress that the advice length is polynomial in the input length. The more common name for this model is *constant-width, polynomial size branching programs.*

deed, the same holds for every function computed by a polynomial-size Boolean formula.

我怀疑这个证明（它简短而甜蜜）比之前的那个更难重新发现，所以让我给你一些提示。一个重要的提示，也是证明的主要洞察，是关键地使用一个不可解群（状态数5与5个字母的交错群是最小的不可解排列群有关）。另一个提示是，构建的自动机*reversible* (不仅从读取的输入符号中唯一确定下一个配置，就像任何确定性机器一样，而且从下一个)确定前一个。最后，不要试图证明关于多数的第一个陈述，而是第二个，因为它允许对计算公式的结构进行归纳。2PFA使用的建议应解释为输入位在什么步骤要读取的序列；建议序列的长度是公式大小的平方。

让我再对这个显著的结果添加两条评论。首先，它激发了一个类似的结果用于算术计算，展示了如何使用固定数量的寄存器（实际上，三个就足够了）来评估一个算术公式[BOC92]。其次，公式评估的可逆性方面对于在密码学中应用定理14.11至关重要，始于[GMW87, Kil88]。

现在让我们回到概率自动机，并以Freivalds定理14.10背后的思想作为结论。你有（比如说）10位内存和一枚可以抛掷的硬币。面对一个二进制序列（为了简单起见，假设长度为奇数），从左到右扫描它，并对你观察到的0和1执行以下“锦标赛”。对于你看到的每个比特抛掷一枚硬币，并分别记录它是否为所有0（称此事件为W_0），或者是否为所有1（称此事件为W_1）。这只需要每个事件一个比特的内存。如果两个事件都没有发生或都发生了，再次进行这场锦标赛（这个决定需要更多一些的内存）。如果恰好只有一个事件发生，宣布该事件为失败者（即如果W_0发生，输出存在1的大多数，反之亦然）。分析很简单：少数比特事件发生的概率至少是多数比特事件概率的两倍，因此这个算法将以$2/3$的概率是正确的。为了提高成功概率，并证明定理的第二部分，只需为每个输入比特抛掷t枚硬币。这将需要t额外的状态，但将提高两个事件概率比从2提高到2^t。

该算法存在一些令人不满意的性质。首先，它可能永远不会停止。这是必要的；事实上，总是停止的2PFA只计算正则语言，正如原始论文[Fre81]所证明的。当然，以概率1它确实会停止，但以期望指数时间。这也同样是必要的；Dwork和Stockmeyer[DS90]证明，如果一个2PFA以期望多项式时间停止，那么它只计算正则语言。

一个仍然悬而未决的有趣问题是这种概率多项式时间、常量空间模型在交互式证明设置中的能力，即当双向自动机允许随机性和非确定性转换时。这个模型是否只计算正则语言？这个问题在[DS92]中被提出，并在[CHPW98]中取得了进一步进展。

15 Communication complexity: Modeling information bottlenecks

此通信复杂度研究领域研究将输入分配给两方的计算 $discrete$ 函数的通信成本。它始于1979年，由姚 [Yao79] 发表的一篇论文，紧随Abelson对 $continuous$ 函数的类似研究。该领域在数学和应用方面都迅速发展，Kushilevitz和Nisan [KN97] 的综合著作涵盖了前二十年的活动。如今，在那本书出版后的二十年里，迫切需要一本新书来总结自那时以来所取得的惊人成果，包括一些最新的突破，包括解决非常古老的问题和令人兴奋的新方向。

本章的目的不是总结所有这些内容，而是专注于通信复杂度模型的一个特征：其多功能性。当我还是学生的时候，我目睹了另一位学生和姚期智关于这个模型的对话。由于姚期智的原论文几乎没有任何动机，这位学生问他为什么计算机科学家应该研究这样一个简单、风格化的模型，这个模型纯粹是信息论的，并且特别忽略了所有的计算方面。姚期智的回答很简单：因为它就是 $basic$。对我来说，开始研究生涯的这个回答是一生的教训。随后，许多不同 $computational$ 设置的发现，这个模型提供了关键洞见，一次又一次地证实了这个教训。在本章中，我们将看到这些应用在VLSI设计、拍卖理论、电路复杂性、线性规划、伪随机性、数据结构等领域。不用说，通信 is 是分布式系统中的一个重要计算资源——但在这些应用中，我们将看到通过简单或微妙的简化，它使我们了解其他计算资源，如时间、空间、大小、随机性、查询和芯片面积。不出所料，在几乎所有情况下，证明都是通过一个具有以下性质的简化进行的：使用给定资源（或资源组合）过少的计算模型被证明会表现出通信瓶颈，这允许将这种计算转换为相关通信任务的廉价通信协议。

所有这些之后，我无法抗拒描述一些新的激动人心且基础的研究工作，这些研究实际上进一步探讨了通信复杂性与信息与编码理论之间的联系：通信协议的 $compression$ 和 $error\ correction$。

15.1 Basic definitions and results

我在这里简要概述了一些通信复杂性的基本方面，旨在追求简洁而非普遍性，以及我们在下面讨论的应用中将要使用的结果。正如所提到的，那个时代的全面文本是[KN97]，而关于这个不断发展的领域的不同方面的更多近期优秀综述包括[Lov14,She14,Rou16]。在这里，我只介绍姚的基本模型。在下面不同应用的要点中，我们将看到每个是如何在这个简单的基本主题上呈现出不同的变体。

一个通信问题简单来说是一个具有两个参数的函数，$f : X \times Y \to Z$，其中 X, Y 和 Z 是有限集合。给 Alice 一个输入 $x \in X$，给 Bob 一个输入 $y \in Y$。他们应该共同通过依次交换信息位来计算 $f(x, y)$，根据之前商定的协议。他们的计算能力没有限制；我们唯一关心的是最小化通信成本。我们以下形式化这些概念，首先是确定性协议，然后是概率性协议。在计算复杂度理论的 spirit 下，从图灵机借用了除确定性和概率性之外的“模式”，并将其适应到通信模型中，包括非确定性、交替、量子、Arthur-Merlin 等。实际上，Babai、Frankl 和 Simon [BFS86] 创立了相关复杂类定义和系统研究。我们这里不会讨论这些扩展。

一个 $deterministic\ \ protocol$ 为两位玩家中的每一位指定下一个要发送的比特，作为他们输入和到目前为止的通信历史的函数，以及终止时的输出。它可以自然地描述为一个二叉树，其中内部顶点由布尔函数在 X 或 Y(上标记，具体取决于在此节点说话的是谁)，而叶子节点由输出值（在 Z 中）标记。协议的通信复杂度（或成本）简单地是其深度（计算最大数量的节点）

任何输入上交换的比特数）。它计算一个函数 f，如果对于每个输入对 (x, y)，玩家在此输入上的通信路径到达一个标记为 $f(x, y)$ 的叶子节点。通信函数 f 的 *deterministic communication complexity* 是计算它的最低成本协议的成本。我们用 $D(f)$ 表示这个数量。

一个方便查看通信函数 f 的方法是将其视为一个矩阵 M_f，其行由 X 的元素标记，列由 Y 的元素标记，并且 (x, y) 项是 $f(x, y)$。理解通信协议对这一矩阵的影响是方便的。一个基本的洞察是，当发送第一个比特（例如，从 Alice 到 Bob）时，其值将 X (Alice 的可能输入) 分成两部分，从而创建一个子矩阵，它们在此基础上进行。在其中，Bob 的下一个比特到 Alice 将 Y 分割，以此类推。随着这一过程的继续，我们看到每个 c-比特协议都会将矩阵 M_f 分割成（最多）2^c 个“组合矩形”，其中矩形是子集 $X' \subseteq X$ 和 $Y' \subseteq Y$ 的笛卡尔积。此外，如果协议计算 f，则该分割中的所有矩形都是 *monochromatic*，即由 Z 的唯一元素标记。这一洞察是所有确定性下界的基础。一个特别有用的观察，归功于 Mehlhorn 和 Schmidt [MS82]，是如果我们把 Z 视为一个域 K 的子集，那么 $D(f) \geq \log \text{rk}_K(M_f)$，其中 rk_K 表示该域中的秩函数.[1] 我们很快将使用的一个有用观察是，当矩阵是具有非零对角线的三角形时，其秩是满的。

让我们看看一些自然函数的例子，其中一些将在下面出现，并对模型有一些直观的认识。在所有情况下，我们取 $X = Y = \{0, 1\}^n$，和 $Z = \{0, 1\}$。通信复杂度作为输入大小 n 的函数来衡量。从一开始就注意，在这个模型中，$n + 1$ 是每个通信函数的通信复杂度的上界（因为其中之一可以向另一个发送其输入，它将计算答案并发送回来）。

1. **Equality:** $EQ(x, y) = 1$ 当且仅当 $x = y$。
2. **Greater-or-equal:** $GE(x, y) = 1$ 当且仅当 $x \geq y$。
3. **Disjointness:**[2] $DISJ(x, y) = 0$ 当且仅当对于某些 i，$x_i = y_i = 1$。

对于所有三个函数，它们的矩阵都是三角形的[3]，对角线上的元素为1，因此根据上述秩下界，它们的确定性通信复杂度本质上都是最大的。

Fact 15.1.

1. $D(EQ) \geq n$。 2. $D(GE) \geq n$。 3. $D(DISJ) \geq n$。

我们现在允许玩家抛硬币，这增加了相当大的能力（有时）。一个 *probabilistic protocol* 简单地是一个在确定性协议上的分布。其成本是支持此分布的任何树的最大深度。这样的协议在对于每个输入对 (x, y)，如果从这个分布中随机选择一个协议到达标记为 $f(x, y)$ 的叶子节点的概率至少为 $1 - \epsilon$，则计算一个函数 f 并具有错误 ϵ。与之前一样，通信函数 f 的 *probabilistic communication complexity* 是以这种意义计算它的最低成本的概率协议的成本。我们用 $R_\epsilon(f)$ 表示这个量。在大多数情况下，我们选择 $\epsilon = \frac{1}{3}$，并让 $R(f) = R_\epsilon(f)$ (通过独立重复，如概率算法中的那样，实现错误减少。在许多应用中非常有用的概率通信复杂度的下界通常需要更复杂的技巧；其中一些及其相对能力在 [KMSY14, LS09] 中讨论。

[1] How tight this lower bound is in general is the subject of the notorious *log-rank conjecture*; see [Lov14] for history and state-of-the-art.

[2] Here x, y are viewed as characteristic vectors of subsets of $[n]$.

[3] For disjointness, it is the bottom *right* of the matrix that is all zeros, when rows and columns are sorted lexicographically.

[4] This is sometimes called the *shared randomness* model (which samples for both players the protocol to use). A *private randomness* model (in which each player tosses its own coins) is also used, but the two differ very slightly in complexity, within additive $O(\log n)$ [New91].

三个上述函数的概率通信复杂度，在确定性模型中是等价的，结果却彼此非常不同。

Theorem 15.2.

1. $R(EQ) = O(1)$.log n) *(e.g., see [Vio15]).*
2. $R(GE) = \Theta($
3. $R(DISJ) = \Theta(n)$ *[KS92, Raz92, BJKS04].*

这些定理中的上界非常简单，突出了 *hashing* 在随机协议中的重要性。例如，对于 EQ 的第一个结果，假设 Alice 有 x，Bob 有 y，并且他们共享一个长度为 n 的公共随机序列 r。然后 $\langle x, r\rangle$ 和 $\langle y, r\rangle$ 分别提供 x, y 的 1 位哈希值，如果 $x = y$，则它们总是相等的，如果 $x \neq y$，则它们以概率 $\frac{1}{2}$ 不同。对于 GE 的第二个结果，可以在两个输入的段上使用相同的哈希思想，通过二分搜索发现 x 和 y 不同的最显著位，从而确定哪个更大。当然，对于第三个，上界是平凡的。

下界问题值得更深入的讨论。概率下界通常更难证明，在应用中更有用。几乎所有这类证明的第一步（以及许多其他类型的概率算法）遵循所谓的姚氏最小-最大原理[Yao77]，即考虑以下对偶问题。我们不是关注针对最坏情况输入的最佳概率协议，而是关注针对平均情况输入的最佳确定性协议。这允许在输入随机选择并遵循某种分布时，对确定性协议证明一个下界。

更精确地说，设 μ 是 $X \times Y$ 上的任何分布。在分布 μ 下，函数 f 的 *distributional* 通信复杂度，表示为 $C_{\mu,\epsilon}(f)$，是在一个协议中正确计算 $f(x, y)$ 的概率为 $1 - \epsilon$ 时所需的位数，当 (x, y) 按照分布 μ 抽取时。容易看出，对于每个分布 μ，我们都有一个下界 $R_\epsilon(f) \geq C_{\mu,\epsilon}(f)$。此外，正如 Yao [Yao77] 指出的，对于某些分布 μ^*，等式成立，这些分布实现了 $R_\epsilon(f) = C_{\mu^*,\epsilon}(f)$；这种设置仅仅是 von Neumann 的零和博弈最小-最大定理 [vN28] 的一个特例。因此，任何分布都会给出一个下界，在这个方法中，没有一般性的损失，因为某些分布会给出最优下界。再次，当我们将其设置为等于 $\frac{1}{3}$ 时，我们抑制 ϵ；即，我们表示 $C_\mu(f) = C_{\mu,1/3}(f)$。

选择一个好的分布不是一个明显的问题。让我们考虑上面的下界。对于第二个在 GE 上的下界，选择 $\mu = \mu_X \times \mu_Y$ 为任何 *product* 分布（其中 x 和 y 是独立选择的）将导致 [5] $C_{\mu_X \times \mu_Y}(GE) = O(1)$。然而，对于选择 μ 使两个输入 (x, y) 相关的情况，可以得到一个紧的下界 $C_\mu(GE) = \Omega($log $n)$ [Vio15]。对于 $DISJ$ 的第三个下界，再次取一个乘积分布是不够的；[BFS86] 证明了 $C_{\mu_X \times \mu_Y}(DISJ) = O(\sqrt{n}$ log $n)$。相关选择 μ 导致了上面的最优结果：$C_\mu(DISJ) = \Omega(n)$。在 [KS92, Raz92, BJKS04] 中的三个（困难的！）证明在语言和风格上相当不同。但它们都遵循类似的直觉，并且它们都有一个重要的共同特征，我们将在以后使用。那就是它们为 $DISJ$ 选择的有难度的分布支持在要么不相交要么只有一个交集的集合对上！我们明确地陈述这个重要的分布下界。

Theorem 15.3 [KS92, Raz92, BJKS04]. *There is a distribution*[7] μ *on* $\{0,1\}^n \times \{0,1\}^n$*, supported on pairs of sequences* x, y *with at most one coordinate that is 1 in both, such that* $C_\mu(DISJ) = \Omega(n)$.

虽然我将分布通信复杂度作为一种证明概率下界的方法，但它在自身当然是有趣的，因为确实存在一些自然情况下输入分布是给定或已知的。无论如何，证明分布下界，尽管我们现在可以假设

[5] Please check this at least for the uniform distribution.

[6] In a simple-to-guess way: Pick uniformly at random an index $i \in [n]$, and sequences $z \in \{0,1\}^i, w, w' \in \{0,1\}^{n-i}$, and set $x = zw$, $y = zw'$.

[7] We describe the distribution μ chosen in [BJKS04]. Pick pairs of bits (x_i, y_i) uniformly and independently at random from the set $\{(0,0),(0,1),(1,0)\}$. With probability $\frac{1}{2}$, give the players the resulting x, y, respectively. With probability $\frac{1}{2}$, give the players these inputs, but flip the ith coordinate in both to 1, for a randomly chosen $i \in [n]$.

协议是确定的，通常是困难的！有许多技术，例如 *discrepancy bounds*、*corruption bounds* 以及这些技术的平滑和相对变体。我们在此不讨论它们，当我们在第15.3.1节讨论信息复杂性时，我们将讨论一个下界想法，通过 *direct sum*。

证明通信模型下界和理解它们相对能力的一般任务，最近由于一种称为 *lifting* 的技术而取得了巨大进展。它实际上允许将许多较弱模型（如决策树深度或多项式度）的下界提升到通信复杂度的下界。例如，参见 [GPW15, GPW17] 及其参考文献以了解更多关于此技术和其后果的信息。

15.2 Applications

我们现在描述了许多模型中信息成为瓶颈的情况，以及证明这些限制所需的基本通信复杂度模型的变体。请注意，其中一些以及许多其他内容在Roughgarden [Rou16] 精美撰写的专著中有更详细的描述。

15.2.1 VLSI time-area trade-offs

VLSI代表超大规模集成电路。这种基于半导体的技术，在20世纪70年代开发，至今仍主导着集成电路的制造，如运行在手机和笔记本电脑中的微处理器芯片。如今，我们可以在一枚邮票大小的微处理器上集成数十亿个晶体管，这在早期是完全不可能的。但无论是那时还是现在，找到利用芯片面积的最佳方法至关重要.[8] 通信复杂性的最早应用（实际上，在模型完全形式化之前）是表明，芯片计算函数的面积与其计算时间之间存在固有的权衡。这一重要联系是汤普森的博士论文[Tho79, Tho80]，后来在许多其他工作中得到了扩展。

让我们指定计算模型。为了从讨论中消除技术因素，我们假设芯片的计算元素位于单位长度的网格上，并且发送一个比特到相邻的计算元素需要单位时间。每个计算元素可以存储一定数量的比特，并且在单位时间内可以计算其内存中的任意函数并向其任意邻居发送一个比特。输入比特的初始放置是任意的，输出比特的放置也是任意的。

通信复杂度度量函数 g 我们需要的表示为 $C(g)$ (，有时称为 *arbitrary partition* 的通信复杂度 g)，其定义如下。假设 g：$\{0,1\}^{2n} \to \{0,1\}$ 是一个 $2n$ 比特上的函数。每个大小为 n 的子集 $S \subseteq [2n]$ 自然定义了一个通信函数 f_S：$\{0,1\}^n \times \{0,1\}^n \to \{0,1\}$，通过将输入 S 给 Alice，将那些在 S 之外的输入给 Bob，并要求他们计算 g。现在 $C(g)$ 被定义为所有此类划分 S 上的通信复杂度 $D(f_S)$ 的最小值。

Theorem 15.4 [Tho79]。*Every function g computed by a chip of area A in time T satisfies $AT^2 \geq (C(g))^2$.*

证明很简单。一个几何论证表明，在任何面积为 A 的芯片中，以及其大小为 $2n$ 的任何计算元素子集中，都必须存在一种方法，可以沿着网格线将芯片切割成两部分，使得切割的长度最多为 $\sqrt{A}$，并且每个部分包含 n 个给定子集中的元素。显然，如果初始输入位位于此子集中，$C(g)$ 位必须穿过切割，因此 $\sqrt{A}T \geq C(g)$。

汤普森已经使用这个论点来证明傅里叶变换的二次 AT^2 下界，这是一个多输出函数。然而，很容易看出存在简单的函数 g，其 $C(g) = \Omega(n)$（即可能的最大值），从而为这类函数提供 $AT^2 \geq n^2$ 下界。换句话说，一个产品过低的芯片给出的是成本过低的通信协议！

[8] This is a general comment about technology, which explains the (unreasonably short) shelf life of any hi-tech product: As its speed and storage increase, so does the desire to apply it to larger problems (larger input data, finer resolution, etc.).

[9] Obtaining such functions g (that are hard for any partition) from any communication function f (that is hard for a fixed partition) is done simply by encoding the partition S into g's input. This only doubles the input length.

15.2.2 Time-space trade-offs

如您所记得，我们仍然没有关于计算显式函数所需时间和空间的非平凡下界。与前文第15.2.1节类似，我们试图证明这两种资源不能同时非常小。现在，我们证明了一个关于这种性质的结果，针对一个称为 *oblivious branching program (OBP)*的模型。直观上，OBP以*fixed*的顺序访问输入的位（可能多次），与其值无关。更精确地说，空间为S和时间T的OBP计算函数h：$\{0,1\}^n \to \{0,1\}$，如果存在一个序列$\sigma \in [n]^t$和一个只读空间Turing机，该机器在输入$x \in \{0,1\}^n$时读取输入位$x_{\sigma(1)}, x_{\sigma(2)}, \ldots, x_{\sigma(t)}$并输出值$h(x)$。

OBP的最小资源是常量空间和线性时间。Alon和Maass [AM88] 的以下结果证明了它们不能同时用于Majority函数，特别是对于常量空间，时间$\Omega(n \log n)$是必要的。

Theorem 15.5 [AM88]. *For every n, any space S, and time T, OBP computing the Majority function on n bits satisfies $ST \geq \Omega(n \log n)$.*

我们注意到令人惊讶的定理14.11，我们在第14.3节中描述了它，[10]表明多数确实有一个具有常数空间和多项式时间的OBP，实际上，比n^5还要小。上下界之间仍然存在很大的差距，获得这个基本函数复杂性的更紧界限将非常有趣。

对于比多数函数稍微复杂一点的自然函数，同一篇论文[AM88]证明了更好的时间-空间权衡。为了简单起见，我们定义这个函数在3个字母的字母表上，尽管它可以被布尔化。令$Palindrome : \{0,1,*\}^n \to \{0,1\}$为该函数，当且仅当输入的0,1模式（忽略*）是回文时，该函数值为1。

Theorem 15.6 [AM88]. *For every n, any OBP computing the Palindrome function on inputs of length n in space S and time T must satisfy $T = \Omega(n \log(n/S))$. In particular, if $S = o(n)$ then T is super-linear .*

证明这两个定理都使用了相同的“拉姆齐理论”引理（本身很有趣，并有其他应用），该引理可以将一个困难的通信复杂性问题嵌入到计算中。直观上，在$[n]$上的每个足够短的序列都必须有两个大子集，它们的出现交替次数很少。在这里，序列的长度将是时间 T，假设wlog是n的倍数。

Lemma 15.7 [AM88]. *For every n, k, for every sequence $\sigma \in [n]^{nk}$ there exist two disjoint subsets $A, B \subseteq [n]$ such that $|A| = |B| = n' = n/2^{8k}$ and σ contains no k-long subsequence alternating between elements of A and B.*

让我们看看一个空间 S OBP（对于一个函数）如何产生一个低通信协议（对于相关函数），并使用通信下界来推导 OBP 下界。观察如果将 A 中的输入位给 Alice，将 B 中的输入位给 Bob，任何将剩余位固定为常数的操作都定义了一个通信复杂度至多为 Sk（的函数。每个交替都可以通过从一个玩家发送 S 位到另一个玩家来模拟。定理 15.5 和 15.6 现在很容易推导出来。对于回文函数，只需将 A, B 之外的输入设置为 *，从而在 n' 位上得到等价函数（其通信复杂度为 n'）。对于多数函数，将 A, B 之外的位设置为 0 和 1 的数量相等，从而在 $\log n'$ 位上得到大于等于函数，其通信复杂度为 $\log n'$。

记住，对于 *general* 图灵机（与这里考虑的较弱 *oblivious* 图灵机相反），时空权衡后来使用完全不同的方法得到证明——参见定理14.9。

15.2.3 Formula lower bounds

我们现在展示如何使用通信复杂度论证来证明布尔公式的尺寸下界。这种联系是在Karchmer和Wigderson的论文[KW90]中发现的。

[10] Where OBPs are viewed as 2-way finite automata with advice.

基本定义和关于布尔公式的背景信息在5.2.2节中给出，但在此只需回顾我们处理的是基于标准逻辑连接词$\{\wedge, \vee, \neg\}$ (的公式，例如$(x \vee \bar{y}) \wedge z$)，并且一个*monotone*公式不能使用否定（它只能使用$\{\wedge, \vee\}$）。此外，Spira [Spi71] 的一个基本结果表明，公式和单调公式总是可以“平衡”的，使得它们的深度与其大小成对数关系。

以下两个（本质上紧）的下界通过上述连接得出，对于图上的连通性和完美匹配函数（当然，这些是单调函数）。在两者中，n表示输入图的顶点数。

Theorem 15.8 [KW90]. *Every monotone formula for testing whether a graph is connected requires size* $n^{\Omega(\log n)}$.

Theorem 15.9 [RW92]. *Every monotone formula for testing whether a graph has a perfect matching requires size* $2^{\Omega(n)}$.

为了解释与通信复杂度的联系，我们需要将通信问题的概念从计算函数扩展到计算关系。

对于有限集合 X, Y, Z，关系 $F \subseteq X \times Y \times Z$ 定义了一个通信问题，其中 Alice 获得一些 $x \in X$ 作为输入，Bob 获得一些 $y \in Y$，他们应该计算一些满足 $(x, y, z) \in F$ 的 $z \in Z$。我们只考虑对于每个输入对 (x, y) 至少有一个合法答案 z 的关系。

确定性通信复杂度 $D(F)$ 和概率通信复杂度 $R(F)$ 的关系 F 定义方式与函数定义方式相同。

连接公式复杂度与通信复杂度的关键思想是将通信关系 F_g （（有时称为“KW游戏”））与每个布尔函数 g 相关联。此外，如果 g 是单调的，则可以将其分配给另一个（更难）的通信关系，表示为 F_g^{mon}。我们现在定义这两个。

假设 $g : \{0,1\}^m \to \{0,1\}$。对于这两个关系，$F_g$ 和 F_g^{mon}，设置 $X = g^{-1}(0)$，$Y = g^{-1}(1)$，和 $Z = [m]$。换句话说，在两者中，Alice 接收 g 的 0 输入，Bob 接收 g 的 1 输入，他们应该计算某个变量的输入坐标（我们将用 i 而不是 z 表示）。唯一的区别是哪些索引是合法的答案：

- $(x, y, i) \in F_g$ iff $x_i \neq y_i$.
- $(x, y, i) \in F_g^{mon}$ iff $x_i < y_i$.

注意，在这两个关系中，对于每个输入对（x, y），总至少有一个合法答案。由于x和y总是具有不同的g值，它们必须不同，因此它们至少在一个坐标上输入不一致。此外，如果g是单调的，很容易看出至少在一个坐标上，差异将按照指定的顺序出现。

让我们用 $d(g)$ 表示函数 g 的最浅公式 *depth*。同样，我们用 $d^{mon}(g)$ 表示单调函数 g 的最浅单调公式的深度。主要联系简单地陈述 g 的深度和 F_g 的通信复杂度始终相等，单调情况下也是如此。

Theorem 15.10 [KW90]。

- *For every function* g, $d(g) = D(F_g)$.
- *For every monotone function* g, $d^{mon}(g) = D(F_g^{mon})$.

这个定理有一个简单的归纳证明，留给读者。简而言之，g 的一个公式和 F_g 的一个通信协议是同一对象的两种观点。两者都由二叉树描述，关键概念上的差异（在某些情况下使得证明协议的下界更容易）在于，在公式中，计算被视为从叶子开始并传播到根；在通信协议中，计算被视为从根开始并结束于一个叶子。

与这种连接相关，回顾一个简单事实[11]，即一个公式的深度（在常数范围内）是大小的对数，定理15.8和15.9可从定理15.11和15.12得出，其中证明了相关关系上的通信下界。让Conn和PM分别表示图上的连通性和完美匹配函数。注意，对于有n个顶点的图，这些函数的输入大小是$m = n^2$。虽然确定性通信复杂度的下界足够，但我们实际上知道它们甚至对概率通信也成立，这是一个很快就会派上用场的实际情况。

Theorem 15.11 [KW90, RW89]. $R(F^{mon}_{Conn}) = \Omega((\log n)^2)$.

Theorem 15.12 [RW92]. $R(F^{mon}_{PM}) = \Omega(n)$.

虽然我不会证明这些定理，但我想要指出一个重要观点。鉴于公式深度与通信复杂度之间的等价性简单，人们可能会问通信复杂度观点带来了什么优势。两种情况下的答案不同（见下文），但两者以及其他证明的共同之处仅仅在于我们拥有通信复杂度中的一系列工具和结果，其描述性语言完美地适用于两个通信参与者，但如果翻译成公式则完全晦涩难懂。这些工具专门以两种不同的方式为上述定理的证明服务。

对于图连通性的第一个下界，关键地利用了这样一个基本事实：在通信复杂度框架中，我们有 two 个输入，而公式中只有 one。证明结合自顶向下的归纳和随机限制论证，以证明玩家甚至不能在各自输入的“足够大”子集上解决这个问题。

第二个完美匹配的下界通过直接减少到第15.1节中讨论的集合不相交函数 $DISJ$ 得到证明。由于这种减少是概率性的，因此需要一个关于该问题的随机通信复杂度的下界，这在定理15.3中得到，幸运的是，在本文之前不久被发现。很难想象在公式的背景下使用这样的结果，因为在这个不相交下界是无意义的。

许多其他关于这些和其他单调计算模型的下界已经被使用通信复杂度方法证明，尤其是在所谓的*lifting*方法最近取得进展之后。读者可以从[RPRC16, PR17]中了解更多信息。

目前我们只看到了单调的下界。非单调的呢？观察给定单调函数的单调与一般KW关系之间的显著相似性，似乎对单调关系的下界进行微小修改后，可以“修复”以提供非单调下界。为了说明这两个是相当不同的，这里有一个它们不同的主要特征。对于每个在m位上的函数g，F_g的随机通信复杂度至多为$2 \log m$ (，这来自于定理15.2的第一项)，而对于某些g，如上面的完美匹配函数，我们看到了一个$\sqrt{m}$下界。因此，特别是，为了证明非单调下界，不能使用分布性论证（其中输入是随机选择的），也不能使用概率归约到其他问题。

可能很难，但这里有一个具体的挑战。证明如果Alice有一个n-位素数，而Bob有一个n-位合数，那么他们没有确定性的$O(\log n)$位通信协议来找到一个坐标，使得他们的输入不同。显然，证明这一点意味着测试素性没有多项式大小的布尔公式。

15.2.4 Proof complexity

尽管这个应用程序主要是自包含的，但读者可能想回顾第6章关于证明复杂性，特别是第6.3.2节关于切割平面证明系统和第6.4节关于可行插值方法。

证明复杂性的目的是表明自然命题重言式在自然证明系统中需要长证明。为了简单起见，我们在这里定义了一切*semantically*，尽管在证明系统中语法是至关重要的。考虑n变量上的布尔函数。我们说两个布尔函数f, g蕴含

[11] Via the balancing of binary trees.

一个第三者 h (由 $f, g \vdash h$) 表示，如果每个 x 都满足 f, g 也满足 h；具体来说，$f(x) = 1$ 和 $g(x) = 1$ 意味着 $h(x) = 1$。

现在令 F 是 n 变量上的任何布尔函数族，其中包含常数 0 函数（F 大约对应于当前的手头证明系统）。一个子集 $A \subseteq F$ 是一个 *contradiction*，如果不存在任何输入 x 满足所有函数在 A 中。对于 A 的树形 *refutation* 是一个二叉树，其节点由满足以下条件的 F 中的函数标记：

- 每个叶子都由一个来自 A 的函数标记。
- 根由常函数0标记。
- 如果 h 标记一个节点，并且 f, g 是其子节点的标签，那么 $f, g \vdash h$。

请注意，任何这样的反驳确实是一个逻辑证明，即 A 是一个矛盾。我们寻求自然的矛盾 A (，通常具有多（n）“简单”函数），其中每个树形反驳的 *size* (，例如叶子的数量) 都很大（希望是 n 上的超多项式或指数）。让我们描述一种证明反驳的下界的一般方法，这种方法由 Pudlák 和 Buss [PB94] 提出。它的价值在于，对于某些家族 F，例如我们很快将关注的家族，深度的下界 d 将意味着大小 exp(d) 的下界——这是我们正在寻找的结果。

一个 F-查询只是对任何函数 $f \in F$ 在任何输入上的评估。考虑一个被给予输入 x 并希望找到一个函数 $a \in A$ 使得 $a(x) = 0$（如果 A 是矛盾，则必须有一个）。设 $Q_F(A)$ 为解决此搜索问题所需的最小查询数，在最坏情况输入 x 上。显然，$Q_F(A)$ 的下界给出了任何树形反证法深度的下界.[12]

我们现在准备解释与通信复杂度的联系。上述单人游戏描述了通常所说的*decision tree*，尽管在这里允许查询的集合是非标准的[13]，并且它计算的是一个关系而不是一个函数。现在假设输入x的位实际上被分给了两个玩家，Alice和Bob（例如$x = y, z$，Alice接收y，Bob接收z）。进一步假设在这种输入划分下，*all*函数 $f \in F$具有低通信复杂度（例如$D(f) \leq c$或$R(f) \leq c$）。那么现在一个深度d的A反证现在转化为一个$2dc$ (确定性或概率性的通信协议，分别对应于原始搜索问题。换句话说，通信复杂度下界意味着证明复杂度下界，如果上界c很小，这将特别有用。

这个想法被Impagliazzo、Pitassi和Urquhart [IPU94] 用于证明树形切割平面反演的下界，如下所示。首先，通过将家族 F 设置为所有关于输入 x 变量 x_i 的整数系数线性不等式来捕捉切割平面证明。更正式地说，对于形式为 $\sum_i s_i x_i \leq t$ (的每个线性不等式，其中 t, s_i 是整数) 和 [14] $f \in F$，将满足此不等式的每个 x 分配为1，否则为0。应该清楚，任何此类 $f \in F$ 的通信复杂度都简化为函数 GE 的复杂度，即测试两个 n log n 位整数哪个更大。我们在定理15.2的第二项中看到 $R(GE) = O(\log n)$。

接下来，[IPU94] 证明了对于这个切割平面 F，任何树形证明都可以平衡；因此，深度下界意味着大小下界。最后，Impagliazzo 等人需要一个矛盾，这需要高通信复杂度。幸运的是，定理 15.12 的证明已经隐含地包含了这样的矛盾。对于完美匹配问题，上述单调 KW 游戏立即暗示了这样的“完美匹配”矛盾，这可以通过线性不等式轻松表达。我不会在这里明确写出这个矛盾。与定理 15.12 中的 $\Omega(n)$ 概率通信复杂度下界（该下界是从定理 15.3 推导出来的）一起，Impagliazzo 等人推导出所需的指数下界。

[12] Simply put, if we had a refutation of depth d, our player could start at the root (where the 0 function falsifies its input x) and proceed down to a leaf, each time querying the functions labeling the children of the current node, and proceeding to any that falsifies x. This will require $2d$ queries.

[13] The most well-studied case, called a Boolean decision tree, is when queries are simply the coordinate functions x_i.

[14] Which, without loss of generality, are no more than $n \log n$ bits in length; verify this fact!

[15] Alice and Bob can each separately compute the partial sum on their input bits.

Theorem 15.13 [IPU94]. *The "Perfect Matching" contradiction on n-vertex graphs requires cutting-planes tree-like contradictions of size* $\exp(n/\log n)$.

15.2.5 Extension complexity

Yannakakis [Yan91] 的论文中，通信复杂度最巧妙和出乎意料的用途之一来自这篇论文。在这篇论文中，Yannakakis 将其与凸几何联系起来，并展示了如何通过相关通信问题的非确定性通信复杂度捕获每个高维多面体（我将很快定义）的 *ex- tension complexity*。这种理解自那以后在凸组合优化和优化中发挥了重要作用。

但在我告诉你那个故事之前，我必须先告诉你这个故事。每年，ToC社区都会接触到许多声称解决$\mathcal{P}$与$\mathcal{NP}$问题（与数学社区接触到声称解决黎曼猜想的方法类似）的新论文。幸运的是，几乎所有这样的论文都存在明显的错误或误解，因此可以不予考虑，无需投入太多时间。然而，偶尔这样的论文并不容易轻易否定：它们似乎包含真实的思想，被严格地呈现，而且（谁知道呢？）可能包含一个证明。处理这些论文对社区来说是一个非同小可的挑战，因为这些主张极具兴趣。这是关于一次尝试证明$\mathcal{P} = \mathcal{NP}$的故事，由Swart [Swa86] 提出。

想法简单而强大。*traveling salesman problem (TSP)*是$\mathcal{NP}$-完备的。它可以写成线性规划，这是一个几年前被发现属于$\mathcal{P}$ (的问题，如第3.2节末所述)。主要问题是，当将其写成自然线性规划时，给定图 *original variables* (的边具有指数级的线性约束。Swart建议添加*auxiliary variables*，并提出了一个新的线性规划，其中只包含多项式级的约束，并添加了这些变量。他声称新的线性规划也捕捉到了TSP，因此$\mathcal{P} = \mathcal{NP}$。这并不容易，但一些专注的研究人员发现了Swart程序中的某些错误。Swart建议了一个固定的程序，但后来在这个程序中也发现了更多的错误，以此类推。当社区应该放弃时，当其最基本问题的解决方案可能就在触手可及的时候?

Yannakakis的论文[Yan91]旨在证明Swart的方法必然失败*in principle*。为了讨论这篇论文，我们首先必须将这种方法数学化。我们需要一些预备知识。一个*polytope* $P \subseteq \mathbb{R}^n$是有限点集$V \subseteq \mathbb{R}^n$的凸包。多面体P的另一种方便的描述是有限个半空间F的交集，称为*facets*。我们假设V和F都是*minimal*：从其中移除一个元素都会得到一个比P更小的多面体。设$|F| = f$和$|V| = v$。通过P的面描述给出f不等式，简要描述为 *linear program*，$P = \{x\colon\ Ax \le b\}$，其中$x \in \mathbb{R}^n$，$b \in \mathbb{R}^f$和$A$是一个$f \times n$实矩阵。基本决策问题，测试$P$是否为空（以及优化$P$，我们不会讨论），可以在$A$的维度内多项式时间内解决。

以下示例有助于说明复杂性问题。考虑在图 $G(U, E)$ 中找到一个完美匹配[16]的问题。相关的多面体P_G简单地是所有完美匹配的凸包，或者更确切地说，是所有 $\mathbb{R}^E$ 中的 $0-1$ 向量，这些向量是完美匹配的特征向量。该多面体的面由 Edmonds [Edm65a] 确定，他证明了以下线性规划定义了 P_G。变量是对于每条边 $e \in E$ 的 x_e，不等式是:

- $\sum_{e \ni u} x_e \le$ 每个顶点 $u \in U$ 对应 1。
- $\sum_{e \subseteq S} x_e \le (|S| - 1)/2$ 对于 *odd* 中的每个 $S \subseteq U$ 子集 G。

当 G 是二分图时，它没有奇数环，第二个不等式族不再需要，f 变为顶点数，因此多项式时间线性规划算法可以在多项式时间内解决任何二分图中的完美匹配。然而，某些其他图 G 有指数级的奇数环，上述多面体有指数级的面，这种方法不能使用。对于完美匹配问题，Edmonds 跳过了这个问题，并发现了一个非常不同的多项式时间算法 [Edm65b]。然而，当制定标准 $\mathcal{NP}$-完全问题（例如，团，哈密顿图）时，...

[16]Recall that a perfect matching is a subset of the edges that cover every vertex exactly once.

循环，旅行商问题）与线性规划相同，它们都具有指数级的面数，当然，任何问题的多项式时间算法都将意味着 $\mathcal{P} = \mathcal{NP}$。

Extension 多面体理论提出了一种减少面数的一般方法，通过添加一些辅助变量。如果一个多面体 $Q \subseteq \mathbb{R}^{n+n'}$ 被称为 *extension* 的 $P \subseteq \mathbb{R}^n$，当且仅当 P 是 *projection* 的 Q，即 $P = \{x : \exists y, (x, y) \in Q\}$，其中 $y \in \mathbb{R}^{n'}$.。在这里，x 是 P 的原始变量，y 是新的辅助变量。显然，P 为空当且仅当 Q 为空。设 m 为 Q 的面数。因此，如果我们能使 $n' + m$ 远小于原始面数 f，我们就赢了。这种可能性并非空想；$n' + m$ 有时可以比 f 线性小得多。[17] 对于 TSP 多面体，能否像 Swart 尝试的那样做呢?

定义 *extension complexity* 为 P，$e(P)$ 的最小面数 m，在任意扩展 P (的多面体 Q 中，对于这个多面体 Q，将自动得出 $n' = O(m)$，所以我们不必担心它)。Yannakakis 的第一个绝妙想法是对 $e(P)$ 的简单、完整刻画，他的第二个想法是展示非确定性通信复杂度如何意味着（几乎紧）对 $e(P)$ 的下界。一个关键的定义是多面体 P 的 *slack matrix*，我们现在给出。

让 F 和 V 分别是 $P = \{x : Ax \leq b\}$ 的面和顶点。那么 P 的 *slack matrix* S_P 是一个 $F \times V$ 矩阵，其 (i, j) 项仅仅是 $b_i - \langle a_i, v_j \rangle$，即第 j 个顶点到第 i 个面的距离。观察所有 S_P 的项都是非负的是至关重要的！下一个定义是非负矩阵的 *nonnegative rank*。它像矩阵的通常秩一样定义，始终强调非负性。也就是说，对于一个 $f \times v$ 非负矩阵 S，令 $\text{rk}^+(S)$ 是最小的 m，使得 $S = RT$ 对于非负矩阵 R, T 的维度 $f \times m$ 和 $m \times v$ 分别。有了这些定义，我们现在可以陈述特征，有时称为“Yannakakis分解定理”。

Theorem 15.14 [杨91]。*For every polytope P, $e(P) = \text{rk}^+(S_P)$.*

该定理的证明直接来自定义、线性代数以及单个“凸性”事实（称为“法卡斯引理”），即如果一个线性不等式被另一组线性不等式逻辑蕴含，那么实际上它必须是这些不等式的非负线性组合。

关于这次讨论，我们希望证明（希望是指数的）上述一些多面体中$e(P)$的下界。关于函数rk^+可以说什么？首先，与通常的秩不同，非负秩是一个棘手的函数，甚至在布尔矩阵上也是$\mathcal{NP}$-难计算的[Vav09]。显然，通常的秩（在实数上）提供了一个下界，即$\text{rk}^+(S) \geq \text{rk}(S)$对于每个$S$。但是这个界限可能非常弱；例如，存在矩阵$S$，其$\text{rk}(S) = 3$，且$\text{rk}^+(S)$无界[Hru12]。这时，通信复杂度变得很有帮助，而为了使用它，将我们的矩阵变为布尔矩阵将有所帮助。

对于一个非负矩阵 {v*}，令 {v*} 为布尔矩阵，其中我们将每个正项替换为 1（并保留 0）。注意，如果 rk{v*}，那么 {v*} 中的 1 可以被 {v*} 个单色 1-矩形覆盖。具体来说，如果 {v*}，那么 {v*} 个矩形 {v*} 覆盖了 {v*} 中的 1，其中 {v*} 和 {v*} 分别是 {v*} 和 {v*} 的 {v*} 行和 {v*} 列。简而言之，我们只需要一个多面体 {v*}，使得 {v*} 没有小的单色覆盖。

Yannakakis [Yan91] 设置了所有这些，但实际上并不能完全交付成果。他能够展示出Swart的尝试注定失败，利用Swart构造的额外属性——对称性，在这个框架下，他能够获得TSP凸体相关“对称”扩展复杂度的指数下界。但他留下了证明一般扩展复杂度指数下界的未解问题，这将完全排除这种线性规划方法来证明 $\mathcal{P} = \mathcal{NP}$。

这是最终在25年后，在Fiorini等人[FM^+15]的论文中实现的。令K_n表示在n顶点上的完全图。

Theorem 15.15 [FMP^+15]. *The extension complexity of the TSP polytope of K_n is* $\exp(n)$.

我们只概述了证明的高层次思想。而不是研究TSP凸体的松弛矩阵，证明是间接进行的。鉴于TSP是一个$\mathcal{NP}$-完全问题，Fiorini等人。

[17] As an exercise, prove that the convex hull of all *odd* weight n-bit vectors requires $\exp(n)$ facets, and that adding $O(n^2)$ new variables reduces the number of facets to $O(n^2)$.

原因是在上述意义上找到一个 {v*} 显式多面体，其松弛矩阵难以覆盖。一旦找到，标准的$\mathcal{NP}$-完备性归约就能完成这项工作。困难在于找到适当的多面体。他们的巧妙想法是使用所谓的 *cross polytope*。这里我不会描述它，而是描述其松弛矩阵 S 的性质。交叉多面体的面和顶点（即 S 的行和列）可以与 $[n]$ 的子集建立 $1-1$ 一一对应。如果两个集合 *disjoint*，那么 S 的相应项为 1。如果两个集合在单个元素处相交，那么 S 的相应项为 0。其他对呢？我们不在乎！再次回忆定理 15.3 的强下界。不难看出，它意味着（实际上一些证明就是这样证明的）任何 S 中 1 的覆盖都需要 $\exp(n)$ 个矩形。这提供了对 $\mathrm{rk}^+(S)$ 所需的下界，从而证明了该定理。

可以使用相同的思想通过归约来证明与许多其他$\mathcal{NP}$-完备问题相关的多面体扩展复杂性的指数下界。无法以这种方式完成，且仍然引人入胜的是确定本节开头讨论的通用图完美匹配多面体的扩展复杂性。这由Rothvoss [Rot14] 解决。有趣的是，这个下界与定理15.12中的下界有一些相似之处，并且确实，这种联系在 [Hru12] 中得到了形式化，以表明扩展复杂性的下界如何产生公式大小下界。

Theorem 15.16 [Rot14]. *The extension complexity of the Perfect Matching polytope of K_n is* $\exp(n)$.

15.2.6 Pseudo-randomness

此处应用可能看起来与之前提到的所有应用都不同。迄今为止，我们已经看到了从通信复杂度下界推导出的计算下界。在这里，我概述了如何基于通信复杂度构建伪随机生成器。然而，它在精神上与其他应用相当相似，我们将其留给读者来阐述这种相似性（一个提示是回想一下伪随机性意味着困难）。

尼桑著名的空间有界伪随机生成器[Nis92]已在本书中提及两次，分别在8.5节和14.1节，再次作为定理14.4。这个生成器是由14.2节末提到的空间下界所启发，生成器构造和伪随机性的证明都隐式地使用了它。几年后，Impagliazzo、Nisan和Wigderson[INW94]描述了一个不同的伪随机生成器，其伪随机性的证明通过直接将通信复杂性问题作为结论而明确地遵循。为了描述它，我们需要回顾扩展图的概念（见8.7节）。为了适应通信复杂性的框架，我在这里定义了一个*bipartite*扩展图，其所述的扩展性质与8.7节中的性质S_1类似（即在任意两个顶点子集之间，我们找到的边数与具有相同度数的随机图中的边数大致相同）。8.7节中定义和描述的显式构造产生了我们这里需要的二分扩展图。

一个 *bipartite expander* 是一个 $D = 2^d$-正则二部图 $H(A, B;\ E)$ 在大小为 $N = 2^n$ 的两个顶点集 A, B 上，使得对于任意两个子集 $A' \subseteq A$，$B' \subseteq B$，我们有 $||E(A', B')|/dN - |A'||B'|/N^2| \leq \epsilon = 2^{-d/3}$ (其中 $E(A', B')$ 是两个子集之间的边集)。如果存在一个多项式(n, d)时间算法，对于每个顶点 v 和一个索引 $i \in [D]$ 输出 v 在 H 中的第 i 个邻居，则 H 是 *explicit*。

请注意，扩展条件本质上是一个关于1的密度 *rectangles* 在 H 邻接矩阵中相对于它们大小的条件。不出所料，由于通信协议本质上是对矩阵进行矩形划分，这个性质立即转化为以下通信复杂度陈述。我们使用以下符号。对于Alice和Bob的 n-位输入上的通信协议 P，以及输入上的分布 μ，让我们用 $P(\mu)$ 表示当玩家的输入随机根据 μ 选择时，P输出1的概率。

Lemma 15.17. *Let P be any c-bit communication protocol, and H any bipartite expander with $d = 4c$. Then $|P(U) - P(H)| \leq 2\epsilon$, where U is the uniform distribution, and (abusing notation) H is the distribution picking a random edge from H and giving each player one endpoint of this edge.*

[18] Due to this property, [INW94] can generalize the construction and give pseudo-random generators for more general computations.

我不会在这里描述生成器的构造，但请注意，[INW94]中使用了扩展器，其方式与原始的[Nis92]中使用哈希函数的方式相似。对于最终生成器，关键属性是使用分布H而不是U所涉及随机性的减少。从U中采样需要$2n$位，而从H中采样只需要$n+d$位。这种节省通过递归以与[Nis92]中相同的方式累积。

15.3 Interactive information theory and coding theory

通信复杂度领域，除了创建了一个深度和广度丰富的理论，具有许多应用，正如我们所见，它还自然地分支到研究信息论和编码理论领域中的基本问题。这些大型领域主要关注*information transmission*，即一方持有的*message*的*1-way*通信.[19]相比之下，通信复杂度研究*information exchange*，即*2-way*的*adaptive conversations* (通信，例如，双方输入的一些任意函数或关系)。询问这些领域的经典问题，如压缩的可能性和对信道噪声的容忍度，在一般的*interactive*设置中自然变得更加具有挑战性。这项研究已经导致了一些美丽的结果和更多开放的问题，正在成为一个独立领域，位于信息理论和计算复杂性的交汇处。以下是一些亮点，首先关于通信协议的压缩，然后是它们的纠错方案。在这两方面，我解释了单向和双向设置中的相似之处和不同之处。

在开始之前，回忆以下由香农在其开创性的论文[Sha48]中定义的标准信息论量，该论文创建了信息论。设存在同一有限集合上的随机变量A, B, Z。$\mathbb{E}$表示期望。所有对数都是以2为底。随着我们下面讨论这些量，我们将进一步发展对这些量的直觉。

entropy of A，直观上，A的以比特为单位的测量不确定性，表示为 $H(A)$，并由以下定义：

$$H(A) = -\sum_a \Pr[A=a] \log \Pr[A=a].$$

conditional entropy of A given B (直观上，在 A 揭示后) 剩余多少不确定性，用 $H(A|B)$ 表示并定义为

$$H(A|B) = \mathbb{E}_b[H(A|B=b)].$$

互信息 {v*} 的直观理解，知道 B 有多少可以揭示 A 以及反之亦然) 表示为 $I(A;\ B)$ 定义如下

$$I(A;B) = H(A) - H(A|B) = H(B) - H(B|A) = H(A) + H(B) - H(A,B).$$

所有这些概念在给定第三个变量 Z 的条件下是有意义的。特别是，*conditional mutual information* 对我们特别相关，并由以下定义：

$$I(A;B|Z) = H(A|Z) - H(A|B,Z).$$

15.3.1 Information complexity, protocol compression, and direct sum

中心复杂性度量对于通信任务来说，除了通信复杂性本身，还有 *infor- mation complexity*，由 Chakrabarti 等人在论文 [CSWY01] 中引入，并在 [BJKS04] 和随后 [BBCR13] 中发展，我们在这里将使用其定义。有几种（相关）方式来激发信息复杂性。一种是将 Shannon 的单程通信源编码定理推广到通信复杂性的双程设置。另一种是试图找到一个比通信复杂性更“连续的参数”，因为通信复杂性始终是整数。第三种是从 *amortized* 复杂性的角度来看，当试图一次性解决许多实例的问题时。我们将看到所有这些，但让

[19] Needless to say, this focus is plenty broad as is. The number of theoretical models and practical settings under which such a basic question is studied is vast and has occupied thousands of researchers for decades and many more in industry; the results are present in the technology we all regularly use.

我们试图通过信息论手段来证明（分布式的）通信复杂度的下界，从而激发了这个定义。

首先，让我们考虑一个任意的通信协议 π 当应用于一对按照分布 μ 分布的输入。我们滥用符号，并让 (X, Y) 表示输入随机变量（联合分布按照 μ）。当在这样随机的输入上执行时，π 定义了另一个随机变量 $\Pi = \pi(X, Y)$，Alice 和 Bob 之间的 *transcript* 或 *conversation*。根据互信息的直观意义，很明显，Alice 持有 X，从他们的对话 Π 中至少学习到关于 Bob 输入 Y 的 $I(Y;\ \Pi|X)$ 比特信息；Bob 平均学习到 $I(X;\ \Pi|Y)$ 关于 X 的信息。因此，玩家必须交换至少与这两个数量之和一样多的比特数。让我们将这一点形式化。

对于任何协议 π 和分布 μ，定义 *information complexity*[20] $I_\mu(\pi) = I(Y;\ \Pi|X)+I(X;\ \Pi|Y)$。另外，设 $C_\mu(\pi) = \mathbb{E}_\mu|\pi(X, Y)|$ 为通信的期望长度。那么上述论点可以简单地表述为证明期望长度 $C_\mu(\pi)$ 的基本下界。

Theorem 15.18 [CSWY01, BBCR13]. *For every π, μ, $C_\mu(\pi) \geq I_\mu(\pi)$.*

显然，这个下界也适用于协议集合。例如，设$\Pi(f, \mu, \epsilon)$为所有以至少$1 - \epsilon$的概率在输入分布μ上计算f的确定性协议集合。那么，这个任务的分布通信复杂度$C_{\mu,\epsilon}(f)$已经被定义为当π遍历这个集合$\Pi(f, \mu, \epsilon)$时的$C_\mu(\pi)$的最小值。我们类似地定义$I_{\mu,\epsilon}(f)$，即这个任务的信息复杂度，为该集合中所有π的*infimum*[21] $I_\mu(\pi)$。然后我们得到以下定理。

Theorem 15.19 [CSWY01, BBCR13]. *For every f, μ, ϵ, $C_{\mu,\epsilon}(f) \geq I_{\mu,\epsilon}(f)$.*

从现在起，我们将把 $\epsilon > 0$ 视为微小（实际上，在其他参数中可以忽略不计）并在我们的符号中忽略它。进一步，让我们固定一个分布 μ，并因此也将其移除。最后，将 f 视为一个比仅仅计算一个函数更一般化的任务是有用的，并假设它可以是对协议输出（例如，一个关系，或者每个玩家的不同函数，甚至分布）的任何要求。到目前为止讨论的以及以后将要讨论的所有内容都适用于这种普遍性。我们将致力于理解基本问题：这个下界 $C(f) \geq I(f)$ 有多好（或有多紧）？我们还将对现在定义的这些量的 *amortized* 版本提出相同的问题。

自香农的论文[Sha48]以来，信息论的一个主要焦点是在极限情况下，对独立输入执行相同任务的成本。在计算机科学文献中，这被称为*direct sum*问题，它对于*any*计算模型和资源自然产生。用f^k表示Alice得到$(X_1, X_2, \ldots, X_k)$，Bob得到$(Y_1, Y_2, \ldots, Y_k)$的任务，其中(X_i, Y_i)独立，每个都按照μ分布，各方必须以至少$1 - \epsilon$ (的概率成功执行$f(X_i, Y_i)$。记住，他们的消息可以使用他们的全部输入)。让我们用$\bar{C}(f) = \lim_{k\to\infty} \frac{1}{k} C(f^k)$表示，同样地，$\bar{I}(f) = \lim_{k\to\infty} \frac{1}{k} I(f^k)$。论文[CSWY01]考察了基本直接和问题以及明显下界的紧密度$C \geq \bar{C}$。它引入信息复杂度主要是因为，对于这个度量，两者是相同的！[22]

Theorem 15.20 [CSWY01, BBCR13, BR14]. *For every f, μ, ϵ, $I = \bar{I}$.*

为了研究两个基本问题（即，下界 $C \geq I$ 和 $C \geq \bar{C}$ 的接近程度），让我们从经典信息理论中研究的早期通信任务中寻求直觉。请注意，所有这些都可以视为 *compression* 结果；总的来说，I 自然地捕捉了问题的 *information content*，我们试图使通信 C 尽可能接近它。自然地，一些原始结果并没有用这种语言表述。我们还将利用事后诸葛亮来添加或简化。

首先考虑Bob没有输入的情况（因此可以将y视为空或常数）。要计算的函数是Alice输入上的恒等函数：$id_A(x) = x$。因此，Alice只想将她的输入（从某个分布X中采样）发送给Bob。因为Bob学习x (，而Alice无法学习到任何信息

[20] This is often called *internal* information complexity, to distinguish it from a related measure, *external* information complexity, capturing the amount of information a protocol reveals to an external observer about the players' inputs.

[21] Being a continuous measure, there are tasks (even simple ones, like taking the AND of two bits) that have an infinite sequence of protocols with better and better information complexity.

[22] Proving this to within a factor of 2 is easy and sufficed for the motivation; the exact result follows from [BR14], which we shall soon discuss.

新），每个协议 π 必须满足 $I(id_A) = I(X; \Pi) = H(X)$。Shannon [Sha48] 证明了以下下界，而优雅的 Huffman 编码 [Huf52] 给出了几乎匹配的上界。[23] 平摊界限是 Shannon 著名的 *source coding theorem*。

Theorem 15.21 [Sha48, Huf52]。

- $H(X) = I(id_A)$.
- $I(id_A) \leq C(id_A) \leq I(id_A) + 1$.
- $I(id_A) = \bar{C}(id_A)$.

现在重新考虑这个问题：Alice必须将她的输入x传输给Bob，但现在是在Bob确实有一个输入y的一般情况下，并且这对输入被分配为(X, Y)。简而言之，他们正在计算$id_A(x, y) = x$。使用与之前相同的推理，任何协议都必须让Bob完全了解Alice的输入。因此，每个协议π都满足$H(X|Y, \Pi) = 0$，因此$I(X; \Pi|Y) = H(X|Y)$。当然，一些协议可能会给Alice关于Bob输入的信息。无论如何，我们有$I(id_A) \geq H(X|Y)$。摊销情况在Slepian和Wolf [SW73] 的著名论文中得到了研究，一次性情况由Orlitzky [Orl92] 研究。请注意，与只考虑单向通信的Slepian-Wolf不同，Orlitzky考虑了双向通信，但表明这对此问题没有帮助（因此Alice学不到任何东西）。[24]

Theorem 15.22 [SW73, Orl92]。

- $H(X|Y) = I(id_A)$.
- $I(id_A) \leq C(id_A) \leq (1 + o(1))I(id_A)$.
- $I(id_A) = \bar{C}(id_A)$.

一个最终例子是问题 $id(x, y) = (x, y)$，即两个玩家必须交换他们的输入。这被El Gamal和Orlitsky [EGO84] 研究过。他们有几个结果，我们略微夸大并非正式地总结，关于一次任务.[25]当然，摊销情况是从Slepian-Wolf定理15.22得出的，分别应用于 id_A 和 id_B。注意，如上所述，对于任何 id 协议，必须满足 $H(X|Y, \Pi) = H(Y|X, \Pi = 0)$，因此 $I(id) \geq H(X|Y) + H(Y|X)$。

Theorem 15.23 [SW73, Orl92]。

- $H(X|Y) + H(Y|X) = I(id)$.
- $I(id) \leq C(id) \leq (1 + o(1))I(id)$ *for "almost" all distributions* μ.
- $I(id) = \bar{C}(id)$.

让我们尝试从这些例子中推广。首先，考虑平均情况，这似乎更简洁。在所有例子中，我们都有 $I(f) = \bar{C}(f)$。确实，这些方程通常被视为熵和条件熵的 *operational* 定义，为它们上述抽象数学定义提供了实际动机。其他任务 f 呢？Braverman 和 Rao 的重要定理 [BR14] 表明我们总是有等式，从而精确地描述了平均通信复杂度。

Theorem 15.24 [BR14]. *For every task* f, $I(f) = \bar{C}(f)$.

让我谈谈这个证明。正如我们在其他章节中看到的，选择正确的“通用”或“完整”任务 f 通过归纳来完成工作，这正是他们所做的事情。这个任务被抽象为在 *KL-divergence* 距离度量上一对接近分布的联合采样问题。他们的

[23] I am cheating a bit here, because for Huffman coding, C really denotes "average-case" as opposed to the "worst-case" communication complexity that we are using throughout.

[24] A good example to consider is when Bob's y is a pair of n-bit files, (z_0, z_1), and Alice has one of them. If Bob talks first, it is easy to solve the problem with $\log n + 1$ communication. Can you do the same when only Alice talks? Hint: hashing!

[25] A good example to consider here is that x is a random n-bit file, and y is another random file differing from it in some random set of coordinates of size at most s.

通信高效的协议解决这个问题（其中 *does* 需要双向通信）可以视为对定理15.22（不这样）的推广和加强[26]。

我们现在回到“一次性”压缩问题，即一般而言$C \geq I$有多紧。第一个直接解决这个一般问题的论文；发展了协议压缩的一般技术；并且特别地，证明了迄今为止最好的通用压缩结果是Barak等人[BBCR13]提出的。他们专注于压缩给定的协议，这当然意味着对于一般任务的压缩结果。这可以非正式地定义为以下内容。固定一个任意的输入分布[27]和一个协议π。我们正在寻找另一个协议π'，它（以概率$1-\epsilon$，如通常情况）将计算会话记录$\pi(X,Y)$。希望π'将被压缩（即使用比π更少的通信），可能接近或达到其信息复杂性（这是可能的最小值）。另一个期望的特性是Alice和Bob在π'中的计算基本上与π中的计算一样高效。这个效率要求适用于所有已知的模拟。

Theorem 15.25 [BBCR13]. *For every protocol π, there exists another protocol π' such that $C(\pi') \leq \sqrt{C(\pi) \cdot I(\pi)}(\log C(\pi))^{O(1)}$.*

此结果表明，可以将任何协议的通信大致压缩到其原始 C 和 I 的几何平均。这个结果有多好？可以构造出人工协议（甚至存在自然协议），其中 C 远大于 I，甚至无限大。在这种情况下，通信减少到平方根，但并不接近信息复杂性。因此，如果能得到仅依赖于 I 的压缩结果就很好了。Braverman [Bra15] 发现了这样的压缩。

Theorem 15.26 [Bra15]. *For every protocol π, there exists another protocol π' such that $C(\pi') \leq \exp(I(\pi))$.*

是否存在更好的压缩方法？Ganor、Kol和Raz [GKR14, GKR15] 证明了定理15.26无法改进，不仅对于单个协议，而且对于计算某些任务也是如此。

Theorem 15.27 [GKR14,GKR15]. *For every integer m, there exists a Boolean function f such that $I(f) = O(m)$ but $C(f) \geq 2^m$.*

我们现在可以根据摊销通信复杂度的特征$\bar{C} = I$来解释这些结果。这些代表了在通信复杂度中非常古老的问题 *direct sum* 上的重大进展。当这个问题在 1980 年代被提出时，人们认为解决任何问题 k 次大约需要通信成本的 k 倍增加：[28] $\bar{C}(f) = \Omega(C(f))$。然而，几十年来没有找到非平凡的上下界。信息复杂度方法意味着以下内容。结合定理 15.26 和 15.27，我们有一个紧界。

Theorem 15.28 [BR14, Bra15, GKR15]。

- *For every communication task f, $\bar{C}(f) \geq \Omega(\log C(f))$.*
- *For some Boolean functions f, $\bar{C}(f) \leq O(\text{对数}\ C(f))$.*

此外，定理15.25表明，在摊销过程中必须产生一些乘法成本，这在我们明确考虑解决的实例数量（并抑制对数因子）时表述得更好。

Theorem 15.29 [BBCR13, BR14]. *For every task f and integer k,*
$C(f^k) \geq \Omega(\sqrt{k}C(f))$.

15.3.2 Error correction of interactive communication

我们现在转向处理通信通道上的 *noise*，以及如何在它的情况下使通信可靠。我们专注于最典型的噪声模型，即 *bit-flips*。使用噪声的斗争主要思想是

[26] Especially with respect to the convergence rate to the limit.

[27] We survey here only results regarding general input distributions μ. Much more is known for restricted families of distributions, in particular, when X and Y are independent.

[28] Indeed, the fact that this bound holds for the monotone KW-relations mentioned above was key for proving some super-polynomial lower bounds for monotone formulas in [KRW95].

error-correcting codes 在香农[Sha48]和汉明[Ham50]的两篇论文中介绍了。香农研究了*random errors*，汉明研究了*adversarial errors*；我们将讨论这两者，首先是在单向通信中，然后是在交互式设置中。本节中总结的材料的优秀详细调查是[Gel15]。

假设Alice想向Bob发送一个n-比特消息x。然而，假设他们之间的通信通道中每个比特发送时都可能被独立翻转，翻转概率为$\leq p$。参数p是每个比特的最大“噪声率”。纠错码的思想是发送一个*encoding*的x，它具有一些冗余来抵消噪声。形式上，纠错码是一个函数C：$\{0,1\}^n \to \{0,1\}^m$。捕捉冗余C的比率n/m被称为码C的*rate*，表示为$R(C)$。

代码 C 可以容忍 *adversarial* 噪声 p，如果对于每个 n-位消息 x，以及 m-位序列 z，它与 $C(x)$ 至多在 pm 个坐标上不同，原始消息 x 可以唯一地 *decoded* 从 z 中恢复。正如 Hamming 指出的，只有在每个不同消息 $x \neq x'$ 的 *Hamming distance* $d_H(C(x), C(x')) > 2pm$ 的情况下才可能。请注意，这迫使 $p < 1/4$。

代码 C 可以容忍 *random* 噪声 p，如果对于每个 n-位消息 x，如果我们独立地以概率 $\leq p$ 翻转 $C(x)$ 的每个位以获得 z，则消息 x 几乎肯定可以从 z 中唯一解码。[29] 注意，这迫使 $p < 1/2$。香农确定了每个噪声率 p 所需的冗余。实际上，他在 *probabilistic method* 的最早应用之一中证明了，该速率的 *random code* 将会这样做。令 $H(p)$ 表示二元熵函数。

Theorem 15.30 [Sha48]. *For every $p < 1/2$, and every $R > 1 - H(p)$, there is a code C of rate $R(C) = R$ that tolerates random noise p. Moreover, no code C of rate $R(C) < 1 - H(p)$ tolerates random noise p.*

在以下论文中，Gilbert [Gil52] 和 Varshamov [Var57] 证明了对于对抗性错误，也可以实现恒定速率，尽管所需的精确速率仍然是一个未解问题。

Theorem 15.31 [吉尔52，瓦64]. *For every $p < 1/4$, there is a code C of rate $R(C) > 1 - H(2p)$ that tolerates adversarial noise p.*

特别地，一个固定大小的膨胀 $m = O(n)$ 足够冗余以容忍两种模型中可能的最大噪声水平。当然，这对于实践来说几乎是不够的，因为上述代码没有明确给出，它们所提出的编码和解码过程效率非常低。经过长时间的研究，最终在两种模型中得到了具有极高效编码和解码算法的显式代码。主要成就包括Spielman的[Spi95]具有线性时间编码和解码的常速率代码，以及Ar kan的[Ar 09] *polar codes*，它们在对抗噪声设置中实现了香农界限，并具有近线性时间的编码和解码。简而言之，尽管关于这些和其他参数仍有许多技术问题尚未解决，但保护单向通信免受错误的基本问题已经得到了很好的理解。

现在假设我们将问题从Alice通过那个嘈杂的信道向Bob发送一个 n-比特 *message* 改变为Alice和Bob有一个 n-比特 *conversation*。这些任务之间的主要区别是，对话是 *adaptive* 的。Bob对Alice的第一个比特的响应可能取决于其值。随着对话的进行，这种依赖性会指数级发散。有许多例子说明了即使是微小的噪声对自适应对话的破坏性影响。例如，想象一下“20个问题”（或更一般地，二分搜索）的游戏，第一个答案被翻转；搜索完全在错误的空间部分进行。同样，假设双方玩一盘棋，Alice的第一步，比如说“e4”，在Bob这边被接收为“d4”。接下来的游戏将毫无意义。

因此，假设我们想要通过纠错码来防止噪声，就像在单向情况中那样。这两个设置之间的主要区别在于，在单向情况中，整个消息都在那里，由一方持有，可以对其进行编码 *at the start*。但现在鲍勃和爱丽丝都不知道对话内容，因为它正在演变。他们能做的最好的事情是编码他们发送的每个新比特，可能包括到目前为止的对话历史。但这看起来可能价值不大，因为所有经典纠错码中的一个关键元素是每个输出比特在 $C(x)$ 上的依赖性，它依赖于来自 x (的大量比特，实际上是一个常数比例)。

Shannon的此交互式设置中错误纠正的类似工作是Schulman在20世纪90年代的一系列开创性论文[Sch92, Sch93, Sch96]。在这些工作中，他既提出了问题

[29] For Shannon, “almost surely” meant with probability $1 - \exp(-n)$.

并且提出了第一个解决方案，显著地表明在交互式设置中，几乎没有什么损失：可以保护常数错误率 p，甚至对抗性错误，只需在通信长度上支付一个常数因子开销！Schulman 定义了纠错 *protocols*，以及对于此类协议在比特率 p 上容忍对抗性和随机错误的意义。此类协议的 *rate*，正如经典情况一样，是原始（无噪声）协议中通信的比特数与容错协议中通信的比特数之比。纠错协议可以具有复杂结构；这里我们只讨论 Schulman 工作中的一个优雅概念，许多纠错协议都是基于这个概念的。

一个Schulman提出的中心概念是经典纠错码的交互式类似物，他称之为*tree code*。由于玩家的有限知识所迫，定义C：如果对于每个x和索引$i \in [n]$，$C(x)_i$只依赖于x的前i位，则$\{0,1\}^n \to \Sigma^n$是一个树码。在这里Σ是一个有限字母表（例如，如果$\Sigma = \{0,1\}^c$，那么C的输出长度是$m = cn$）。树码的速率C与之前一样，是输入长度与输出长度的比率，所以$R(C) = 1/|\Sigma|$。

树码质量的一个中心度量，泛化了经典码的（归一化）汉明距离，捕捉编码在首次分叉后差异的程度。更精确地说，一个树码 C 有 *relative* 距离 $\delta = \delta(C)$，如果对于任意两个输入 $x \neq x'$，如果 $C(x) = uw$ 和 $C(x') = uw'$，那么 $d_H(w, w') \geq \delta|w|$，其中 u 是它们的最长公共前缀（因此，w 和 w' 的前几位不同）。

Schulman的主要成果是存在具有常数速率和距离的树码，以及一种允许各方在交互式错误纠正中使用此类码的协议，即使对于对抗性错误也是如此。请注意，与单向通信不同，利用树码进行解码是非常非平凡的，因为玩家必须在协议期间捕获并纠正错误（而不仅仅是最后），有时还需要重新编码发生过多错误的部分。因此，与单向情况不同，树码C的输入x远非原始、无噪声的对话！

Theorem 15.32 [Sch96]. *There exist tree codes C with $R(C) = O(1)$, $\delta(C) = \Omega(1)$. Consequently, there are error-correcting protocols that tolerate adversarial (and hence also random) error rate $p < 1/240$.*

在经典错误纠正中，证明是通过概率方法进行的，因此代码不是显式的，编码和解码效率低下。这随后得到了纠正，一系列工作导致了高度高效的、概率错误纠正协议，可以容忍常数对抗错误率。最著名的此类协议在[BKN14, GH14]中（最后一个实现了最佳可能的错误率，$p < 1/4$）。没有已知的类似确定性协议。

最后，让我们讨论随机噪声模型，其中经典的单向情况在定理15.30中特别令人满意地完全理解了噪声率和纠错协议速率之间的权衡。回想一下，对于噪声率p，最优码率是$1 - H(p)$。在交互式情况下，我们所知的信息远不如精确。然而，已知这样的速率无法实现，至少对于p的非常小值来说是这样。结果对确切的通信模型有些敏感。Kol和Raz [KR13] 给出了Shannon定理的一个弱类比，Haeupler [Hae14] 也给出了另一个。忽略对数因子和模型细节，可以非正式地总结如下。

Theorem 15.33 [KR13, Hae14]. *The best possible rate of an error-correcting protocol that tolerates random noise rate p is $1 - \Theta(\sqrt{H(p)})$.*

注意，对于小的 p，$H(p) \approx p \log 1/p$，这表明交互式错误纠正比非交互式错误纠正成本更高。在这个范围内（并且忽略对数因子），单向通信需要在关于 pn 的 n-位消息中添加 $\sqrt{p}n$ 位冗余，而交互式通信需要在关于 $\sqrt{p}n$ 的 n-位通信中添加 $\sqrt{p}n$ 位冗余。

确定交互式权衡的精确度是一个非常有趣的开放性问题，特别是对于定理15.33说之甚少的固定p (而言。此外，在这些结果中，协议也是概率性的。获得与概率性协议一样高效且具有确定性的协议——对于对抗性和随机噪声都是如此——是另一个重要问题（参见[GHK+16]中的当前研究现状）。解决最后一个问题的最佳方法，也许是这个理论中最优雅的开放性问题，就是构建具有常数速率和距离的显式树码。

Open Problem 15.34. 构建显式、高效可编码和可解码的常率树码

并且恒定的相对距离。

A beautiful construction, whose correctness rests on an unproven conjecture regarding exponential sums, is given in [MS14b].

16 On-line algorithms: Coping with an unknown future

回望过去真的事事如意吗？正如俗话所说？预知未来的力量究竟有多大？以下是几个来自人们生活的具体例子，涵盖了在不知道未来时需要做出定期决策的情况。这些例子说明了我们需要在本章中讨论的模型、算法和分析的需求。

- **Investment:** 你拥有一些股票投资组合。每天（或每月、每年），根据股票价格，你做出买卖决策。你应该如何选择?
- **Gym:** 你时不时地心血来潮去健身房（或剧院）。当你明天去的时候，你应该买500美元的年卡，还是只支付50美元的单次访问费?
- **Dating:** 您寻求终身伴侣并且处于一段关系中。您应该继续与那个人约会，还是终止关系以寻找更合适的人选?
- **Memory:** 一些心理学家和神经学家认为，我们的“工作记忆”在任何时候只能同时容纳7（好吧，有些人说是4或5）个不同的事物（概念、想法、事实）。当你的环境发生变化（你开始开车、遇到知识分子或坐下吃饭）时，你会无意识地用其他事物替换其中的一些——你的大脑是如何决定丢弃什么和上传什么的?
- **Taxi:** 您是出租车调度员，一个新的请求到来，需要将某人从 A 送至 B。您应该派遣哪辆可用的出租车?

在所有这些例子中，一系列“事件”（请求、事实等）一个接一个地到来，每个都需要做出决定。每个决定都与成本/收益相关联（这取决于你的当前状态），并且它还会改变你的状态。面临这种状况的算法被称为*on-line*算法。虽然输入到达结构在精神上与我们在第14章中看到的流算法相似，但这里手头的任务比节省内存更为通用。事实上，对决策者所需的计算资源没有限制，并且模型是一个纯粹的信息论模型。它仅隔离了核心方面：当未来信号未知时，随着每个信号的到来，最佳行动方案是什么？从各种例子的多样性中可以看出，这是一个极其通用且重要的问题，我们只会触及一些基本方面和例子。关于这个主题的全面书籍是[BEY05]，而一本较新的书籍是[Haz16]。这些以及其他在此引用的资料解释了这个重要领域与博弈论（以及玩得好的策略）、凸优化、学习理论、归纳推理等之间的联系。

最基本的问题是：如何对在线算法的质量进行建模？Sleator和Tarjan [ST85] 提出了一个被称为 *competitive analysis* 的大胆答案。与许多先前的研究相反，他们主张完全忽略关于未来事件可能的“先验分布”的任何知识。相反，他们建议将在线算法在每个输入序列上的性能与最佳算法（具有先见之明）的性能进行比较；一个最优的“离线”（先知）算法，它知道完整的输入序列 *before* 并做出任何决策。简而言之，如果一个在线算法对于 *every* 个可能的输入序列，其成本在最优离线算法成本的一个因子 c 以内，则称该算法为“c-竞争”。如果一个算法对于某个有限的 c 是 c-竞争的，则称该算法为 *competitive*。这个定义中的大胆之处在于，它建议可以取得一个有限的 c，这并不依赖于输入（特别是输入长度）。直觉，也许还有生活经验，表明在上述情况中了解未来应该会赋予巨大的先见之明。然而，这种直觉往往是错误的！从这篇开创性的论文开始，它有力地证明了这种可能性，发现了许多具有竞争性的在线算法场景的例子。我将给出几个（以及一个非例子）这种令人惊讶现象的例子，但首先我将更精确地定义相关术语。

让我们考虑一个非常一般的定义（在[BDBK+94]中引入并命名为*request-answer games*），其中该领域的多数特定场景和模型都是特殊情况。对于任何序列$z = z_1, z_2, \ldots$和任何小于其长度的整数t，令z_t为序列的第t个元素，z^t为z的第t个前缀，即$z_1, z_2, \ldots, z_t$。

一个 *on-line problem* 由事件集合 E、决策集合 D 和每个整数 t 的成本函数族 $C = \{C_t : E^t \times D^t \to \mathbb{R}\}$ 定义。在线问题的输入是一个序列 $e = e_1, e_2, \ldots, e_T \in E^T$。因此，每个 $e_t \in E$ 是时间 t 的事件。总时间（= 事件数量）T 是任意且对算法来说是未知的（并且我们将看到，如果需要，可以将其视为无限）。

一个 *on-line algorithm* A 是一个函数序列 $A_t : E^t \to D$，指定了给定到目前为止的输入的下一个决策。因此，对于每个 t，通过 $A(e^t) = A_1(e^1), A_2(e^2), \ldots, A_t(e^t) \in D^t$ 表示 t 在前 t 个时间步内由 A 所做的决策序列是明确的。*cost* A 在步骤 t 中产生的成本是 $C_t(e^t, A(e^t))$，而 $C_A(e)$ 表示 *total cost* 在所有决策中产生的成本，即 $C_A(e) = \sum_{t=1}^{T} C_t(e^t, A(e^t))$.

序列 *off-line* (的最优) 成本容易描述——它仅仅是最佳决策序列产生的最低成本。也就是说，$OPT(e) = \min_d \sum_{t=1}^{T} C_t(e^t, d^t)$，其中最小值是在所有可能的 $d \in D^T$ (d 上取的，可以看作是一个假设的离线算法的动作，该算法可以提前看到整个输入 e。

一个在线算法被称为“c-竞争性”（对于由(E, D, C)定义的给定在线问题）如果存在一个通用常数[1] M，使得对于每个有限长度e (的每个序列)，我们有

$$C_A(e) \leq c \cdot OPT(e) + M.$$

有时我们说 A 有 *competitive ratio* c。有两个简单的观察结果。首先，如果 A 是 c-竞争的，没有其他算法能比它好超过 c 的因子。此外，这个陈述不仅适用于最终的输入序列，也适用于它的每一个前缀。

让我们从一个简单的在线问题开始，这个问题没有竞争算法。假设 $E = D = \{0, 1\}$。问题是猜测序列中的下一个比特。因此，无论过去如何，如果猜错了，你将支付（比如说）1美元，否则不支付。用符号表示，$C_t(e, d) = e_t \oplus d_{t-1}$ 对于任何 $t > 1$ 和任何两个长度为 t 的序列 d, e。显然，对于每个 e (，只需选择 $d = e$)，就可以使 $OPT(e) = 0$。然而，对于每个（确定性的）算法 A 和每个长度 T，都存在一个长度为 T 的序列 e，使得 $C_A(e) = T - 1$。简单地任意选择 e_1，然后后续 e_t 值作为算法预测的补码，即 $e_t = 1 - A_{t-1}(e^{t-1})$。因此，$A$ 没有有限的竞争比。[2] 这个例子提出了一种非常有用的竞争分析定义的观点：序列 e 可以看作是由一个具有算法 A 完全信息的对手，一次生成一个符号。

我们现在转向两个非常一般的例子，其中确实存在非平凡的在线竞争算法！这些例子将形式化上述一些非正式例子。请注意，其中一些相当简单，您尝试自己解决它们会很好。例如，为健身房问题（一个过于熟悉的“购买或租赁”问题的特殊情况）找到一个2-竞争算法。

16.1 Paging, caching, and the k-server problem

考虑上述“工作记忆”示例的以下形式化，这在组织成层次结构的计算机内存系统中实际上相当实用（正如我们的大脑可能组织的那样）。存在快速内存，或*cache*，可以存储k数据项，以及一个更大的慢速内存。事件集E简单地是所有数据项，其中任意k最初在缓存中，其余在慢速内存中。系统访问数据项的请求到达（这是输入序列e）。如果请求的数据项e_t在缓存中，则无需做出决定，也不会产生成本（由于从缓存中快速检索）。然而，如果请求的数据项e_t不在缓存中，它必须被移动到那里，并且为了给它腾出空间，必须决定将哪个缓存项移动到慢速内存。无论移除哪个项，成本都是1（即支付访问慢速内存的时间）。这个问题被称为k-*paging*问题。

显然，一个对手可以创建一个请求序列，使得任何在线算法在每一步都支付 $1，只需请求缓存外的项目。但请注意，在这种情况下，即使一个

[1] In some contexts, M is allowed to grow, but it should be kept asymptotically smaller than OPT.

[2] With the natural extension of on-line problems and competitive analysis to allow randomness, a similar argument can show that even a probabilistic algorithm A cannot achieve a competitive ratio better than $T/2$.

离线算法每隔一段时间必须访问慢速内存。问题是：这个问题有没有一个竞争算法？竞争比可能依赖于k，但不依赖于序列长度。在继续阅读之前，请思考一下！

自然启发式算法（我们的大脑可能也在使用）被称为“最近最少使用”（LRU）算法。此算法简单地在缓存填满时，丢弃最后被请求的缓存项[3]。此算法显然符合上述通用模型。Sleator-Tarjan论文[ST85]中对该算法的分析是初始示例之一，其中他们精确地确定了其质量（比我们陈述的更为通用）。

Theorem 16.1 [ST85]. *The algorithm LRU is k-competitive for the paging problem. Moreover, no on-line algorithm for this problem has a competitive ratio smaller than k.*

让我们现在将上述设置推广很多，到著名的 *k-server* 问题，它在 [MMS90] 中被引入。它可以被视为本章开头提到的出租车问题的具体模型，以及具有以下结构的许多其他问题的模型。请求是某些度量空间中的点，有有限数量的资源（k）来处理请求，服务成本来自服务所经过的距离。正式地，固定一个度量空间 $M = (E, dist)$，使得请求集合 E 简单地是度量空间中的点，而 $dist$ 是一对点之间的距离函数（满足三角不等式）。k “服务器”最初位于 M 的某些点上。现在，一系列请求（即，M 的点）一个接一个地到达，你的任务是决定将哪些 k 服务器发送到那里。成本是服务器从当前位置到新请求点的行驶距离。

应清楚，这个分页/缓存问题是度量空间为*uniform*的非常特殊的情况，即每对点之间的距离都相同（比如说，1）。在这个一般设置中，我们能否有一个竞争算法？[MMS90]中提出了肯定的猜想；确实，作者们猜想可能的竞争比是k，就像在均匀情况中一样。该领域的一个主要结果，由Koutsoupias和Papadimitriou [KP95]提出，通过他们对该所谓的工作函数算法的巧妙分析，非常接近证实了这个猜想。

Theorem 16.2 [KP95]. *The work function algorithm is $(2k-1)$-competitive for the k-server problem.*

它不会让读者感到惊讶，在线设置中，随机性也可以发挥关键作用，并且该模型扩展到允许概率算法优于确定性算法。例如，存在一个概率算法（通常对于每个输入都是正确的，在随机投掷硬币时具有高概率）用于 k-分页问题，它是 $O(\log k)$-竞争的 [FKL+91][4]。正如我们所知，k 是分页问题可能达到的最佳竞争比，我们看到随机性可以在这个参数上证明指数级更强大。这种力量的一个重要来源是，现在生成输入序列的对手知道算法，但不知道随机投掷硬币。关于对手和概率在线算法（以及捕获它们的模型）之间相互作用的微妙问题在 [BDBK+94] 中进行研究。这些模型展示了随机性可以增强在线算法，最多增加一个多项式因子的情况。

16.2 Expert advice, portfolio management, repeated games, and the multiplicative weights algorithm

最后，我们来到了关于预测未来的问题，这些问题几乎每个人都每天都会关注，比如该相信哪个气象频道，该听哪个金融专家的建议。令人惊讶的是，在非常一般的情况下，你只需了解过去的表现，就可以几乎和最佳专家 *in hindsight* 一样做得好！如果你想知道为什么这样惊人的可能性没有被每个人用来在股市上做得像沃伦·巴菲特这样的传奇投资者一样好，那么，这是一个好问题，有很多答案。请阅读下面的理论结果，你可以自己决定是否在家尝试它们。它们在许多应用中确实得到了广泛的使用，包括金融投资。

[3] Practically one can, e.g., keep the items in the cache in an ordered list, placing each newly requested item at the start of the list, and always discarding the last one when needed.

[4] A possible $(\text{poly}\log k)$-competitive probabilistic algorithm for the general k-server problem remains open. Major recent progress on this problem can be found in [BCL+18].

明确指出，我们正在更改模型！现在我们将比较一个在线算法与一个受限的离线算法族（专家）中的最佳算法，而不是与最佳离线算法进行比较。原因很简单。在这些设置中，通常不存在具有第16.1节意义上的竞争算法，而与事后最佳的专家进行比较的新、更有限的设置仍然非常普遍。

许多这些令人惊讶的结果的关键是一个单一的、极其重要的“元算法”，称为*multiplicative weights update*算法，它可以被视为在做出未来决策时考虑过去信息的一种平滑方式。因此，它可以被视为一种学习算法，并且在机器学习（我们将在稍后讨论）中确实存在。但是，再次强调，在本章的精神下，不对未来事件做出任何假设，并且它们可以被了解算法的对手所确定。许多人在不同的应用领域中发现了这个算法的变体。甚至有人提出，这个算法是自然界独立“发现”的，并且它自然地出现在进化过程中[CLPV13, MP15]！这个算法足够简单，可以被简单生物体以及可能甚至由基因以分布式方式实现。这个算法在第8.8节中出现过，在伪随机性的完全不同背景下；你应该将其与我们在这里讨论的内容进行比较。关于这个算法的许多化身和应用的优秀综述是[AHK12]。

我描述了该算法的一个简单变体，称为Littlestone和Warmuth的*weighted majority*算法[LW94]，该算法旨在处理二元预测.[5]例如，明天会下雨吗？或者，某只股票的价格明天会上涨还是下跌？该算法将建议一种方法，将关于此类事件的不同“专家”的“建议”聚合为一个决策，随着时间的推移，*on every sequence*，其表现将几乎与该序列最佳专家的预测一样好。让我们将手头的在线问题形式化，然后讨论该算法及其性能分析。

事件 E 是一对 (b, v)，其中 $b \in \{-1, 1\}$ 和 $v \in \{-1, 1\}^k$，应理解为以下内容。比特 b 是关于当前时间步现实的事实（例如，某只股票价格今天是否上涨或下跌）。向量 v 是关于下一时间步现实的 k 个不同专家的意见（例如，那只股票价格明天是否会上涨或下跌）。在此，算法做出决定，即关于明天的 1 位预测。在下一步，现实被揭示，算法学习哪些预测（包括自己的和专家的）是正确的，哪些是不正确的。错误决定的成本，对于算法和专家来说，是（比如说）1 美元。因此，我们将计算错误（或错误预测），算法的目标是在与给定 k 中最佳（事后看来）专家的错误相比时，最小化这些错误。请注意，犯最少错误的专家可能会随时间而变化。

现在让我们描述加权多数算法，正如之前提到的，它可以看作是乘性权重更新的简单版本。更新将是对我们信任不同专家的演变估计。最初，我们平等地信任他们，因此给每个分配权重1。随着我们观察到一些专家犯错误，我们对他们的信任将按一个常数因子减少，*multiplicatively*。我们如何聚合他们的预测的决定将简单地根据他们当前的权重进行加权多数。让我更具体一点。

Let $w_t(i)$ 表示时间 t (时 i 第专家的权重，因此对于 $t = 0$，我们有 $w_0(i) = 1$ 对于所有 $i \in [k]$)。用 W_t 表示时间 t 时的总权重，即 $W_t = \sum_i w_t(i)$。因此 $W_0 = k$。在每一步，当当前值 b 被揭示时，我们可以判断在上一轮预测中哪些专家预测正确，哪些预测错误。我们按照以下方式降低对每个犯错的特定专家的信任度。如果专家 i 在步骤 t 中的预测错误，那么我们设置 $w_t(i) = w_{t-1}(i)(1 - \epsilon)$。请注意，更新是乘法的，并且参数 ϵ 通常需要仔细选择，以权衡学习的速度和预测的波动性。将 ϵ 视为一个小的常数，如 .01 是有益的。最后，算法根据当前权重 $w_t(i)$ 对专家的预测 $v_t(i)$ 进行加权多数预测。更确切地说，算法预测 $sign$ 的 $\sum_i w_t(i)v_t(i)$。此算法几乎具有 2-竞争性（相对于最佳专家）。

Theorem 16.3 [LW94]. *For any t, let M_t be the number of mistakes made so far by the weighted majority*

[5]This algorithm follows similar ones originally developed for *boosting* in computational learning theory, which we discuss in Chapter 17.

algorithm, and let $m_t(i)$ be the number of mistakes made so far by the ith expert. Then for every i,

$$M_t \leq 2(1+\epsilon)m_t(i) + O((\log k)/\epsilon).$$

分析相当简单，遵循算法的直观。以下两个断言将专家和算法的权重和错误联系起来，并通过对 t 的归纳进行。首先，对于任何专家 i，其权重在每次错误中减少 $(1-\epsilon)$，因此在时间 t 时正好是 $w_t(i) = (1-\epsilon)^{m_t(i)}$。作为 $W_t = \sum_i w_t(i)$，我们有对于每个 i，$W_t \geq w_t(i) = (1-\epsilon)^{m_t(i)}$。其次，当算法出错时，总权重为 $W_t/2$ 的专家必须按照加权多数规则出错，并且（作为 $W_0 = k$）我们有 $W_t \leq k(1-\epsilon/2)^{M_t}$。将 W_t 的两个界限结合起来并取对数，证明了该定理。

一个自然的问题是，2的竞争比（可能令人惊讶且良好）是否是最佳可能的。也许更令人惊讶的是，借助随机化，我们可以接近1的竞争比，即尽可能少犯像事后最佳专家那样的错误。这在小石子和瓦尔穆思的同一篇论文中有展示，该论文还建议了随机加权多数算法；它以与确定性版本完全相同的方式更新对专家的信任，并仅使决策概率化。该算法不是使用加权多数，而是以概率$w_t(i)/W_t$简单地跟随第i位专家的预测。这恰好平滑了上述分析中的最坏情况，并在*expectation*中产生了一个几乎1-竞争算法。

Theorem 16.4 [LW94]. *For any t, let M_t (which is now a random variable) be the number of mistakes made so far by the randomized weighted majority algorithm, and let $m_t(i)$ be (as before) the number of mistakes made so far by the ith expert. Then for every i, the expected number of mistakes of the algorithm is bounded by*

$$\mathbb{E}[M_t] \leq (1+\epsilon)m_t(i) + O((\log k)/\epsilon).$$

在线算法的性能通常用 *regret* 来表示，即在线算法性能与事后最佳性能之间的最大差距。让我们用这种语言来呈现最后的结果。设 $i*$ 是在某个数字 T 轮次中犯错误最少的专家。因此，（期望）遗憾 $\mathbb{E}[M_T] - m_T(i*)$ 被限制在 $\epsilon m_T(i*) + O((\log k)/\epsilon)$ 内。使用平凡的界限 $m_T(i*) \leq T$，并选择ϵ (可以任意设置的) 来平衡界限中的两个项，可以看到在任何数字 T 的步骤之后，遗憾被限制在

$$\mathbb{E}[M_T] - m_T(i*) \leq O(\sqrt{T \log k}).$$

因此，对于非常大的 T，每步的平均遗憾仅为 $O(1/\sqrt{T})$ (，尽管它可能是 1)。这种对 T 的依赖性是最佳可能的。

不容易想象如何将算法从这种二元决策设置推广到决策和成本连续的情况，例如，在区间[0, 1]内，而不是二元$\{-1, 1\}$。然后，更新取决于专家犯错的幅度，如果这种损失是$g \in [0, 1]$，则算法将按因子$(1-\epsilon)^g$减少其权重。概率算法将像以前一样选择随机选择的专家，其概率与权重成正比。同样的分析表明，概率算法的成本将尽可能接近最佳专家的成本。这种连续推广的一个美妙应用是玩*repeated games*，这是由Freund和Schapire[FS99]发现的，我们将在下面简要描述。

自然设置中，我们试图在一系列对抗性移动中做得好，当然是一个 *game*。考虑一下博弈论中熟悉的零和博弈设置（例如，想想剪刀石头布）。一个真实矩阵 M 如下描述了一个两人（完全信息）零和博弈。行玩家从 M 的 i 行中选择，列玩家从 M 的 j 列中选择。行玩家的收益，即列玩家的损失，是 $M(i,j)$。近一个世纪以来，人们已经理解了如何玩好这类游戏，冯·诺伊曼发现了最小-最大定理，该定理确定了游戏的 *value*（对双方来说都是最佳可能的结果）。线性规划提供了一个多项式时间算法，计算两个玩家的最优混合策略，从而实现游戏价值。

这个理解会杀死主题吗？远非如此，以下问题在20世纪50年代的博弈论中已经被考虑。假设玩家不知道M，并且只有在游戏后才知道他们的损失/收益。假设M太大，以至于线性规划效率不足。假设你的对手以次优策略进行游戏——你能获得比游戏价值更多的收益吗？如果游戏只玩一次，对这些问题的处理并不多，但如果玩家玩相同的游戏*repeatedly* (，例如，两个竞争对手反复设定他们产品的价格)，在线设置给这个问题带来了新的生命。你能看到如何使用上述算法以最佳可能的方式对抗*any*对手进行渐近游戏吗？现在假设M是已知的。想法是使用上述加权多数算法。将你的纯策略（例如，如果你是行玩家，不同的行）视为你的专家。在每一轮，随着你了解对手的移动，揭示了你的每个选择/专家的收益/损失。这允许更新权重，这些权重作为你下一个动作的混合分布。这种玩法基本上实现了对抗对手的最佳纯策略所能实现的效果，因此它至少与游戏价值一样好。此外，如[FS99]中解释的那样，这个算法还导致了对冯·诺伊曼最小-最大定理的简单新证明。当M未知，并且每一步只揭示收益时，如何玩游戏？碰巧，这个算法的变体在这种情况下也有效（性能略差）。

让我们以金钱作为结论，这可能是你坚持看到本章结尾的原因。我们现在讨论*Portfolio Management*问题。有k只股票，你在最初投入了一些金额（比如说，1000美元），按照向量$p = (p_1, p_2, \ldots, p_k)$分配，其中$p_i$的份额投资在第$i$只股票中。这个向量$p$描述了你的投资组合。这些股票的值每天都会变化，问题是如何重新投资它们的总价值。一个常用的方法（特别是对于懒惰的投资者来说很好）是简单的*rebalancing*——再次根据p分配新价值。当然，问题是：哪个p将在长时间内产生最佳性能？只是为了展示这种简单策略如何在波动市场中产生指数级收益，比如说你投资了苹果和微软。进一步假设，随着时间的推移，苹果股票保持在1美元，而微软股票在奇数天为$\frac{1}{2}$美元，在偶数天为2美元。如果你的投资组合是$p = (\frac{1}{2}, \frac{1}{2})$，简单的计算将显示，在任意偶数$t$天后，重新平衡将使你的财富*exponentially*增长一个因子$(9/8)^{t/2}$。当然，如果你的投资组合只选择了这两只股票中的一只（无论是$p = (1, 0)$还是$p = (0, 1)$），你的财富在时间上基本上将保持不变.[6]（事后看来）我们将会比较所有这些选择中最好的！我们希望设计一个在线算法，该算法可以在看到股票价值后每天选择一个*different*投资组合，可以与*best fixed*投资组合再平衡相竞争。Cover [Cov91] 的开创性论文提出了这个问题，并定义了一个在线算法为*universal*，如果，粗略地说，对于每个t和$\epsilon > 0$，当最佳投资组合在t天内实现增长c^t时，那么在线算法实现增长$(c - \epsilon)^t$。令人惊讶的是，这可以做到，Cover描述并分析了一个这样的通用算法。他的算法和随后的算法在股票数量上是指数级的，直到Kalai和Vempala [KV03] 找到一个具有与Cover相同性能的多项式时间算法（稍后[HAK07]给出了一个更有效的算法）。这些算法显然不是乘性权重更新形式；然而，[HSSW98]表明这个问题适合一般框架，并为此提供了一个算法。

[6]Try finding an example where balancing a fixed portfolio will *shrink* your assets exponentially fast.

[7]In a somewhat weaker sense than defined above, which is suitable for situations where exponential growth can be expected.

17 Computational learning theory, AI, and beyond

在这个章节中，我们面临一个极其通用的建模问题：如何定义，然后设计从经验中学习并利用它来应对可能出现的新、不同情况下的算法。在这个通用性下，这样的算法必须能够学会像孩子一样走路和说话，飞翔或游泳，像年轻动物一样寻找食物和庇护，像科学家一样从数据中构建理论，像数学家一样证明定理，演奏乐器和作曲，领导军队或创办公司，讨论哲学和情感，笑笑话，繁衍后代等等。我们周围的所有生命形式（尤其是人类）似乎都天生具有一些基本的学习能力和驱动力，然后通过生活积累经验并利用它（以不同程度的成功）来生存、繁荣和繁衍。即使忽略算法在生物体中执行这些任务是如何进化的巨大科学问题（计算机科学应该在其中发挥重要作用的问题），我们也可以提出一个更具体的问题：如何设计具有这些能力之一的计算机系统。

如您所知，这个项目的某些方面已经变成现实。我们生活在一个“机器学习”取得巨大进步的时代。许多计算机系统实际上是通过*learned*，而不是直接编程，来完成一些惊人的壮举，其中许多我们已经使用或很快就会使用。这些系统提供个人推荐（例如书籍、电影等），图像中的特征和内容识别，语言翻译，医疗诊断和治疗，天气和股市预测等。学习程序现在在围棋和国际象棋中击败人类，并且可能会完全取代数据科学家，而不仅仅是协助他们进行“数据科学”和“知识发现”。自动驾驶汽车几乎就在拐角处，承诺将改变我们的日常生活。还有无数的更多例子，许多是由*deep learning*革命每天产生的。事实上，在这个（可能是第三次或第四次）人工智能梦想（或噩梦）的复兴中，有些人预测，在本世纪内，人类智能和认知的大部分方面将被机器并行或超越。

我们不会在这里讨论这些问题或推测的大部分内容。当前快速进步的很大一部分是基于*heuristics*，它使用了惊人的计算能力，今天可用的巨大数据量（以及始终如一的，高效的算法）。许多实际成功远未得到理解，鉴于这些机器学习启发式方法在社会中的大规模部署，迫切需要的理论理解将会得到改善。在本章中，我们只将讨论一些计算学习的一些初始、具体模型、算法和数学结果，这些结果涉及这个极其复杂主题的一些方面。我们将特别讨论关于这些模型学习能力和局限性的已知信息。虽然它们仍在发展，新的模型也在被引入，但这些初步工作中提出的某些原理和模型自然地被用于当前机器学习的工作中，并且它们也可能用于从进化到认知的自然现象建模。

Learning 这是一个大词，包含多重含义，关于这个词，历代学者都进行了广泛的辩论和写作。Tom Mitchell [Mit97] 给出了关于学习算法的一个经常引用的（操作性的，而不是认知性的）定义：“如果一个计算机程序在经验E和某些任务类别T以及性能度量P方面学习，那么它的性能随着经验E在T中的任务上的提高，按照P来衡量。”

在这一章中，我们讨论了可以赋予本定义中未定义概念的一些具体含义。让我们从“经验”开始，即学习者与环境的互动。根据这种互动，可以将模型大致分为两类[1]。第一类，*supervised*学习，隐含地假设存在一个“教师”，为原始数据提供辅助信息。教师可以是虔诚的父母或实际的课堂教师，也可以是任何知识实体（如为他们的图片贴标签的互联网用户），甚至可以是环境，根据生物的行为进行奖励/惩罚，无论是觅食食物还是投资股市。第二类，*unsupervised*学习，不假设这样的教师或任何辅助信息，学习者必须在没有这种帮助的情况下理解数据。自然地，这些情况的学习要困难得多，我们建议读者参考广泛的教科书[Mur12]，其中介绍了各种模型和技术。教科书[GBC16]专注于最近受到青睐的*deep learning* (元模型，该模型在监督学习和无监督学习中都得到应用。

[1] And many intermediate ones.

设置）。这里我们只讨论监督学习。

主要（一般）任务我们在监督学习类别中讨论的是 *classification*，在各种文本中也被称作 *identification*、*concept learning* 以及其他旨在捕捉自然能力以 *generalize* 并从示例中提取规则（或函数、或模式等）的术语。我们将完全忽略关于从示例中学习的可能性、这样做的原因以及哲学标题下所有属于 *the problem of induction* 的众多哲学辩论。[2]

17.1 Classifying hyperplanes: A motivating example

在思考以下风格化识别问题和后续讨论时，考虑一些现实生活中的熟悉问题可能是有用的。例如，一个小孩在学习从一系列被标记为“这是一只猫”或“这不是一只猫”的图像中识别特定的动物（例如，猫）。或者，也可以考虑一个计算机程序（如亚马逊和其他公司使用的程序），它试图从一系列书籍描述（例如，标题、作者和简介）中识别特定读者的阅读喜好，这些描述带有“我喜欢这本书”或“我不喜欢这本书”的标记（由读者提供）。或者，考虑一位科学家，他试图从一系列人（由一系列生理特征表示）中识别（相关疾病的）迹象，并对每个人是否具有特定的遗传标记进行标记。在这里，我们将上述现实生活中的动物、读者和疾病用高维欧几里得空间中的*hyperplanes*替换，并将任务替换为从空间中的标记点中识别特定的超平面（这个问题通常被称为*linear classification*）。超平面可能看起来是一个极其合成、简单的概念选择来学习，没有人真正关心。然而，识别超平面非常重要，并且比你想象的与识别动物、文学品味、疾病和其他许多概念的相关性要大得多。

确实，让我们看看*spam*电子邮件的概念可以自然地被视为一个超平面（这正是许多垃圾邮件过滤器如何表示它们的）。假设典型的电子邮件词汇来自一个包含1,000个单词的特定词汇表。将电子邮件表示为向量 $x \in \mathbb{R}^n$，对于n =1, 000，其中第j个条目$x(j)$表示词汇表中的第j个单词出现的次数.[3]现在，我们每个人可能都有自己的垃圾邮件标志。例如，如果你看到“Viagra”一次，那显然是垃圾邮件。但可能你可以容忍“pharmaceutical”出现四次，“click here”出现两次，以及两个“pharmaceutical”和一个“click here”的组合。给每个单词分配权重$h(j)$，并确定总权重$\langle h, x\rangle = \sum_j h(j)x(j)$是否超过某个阈值值$v$是有意义的。那么，相关超平面由线性方程[4] $\langle h, x\rangle = v$表示，这样电子邮件x就被认为是*spam*如果$\langle h, x\rangle > v$和*not spam*如果$\langle h, x\rangle < v$。当然，这种垃圾邮件的超平面表示只是一个（非常简单）模型，但让我们假设它是一个准确的模型。现在考虑垃圾邮件过滤器将如何仅根据你标记的不同电子邮件为“这是垃圾邮件”和“这不是垃圾邮件”，来识别你的“个人”超平面（这取决于你的个人品味和容忍度，这些可能甚至是你自己无法指定的潜意识价值观）。让我们首先形式化这个问题，然后讨论解决这个问题的算法。

我们可以不加损失地做出以下两个假设。假设任何向量 h 是一个 *unit* 向量（即，具有欧几里得范数 1），并且 $v = 0$（即，超平面通过原点）。[5] 现在考虑识别空间中通过原点的超平面在 $\mathbb{R}^n$ 中表示的未知向量 h^* 的任务。给出的信息是一个（可能是无限的）点对序列（通常称为 *labeled examples*）(x_1, b_1), $(x_2, b_2), (x_3, b_3), \ldots,$，其中 $x_i \in \mathbb{R}^n$ 是点，$b_i = \text{sign}(\langle h^*, x_i\rangle)$ 是点 x_i 所在的“侧面”（或半空间）h^*，即如果内积是负的，则 -1，如果正的，则 $+1$（如果 x_i 在 h^* 上，则 0，尽管我们可以假设这种幸运的情况永远不会发生）。

我们将把这个（以及一般分类问题）视为一个在线问题，类似于已经研究过的问题

[2]Note that it is sometimes amusing to read how people, who as babies could neither speak nor reason at all, argue in sophisticated language the impossibility of learning or gaining knowledge from examples.

[3]Note that this representation, which again is a very common way to treat text, ignores the order of words and keeps track only of their frequency.

[4]With h being the normal vector to the actual hyperplane in space.

[5]We lose no generality by adding one more dimension.

第16章中，即算法在观察序列后，在看到每个新示例后应提出一个*hypothesis*。[6]等价地，我们寻求一个算法，对于每个有限样本产生一个假设。

这里有这个分类任务的两个自然（高效！）算法。

Perceptron algorithm 此算法将在每个输入对上产生一个关于 h^* 值的新假设，并将无限期地这样做，如果其最后一个假设与下一个标记的例子不一致，则明智地修改它。我们首先将 $h^0 =$ 设置为 0，在处理完第一个 $t-1 \geq 0$ 输入对后，h^{t-1} 为最后一个假设，然后在下一个输入对 (x_t, b_t) 上进行如下操作：如果一致则不采取行动，如果不一致则倾向于 x_t。更精确地说，我们按以下方式设置 h^t：

- **Correct classification:** 如果 $b_t =$ 符号$(\langle h^{t-1}, x_t\rangle)$，设置 $h^t = h^{t-1}$。
- **Incorrect classification:** 如果 $b_t \neq$ 符号$(\langle h^{t-1}, x_t\rangle)$，则设置 $h^t = h^{t-1} + b_t x_t/\|x_t\|$。

感知器算法由Rosenblatt [Ros58] 发明，随后Novikoff [Nov62] 进行了分析（随后对该分析进行了许多改进和推广：参见综述[MR13]）。我将在定理17.1中描述该分析，感兴趣的读者可以提前跳读然后再返回这里。

让我们现在转向一个可能更简单、更自然（尽管效率略低）的算法，该算法使用线性规划.[7]

Linear programming 每个有限数量的标记示例自然定义了一个由 h^* 满足的线性不等式系统。更精确地说，假设我们有一个 s（参数）标记示例。对于每个这样的输入对 (x_i, b_i)，其中 $i \in [s]$，写一个线性不等式 $b_i\langle h, x_i\rangle \geq 0$，其变量是 h 的坐标。该算法使用一个高效的线性规划算法（回想一下，这个问题在 $\mathcal{P}$ 中）解决由此产生的线性不等式系统。其输出 $\hat{h}_s$ 是隐藏超平面 h^* 的假设。

注意，此算法可以改编为在线算法，就像之前的那个一样。具体来说，从 $h^0 = 0$ 开始，在每次后续输入对 x_t, b_t 上，输出一个假设 h^t，如下所示：

- **Correct classification:** 如果 $b_t =$ 符号$(\langle h^{t-1}, x_t\rangle)$，设置 $h^t = h^{t-1}$。
- **Incorrect classification:** 如果 $b_t \neq$ 符号$(\langle h^{t-1}, x_t\rangle)$，则 h^t 为从第一个 t 个示例导出的不等式系统的线性规划输出。

它至少值得注意一个学习超平面问题的明显扩展，这个问题被两个算法轻松处理。这个扩展证明极其有用，并展示了当与简单的约简相结合时，该问题的力量。超平面通过线性方程将 {v*} 划分。相反，我们可以考虑由高次多项式方程定义的 {v*} 划分，这给出了一类更丰富的识别问题。然而，请注意，后者可以简单地约简为前者。具体来说，考虑次数为 {v*} 的多项式 {v*} : {v*}。让 {v*} 表示次数最多为 {v*} 的多项式中的单项式数量。给定一个点 {v*}，它可以映射到一个点 {v*}，该点在每个这样的单项式中评估 {v*}。{v*} 然后，如果 {v*} 是 {v*} 的系数列表，那么 {v*} 的值显然由 {v*} 形式 {v*} 给出。因此，线性不等式（{v*} 符号 {v*}）的标记示例可以转换为线性不等式（{v*} 符号({v*}）的标记示例，并且可以使用上述算法使用这些数据来识别 {v*}，从而识别 {v*}。这种约简的成本自然会随着 {v*} 而增加。各种实现进一步效率和泛化的方法属于“支持向量机”和“核方法”等主题，我们在此不予讨论。

[6] While similar, the focus is different. In on-line algorithms, the task is usually known in advance, and an algorithm designer can use arbitrarily sophisticated methods and analysis in solving it. In machine learning, usually there is far more limited information about what is to be achieved, and the eventual (prediction) algorithms extracted form the data are typically simple, coming from a small arsenal of methods.

[7] It is worth mentioning that the perceptron algorithm itself can be adapted to solve general linear programming problems [BD02].

[8] This is called the *Veronese map* (or embedding) in algebraic geometry.

17.2 Classification/Identification: Some choices and modeling issues

我们将交替使用术语 *classification* 和 *identification*，因为不同的文献集用它们表示相同的概念。

当前的任务是从给定的*functions* (集合中识别一个，有时称为“概念”、“谓词”或“规则”)，使用流式*labeled*示例中的数据。这项任务似乎非常具体和专注。然而，仍然需要做出一些重要的选择，我列举了一些。我们不会讨论这些问题的*awareness*识别算法及其建模决策。在讨论例如儿童（或动物）学习算法及其进化时，这是一个高度非平凡的组成部分。但是（如前几章所述），考虑到这样一个特定的分类任务，对于人类设计的算法，完全有理由假设这种意识。

17.2.1 Target class of functions

这个函数族（通常称为 *concept class*）是我们试图识别单个函数（或概念）的范围。通常它是一个从固定域 X 到某个固定范围 Y 的函数集合 $F = \{f : X \to Y\}$，任务是从中识别一个。域 X 和范围 Y 可以是有限的或无限的，甚至是连续的（尽管对于实际算法，连续对象通常由离散近似表示）。以下是一些例子。我们将在以后讨论其中的一些。

1. 给定域上的所有线性方程。

2. 给定域上的所有多项式。

3. 平面中所有轴平行的矩形（例如，代表一定年龄和收入区间的所有人）。更一般地，可以考虑具有 n 属性的 $\mathbb{R}^n$ (中的平行六面体)。

4. 平面中的所有圆（例如，代表从某个消防站或医院一定距离内的所有住宅）。更一般地，可以考虑 $\mathbb{R}^n$ 中的球体。

5. 所有文字的合取（变量或其否定），例如，$x_7 \wedge \neg x_2 \wedge x_4$ 在一组布尔变量上（例如，表示您想购买的汽车中某些特征（如GPS、FWD、ABS或定速巡航）的联合存在或不存在）。

6. 所有关于给定布尔变量集的DNF公式（合取的析取）例如，表示您愿意购买具有给定特征集中任何一个特征的汽车。

7. 所有由有限自动机可计算的功能。

8. $\mathcal{P}$ (中所有表示您可以实际高效验证的属性的所有函数。

17.2.2 Hypothesis class

识别算法必须对给定数据做出假设。这些假设来自一组函数 $H = \{h : X \to Y\}$，通常（但不总是）包含 F。由于 H 的元素是学习算法的输出，它们通常由计算这些假设的算法（或机器）的类别指定。例如，H 可能包括小公式、低次多项式、有限自动机、决策树、资源有限的图灵机等等。

[9]Here and in the next item, X is not discrete. However, we can replace the reals $\mathbb{R}$ with the rationals $\mathbb{Q}$ without any consequence for the issues at hand.

17.2.3 Admissible presentation of data

我们设置的方式是，数据以一系列标记的示例$(x_i, f(x_i))$的形式，为一系列点$x_i \in X$提供。一个中心问题是决定如何选择这些点$\{x_i\}$的序列。当然，为了使学习模型尽可能通用，人们试图对自然界在“自然”情况下提供此类示例的方式假设得最少。这表明让一个*adversary*生成序列。[10]非常广泛地说，这些对手可以通过两种可能的方式之一进行限制（与第17.2.4节中关于质量度量的两个主要评估标准相关）：

1. 对手可以选择任意序列，该序列最终包括 X 中的每个点。
2. 对手可以在X上选择一个完全任意的概率分布，之后序列的元素独立地从该分布中抽取。

我们上面做出的一个关键假设是数据点的标签总是正确的。这当然在一般情况下是不现实的，已经研究了各种放宽条件，允许存在一些噪声标签（随机或对抗性）和扰动标签（在范围 Y 上的某些度量中是“小”的）。我们不会在这里讨论这些重要的推广，尽管这里提到的某些积极结果可以扩展以容纳这些错误，并且许多算法和启发式方法特别设计来容忍它们。

在正交方向上，通过学习算法获取数据的另一种自然形式允许它们询问 *queries*。已经考虑了多种查询类型，包括让算法选择一些点 x_i 以及请求违反当前假设的点 x_i。同样，当允许这些查询时，查询的数量（或频率）是另一个重要参数。查询可以为学习算法增加显著的力量。我们在此处不再进一步考虑此类模型。

17.2.4 Quality measures for identification algorithms

什么是一个好的识别算法? 理想情况下，人们希望算法能够快速学习目标概念（或函数），以便在假设中犯很少或没有错误。对应于上述两种一般类型的对手，已经考虑了两种错误界限的概念:

1. 在有限数量的样本之后完全停止犯错误。2. 随着样本数量的增加，降低错误发生的概率。

两种代表非常不同的学习哲学——一种更偏向逻辑和语言导向，另一种更偏向统计导向。我们将在第17.3节和第17.4节中讨论它们。

一个重要方面是两种方法的快速学习能力。有两个“输入大小”参数对于识别算法的效率至关重要。第一个是序列中单个标记示例的 *length*；这通常是固定的，考虑到领域 X 和范围 Y (，并且通常捕捉到问题的维度，例如，在识别超平面) 时的维度 n。另一个是需要获得高质量目标函数预测器的示例 *number*（即样本大小）。理想情况下，样本数量应该很小，并且算法在两个参数方面都应该是高效的。也许令人惊讶的是，存在一些重要情况，其中样本数量和算法效率之间存在非平凡的权衡，我们将在后面讨论。

17.3 Identification in the limit: A linguistic/recursion theoretic approach

In a nutshell, this direction generally assumes that data arrive adversarially, allows some unspecified but finite “teaching” period, after which the “learner” has to get it perfectly right.

[10] As we will see at the very end of this chapter, adversaries considered in this chapter are too strong, in that very few concepts can be learned if they are unrestricted. And naturally, variants restricting them, as well as other relaxations have been studied as well.

该概念 *inductive inference* 几乎与计算理论一样古老。推动其发展的主要原始研究领域之一是语言学，它从可计算性和递归理论的计算视角以及试图理解自然语言如何演变和学习（由人类和动物）的科学视角中借鉴。关于这一研究方向有出色的综述[AS83]。我们只讨论其基本特征和结果。

The seminal paper, which has shaped this approach, was written by Gold [Gol67]. In this paper, he addresses the modeling issues considered above. In particular, he defines an important notion of success of an algorithm, namely *identification in the limit*. Gold also suggests a (simple) general technique that achieves such success, called *identification through enumeration*, and studies its power. Let me explain and give examples of both. 原始论文，该论文塑造了这种方法，是由Gold [Gol67]撰写的。在这篇论文中，他讨论了上述考虑的建模问题。特别是，他定义了一个算法成功的重要概念，即*identification in the limit*。Gold还提出了一种（简单）通用技术，称为*identification through enumeration*，以实现这种成功，并研究其能力。让我解释并给出两个例子。

identification through enumeration, Gold 考虑这样的技术：输入表示是其能力。让我们描述解释并给出两个例子。

一个函数族 $F \subseteq (X \to Y)$。Gold 考虑这样的技术：输入表示是其能力。让我们描述解释并给出两个例子。

一个函数 $f \in F$，然后选择示例 x_t 出现的顺序，只要 X 的每个元素至少出现一次（如果 X 是有限的，每个元素出现无限多次）。算法观察序列 (x_t, b_t)，并使用 $b_t = f(x_t)$，在每次这样的示例之后，它输出一个假设 $h_t : X \to Y$。

该类 F 在存在一个算法的情况下是 *identifiable in the limit*，该算法对于每一个这样的对手，只会犯一个 *finite* 数量的错误。更精确地说，在有限时间 T 之后，对于所有 $t \geq T$，所有 h^t 都是相同的（即，$h^t = h^T$）并且是正确的（即，$h^T(x_t) = f(x_t)$）。我们强调，算法可能不知道 T 是什么（即，没有要求算法“知道”它何时停止犯错误）。这是这个学习模型的一个明确的（建模）限制，这确实使它非常强大，并允许它识别非常复杂的函数族。

我们现在将讨论三个目标示例，这些目标在极限算法中具有识别性，以了解什么是可学习的（以及代价是什么），以及什么不是。

Example 1: The class $\mathcal{P}$ 让 $F = \mathcal{P}$ 表示可由多项式时间算法计算的二元输入布尔函数类。一个简单的算法用于在极限情况下识别此类，Gold 称其为 *identification through enumeration*。它使用一个子程序，该程序枚举所有多项式时间图灵机（记住，每个都有有限的描述，就像限制多项式运行时间的整数指数一样），即打印出所有这些图灵机的列表 $M_1, M_2, \ldots$（其中一些可能计算相同的函数）。现在，对于每个 t，识别算法选择最小的 n，使得上述列表中的 M_n 与迄今为止看到的所有示例一致（即对于所有 $s \leq t$，$M_n(x_s) = b_s$），并将该 M_n 作为其假设 h_t 输出。为了看到这个算法在极限情况下识别 $\mathcal{P}$，考虑一个任意（隐藏的）$f^* \in \mathcal{P}$，用于标记数据，并让 k 是上述枚举中 M_k 计算出 f^* 的最小整数。很明显，如果选择 M_n 中的 $n \leq k$ 作为假设，算法将在有限数量的示例之后犯错误。此外，一旦 M_k 被选择一次，它将永远被选择。

这个算法技术非常通用。它只需要函数类 F 的两个属性。首先，存在一个算法来枚举 F。其次，F 中的每个函数都是可计算的（以检查与迄今为止的数据的一致性）。Gold 观察，这些属性特别适用于 Chomsky 层次结构中的所有语言类（有限、正则、上下文无关、上下文相关和递归可枚举），因此它们在极限情况下都可以被识别。

所以，通过枚举进行识别非常强大，但同时也应清楚，其背后的识别算法可以任意低效。此外，尽管类$\mathcal{P}$中的所有函数都可以高效计算，因此第二个属性可以高效测试，但在步骤t中，可能需要枚举指数（在t中）数量的机器，才能找到一个一致的机器（留给读者作为练习来检查，它永远不会比这更差——上述算法永远不会在比数据长度指数时间更差的时间内运行）。当然，其他目标类的运行时间可能要大得多。我们的下一个例子表明，在某些情况下，极限学习是可能的。

[11] Recall that a language (a set of sequences) is naturally associated to a function $f : \Sigma^* \to \{0, 1\}$ is the set of sequences f maps to 1.

Example 2: Rational polynomials 让 $X = \mathbb{Q}$ 为有理数域，$F = \{p : \mathbb{Q} \to \mathbb{Q}\}$ 为所有一元多项式的集合。显然，上述枚举算法也适用于此类目标，因为它可枚举，并且其中的每个函数都可以高效地评估。然而，在上述明显的实现中，即使是对于 $\mathcal{P}$ 的这个简单子类，它也将需要指数时间。然而，由于 *interpolation*，人们可以做得更好，因为不需要搜索最小一致假设——一个独特的假设是明确定义的，而且还可以高效地找到它。更精确地说，识别算法的工作原理如下：

从某个零假设开始（例如，$h^0 = 0$）。对于每个后续输入对（x_t, b_t），输出一个假设 h^t，如下所示：

- **Correct classification:** 如果 $b_t = h^{t-1}(x_t)$，则设置 $h^t = h^{t-1}$。
- **Incorrect classification:** 如果 $b_t \neq h^{t-1}(x_t)$，则让 h^t 成为唯一的多项式 $t-1$ 插值第一个 t 个示例。

读者被邀请验证，如果用于生成数据的隐藏多项式 p^* 的次数为 d，那么通过步骤 $T = d + 1$，所有假设都将相同的多项式 p^*，并且算法将在数据长度上以多项式时间运行。对于有限域上的多项式，这也同样成立。[12]

示例中，在极限下进行这种高效识别的情况很少见。现在我们将看到，对于我们所考虑的第一个激励目标类，即超平面，也可以这样做。

Example 3: Real hyperplanes 从第17.1节回顾识别超平面 h^* 的问题。我们为此问题提供了两种算法，感知器算法和线性规划。事实证明，它们在极限情况下都有效地实现了识别，但与多项式上的示例1相比，程度略弱；它们在有限次错误后收敛到正确答案，但这个有限次数取决于一个称为 *margin* 的参数，该参数在许多识别（以及更一般的学习）算法中是共同的。我将在下面正式定义它，但直观上，它捕捉了数据对微小波动的鲁棒性。例如，在用于激发超平面分类的垃圾邮件示例中，我们期望对于任何一对电子邮件，一封合法邮件和一封垃圾邮件，它们之间将存在显著的、明显的距离，并且这个间隔越大，分类器就越有效。我们现在将看到这种直觉是如何发挥作用的。

我们将仅分析用于分类超平面的感知机算法。为了方便，再次回忆该算法。

首先初始化 $h_0 =$ 为 0。在 t 个样本之后，将 h^t 设置如下：

- **Correct classification:** 如果 $b_t =$ 符号$(\langle h^{t-1}, x_t\rangle)$，设置 $h^t = h^{t-1}$。
- **Incorrect classification:** If $b_t \neq \text{sign}(\langle h^{t-1}, x_t\rangle)$, set $h^t = h^{t-1} + b_t x_t / \|x_t\|$.

要分析此算法，定义 $\hat{x} = x/\|x\|$ 为向量 $x \in \mathbb{R}^n$ 的缩放（到单位长度）。此外，引入一个参数 μ，称为 *margin*，它依赖于数据，并测量点 $\hat{x}_i$ 到超平面 h^* 的最小距离。因此，$\mu = \inf_i \langle h^*, x_i\rangle$。请注意，$\mu$ 越大，超平面两侧的点集之间的分离越好。实际上，它们不是通过 h^* 分离的，而是通过一个同方向的条带分离，其宽度为 2μ。边缘确定了一个有限界限（该界限与维度 n 无关）上感知器算法犯的总预测错误数。

Theorem 17.1 [Nov62]. *The total number of incorrect classifications of the perceptron algorithm on a data sequence of margin μ is at most $1/\mu^2$.*

一个有趣的观察是，即使样本数量是有限的（即，序列 $\{x_t\}$ 在 $\mathbb{R}^n$ 中只包含有限个不同的点），通过足够多次地循环它们（取决于边界），感知器算法将找到与数据一致的一个假设。

[12] The reader is invited to contemplate *multivariate* polynomials over the rationals or over finite fields.

Proof (sketch). 这个Novikoff证明背后的简单思想在分析其他更复杂的算法时被多次使用；它基于对比L_1和L_2范数中的进展。直观上，每次错误分类后对假设h^{t-1}进行的修正提高了h^t与真实分离超平面h^*的相关性。然而，h^t并不比h^{t-1}长很多，因为h^{t-1}和x_t之间的角度是钝角，且$\hat{x}_t$是一个单位向量。结合这两个事实将限制错误数量。让我们使这个想法形式化。

我们将从上往下和从下往上界定内积 $C_t = \langle h^t, h^* \rangle$，它在 $t = 0$ 处为 0。假设 h^{t-1} 在 x_t 上的预测是错误的。对于下界，注意

$$\langle h^t, h^* \rangle = \langle (h^{t-1} + b_t \hat{x}_t), h^* \rangle \geq \langle h^{t-1}, h^* \rangle + \mu,$$

然而，在 N 分类错误之后，$C_t \geq \mu N$。

$$(\langle h^t, h^* \rangle)^2 \leq \|h^t\|^2 \leq \|h^{t-1}\|^2 + 1,$$

在本文中，我们使用了柯西-施瓦茨不等式以及 h^* 和 $\hat{x}_t$ 的单位长度，以及 h^{t-1} 和 $\hat{x}_t$ 之间存在钝角的事实。因此，在 N 次错误之后，我们得到 $C_t \leq \sqrt{N}$。结合这些界限，我们得出结论 $N \leq 1/\mu^2$。

□

17.4 Probably, approximately correct (PAC) learning: A statistical approach

In a nutshell, this direction assumes that data is generated randomly, insists on quantitative bounds on the number of labeled examples and on algorithmic efficiency, but allows unlimited prediction errors as long as they occur with low probability.

学习*distribution-free*的概念起源于Vapnik和Chervonenkis的开创性工作[VC15, VC74]，源于统计学习理论和概率论领域，重点关注*sample complexity*。关于这项工作及其起源和应用的全面论述出现在Vapnik的书籍[Vap98,Vap13]中。独立于Vapnik，但稍晚一些，Valiant[Val84b]提出了相同的概念。然而，受理解学习作为一种认知过程以及在任何学习理论中区分可行和不可行的认知任务的需求的启发，Valiant将学习算法的*computational efficiency*方面作为其模型的核心。这个模型被称为*probably, approximately correct*学习（或简称为*PAC*学习）是在[AL88]中提出的。关于这一学习观点的出色直观介绍是Kearns和Vazirani的书籍[KV94b]。[13]

存在许多本质上的差异，在17.3节中采用的归纳推理方法和此处采用的统计方法之间。主要对比在17.3和17.4节开头的单句总结中显而易见。归纳推理的逻辑框架对预测错误毫不宽容。为了最终达到这种完美，它实际上愿意投入任意长的教学阶段。相比之下，统计框架如果预测错误很少，就会原谅它们，并坚持短教学阶段和高效学习。虽然从数学上讲两者都非常有趣，考虑到今天在计算学习方面的理论和实践工作，可以肯定地说，统计方法已经取得了巨大成功。这甚至适用于逻辑方法的原始动机，即理解语言的演化和学习，以及翻译和生成语言文本。也许统计方法这一优势的最强理由是，在自然界中，*inefficient*学习可能比*imperfect*学习更具破坏性（对于生存和繁荣）。此外，几乎所有机器学习的实际产品都是基于这一观点。最后，统计的复杂性理论观点似乎也暗示了更好的模型，可以解释自然学习机制可能如何进化；这一观点在Valiant的书中得到了详细阐述[Val13]。我们回到这个框架中具体的分类任务。

[13]Note that in the literature, "PAC learning" is sometimes taken to mean the original distribution-free learning of Vapnik and Chervonenkis (which disregards computational complexity), and sometimes as the *efficient* version of Valiant. I will stress the efficiency aspect of learning algorithms in the definitions and results below, distinguishing "PAC learning" and "efficient PAC learning."

17.4.1 Basics of the PAC framework

为了简单起见，我们从此限制自己识别布尔函数，即范围 {v*}。这种情况在统计学习理论文献中通常被称为 *pattern matching*，在机器学习社区中被称为 *binary classification*。理论扩展在更大的范围 Y 上被研究，这些范围可以是离散的或连续的（例如，参见上述引用的书籍）。与布尔域不同，在布尔域中，对于 X 中的每个数据点，一个假设要么是正确的，要么是错误的，而在一个点上的假设质量（即它与正确答案有多少不一致）必须被定义，通常使用 Y 上的某些度量，或者更一般地，一个 *loss function*。这种一般设置的研究通常被称为 *empirical risk minimization*，其中“风险”是相对于损失函数来考虑的。

返回布尔范围。固定一个函数的目标类 $F = \{f\colon X \to \{0,1\}\}$，并且进一步在 X 上定义一个任意概率分布 [14] D。在无分布的 PAC 学习中，分类算法（有时称为 *classifier*）的可接受数据是一系列标记示例（$x_t, f^*(x_t)$），其中样本 $x_t \in X$ 根据以下 D 选择，f^* 是我们试图识别的隐藏函数（或 *classify*）。再次，一个算法在观察前 t 个示例后产生一系列假设 h^t。

让我首先直观地解释一下在这个设置中什么是好的算法，然后更精确地定义它。如果一个算法在经过一些T步骤后，输出的假设以高概率预测下一个点上的f^*的值，那么它就是一个好的算法。我们强调，必要的示例数量T并不依赖于底层分布D（，这解释了术语*distribution-free*)。当然，这个样本大小T可以并且会依赖于目标概念类F的性质以及两个错误参数（以下定义的准确性和置信度），这些参数决定了在这么多示例之后的预测质量。

有两个问题要讨论关于分布 D。首先，它是平稳的，在整个过程中不发生变化；如果将这个 D 视为一个为学习者生成经验的 环境，那么我们可以将平稳性视为公平性：学习者在它所学习过的相同类型的经验上接受测试.[15] 这个假设有时被称为 *invariance*，它肯定是一个强有力的假设。甚至更强的假设是 *independence* 的样本。这两个假设都忽略了（或掩盖了）这样一个事实：在自然和实践 中，学习者的假设往往会产生影响环境及其可能生成的未来示例的行动/行为.[16] 然而，这些假设在初始数学模型中是自然的，而且我们还将看到，即使有这些假设，也只有少数几个通用类别是可学习的。一旦我们接受它们，每个样本就像其他任何样本一样好，因此从现在起我们将考虑一个 *training set* (而不是) 的序列T标记的示例，并且一个假设 h 在 D 的单个随机样本上接受测试。有了这个，我们就为 PAC 学习的正式定义做好了准备.[17]

Definition 17.2 [VC15, Val84b]。一个概念类 $F = \{f\colon X \to \{0,1\}\}$ 是 PAC-可学习的，如果存在一个 ，可能是概率性的（学习）算法 A 和一个整数值函数 $T = T(F, \epsilon, \delta)$ 具有以下性质。对于 X 上的每个概率分布 D，以及对于每个函数 $f^* \in F$，在输入 $\epsilon, \delta > 0$（分别，*accuracy* 和 *confidence* 参数）和从 D 标记的 f^* 独立示例中，算法 A 返回一个假设 $h = h^t$，该假设以至少 $1-\delta$ 的概率满足 $D(h \triangle f^*) \le \epsilon$。在这里 $h \triangle f^*$ 是 X 的子集，其中 f^* 和 h 不一致，$D(h \triangle f^*)$ 表示其在 D 下的质量。成功概率是在生成示例的分布 D 上计算的，以及算法 A 的任何抛硬币操作。

算法 A 被称为 *efficient*，如果它在参数 $1/\epsilon, 1/\delta$ 的多项式时间内运行，并且 T 样本的总长度为 [18]。

如果算法 A 限制只从目标类别 F 输出假设，那么这个类别被称为 *proper* PAC-可学习的。

[14] Or measure, in continuous domains X; we will not be concerned here with measurability issues, which are not central to this topic.

[15] As in high school, when students demand that the test contain only questions previously discussed in class. Or, as in the wild, when African lions test their hunting strategies near the same water supply antelopes come to drink at every evening.

[16] Note that this objection does not affect the previous learning notion of “identification in the limit.”

[17] I prefer *identification* to *learning*, but this notion is very common in the literature.

[18] It would stand to reason to also demand that T itself is small in terms of the error parameters, and properties of F. As we shall see, this is guaranteed automatically.

让我们将PAC缩写中“可能”和“大约”这两个词的来源与两个误差参数联系起来。*Probably*与δ相关——算法必须以高概率产生一个好的假设：至少$1-\delta$。一个好的假设是正确的：在从D的随机样本中，h和f^*不一致的概率至多为ϵ。

哪些函数类是PAC可学习的？如果是的话，通过哪些算法？这两个一般性问题有非常清晰的答案，以下将进行动机和解释。

显著地，目标概念类 F 的一个简单的单组合参数决定了它是否是 PAC 可学习的。它被称为 VC-维度，以它的发明者 Vapnik 和 Chevonenkis [VC15] 命名。直观上，它捕捉了当限制到定义域 X 的任何有限元素集合时，函数类 F 的“丰富”程度。对于有限集 $S \subset X$，令 F_S 表示将 F 中的函数限制到 S 的所有限制的集合。这种丰富性 F 是作为 $|S|$ 的函数的 $|F_S|$ 的大小，对于最坏的情况 S。这种丰富性恰好由使得 F_S 最大的最大 S 决定，即 S 被 F 完全“破碎”。形式上，我们称一个集合 $S \subset X$ *shattered* 如果 $|F_S| = 2^{|S|}$，即，在 S 上的每个可能的布尔函数都可以扩展到 F 中的函数。

Definition 17.3. F的VC维数，表示为VC dim(F)，是X中一个被击碎的集合的最大大小。如果不存在这样的最大集合，则定义VC dim(F) $= \infty$。

尝试证明我们在讨论的一些函数类上的以下VC维界限。这样做将阐明定义并证明即使当F很大、无限或甚至不可数时，它也可以具有小的VC维。

1. 平面中轴平行矩形的VC维数为4。更一般地，$\mathbb{R}^n$中所有轴平行矩形的VC维数为$2n$。

2. $\mathbb{R}^n$ 中超平面的VC维数为 $n+$。此外，任何具有边界的超平面集合（参见第17.3节中的示例3）在任何维度)上的VC维数最多为$1/\mu^2$。

3. 在 n 布尔文字上的合取的VC维是 n。4. 任何有限族 F 的VC维最多为 $\log |F|$。特别是，具有大小-s DNF公式的所有布尔函数的类（或者甚至具有大小-s 布尔电路的类）的VC维最多为 s^2。

这个笑点在于这个简单的组合参数，即函数类的VC维，决定了PAC学习性，并产生一个具有最优学习率的最优学习算法（即给定误差参数的必要样本大小）。

Theorem 17.4 [VC15]. *A class F is PAC-learnable if and only if* VC dim(F) *is finite. Moreover, denoting* VC dim(F) $= d$, *the following conditions hold.*

- *The number of required examples is* $T(F, \epsilon, \delta) \leq O(\frac{1}{\epsilon}(d\log\frac{1}{\epsilon} + \log\frac{1}{\delta}))$.
- *Every algorithm that produces as a hypothesis any function in F that is consistent with the sample, achieves the required error bounds.*

让我强调几点。首先，样本大小界限是 *independent*，这是域 X 或函数的目标类 F 的大小。在精确的意义上，VC 维度捕捉了从 *any* 分布在 X 上的采样目的的“本质大小”。请注意，对于常数误差参数 ϵ, δ，样本大小的界限在 F 的 VC 维度中是 *linear*。最后，这个样本大小的上界 T 在所有参数中都是最佳可能的，如 [KPW92] 中所证明的。

这个定理在学习和统计学（它起源于此)以及其他领域，包括离散几何（参见[HW87]）、差异理论和组合数学，特别是集合系统（超图）的研究。它们之间的联系以及定理的基本性质，都源于它的一种稍微抽象的版本，使用了ϵ-网的语言（例如，参见[Mat02，第10章]以了解其阐述）。对于这种观点，最好将F中的函数视为指示函数的

子集 X。对于一个分布 D，一个 ϵ-网是 X 中点的子集，它与 F 中的每个“大”集合相交，即其 D-度至少为 ϵ 的集合。该定理表明，对于任何有限 VC 维度的 F，从 *any* 分布 D 中采样足够多的点将是一个 ϵ-网，以高概率覆盖 D。此外，如果要求更强的概念，ϵ-近似（即与 F 集合的交集大小在 ϵ 以内，该集合的 D-度）也有类似的陈述。利用这些联系，这个定理及其变体为可能无限（甚至连续）的 X 空间和有限 VC 维度的函数族 F 提供了集中界限 *uniform*。

现在让我们谈谈定理的证明。证明的直觉来源于这样一个事实：有限的VC维数 d 允许我们将 F 视为很小，即使它是无限的。首先，让我们看看为什么当 F 很小时，一个小样本就足够了，然后看看VC维数在何种意义上捕捉了这种小性。为了简单起见，假设失败概率 δ 被固定为某个小的常数，比如说，.001。

我们的任务是证明对于每一个 $f^* \in F$，如果我们从 D 中随机抽取 t 个点，并且 t 足够大，那么任何与 f^* 在该样本上一致性的函数 $f \in F$ 将与 f^* 在除了一个 ϵ 测度（在 D 下）之外的所有地方一致，概率为 $\geq 1 - \delta$。考虑任何 f，使得 $D(f \triangle f^*) \geq \epsilon$。显然，样本错过那个对称差（并且无法区分这两个）的概率以指数形式衰减在 ϵt 中，因此如果 $t \gg \frac{1}{\epsilon}$，这个事件将以至多 δ 的概率发生。为了证明有限 F 的定理，我们可以对所有可能 $f^*, f \in F$ 进行并集绑定，并通过 $t \gg \frac{\log |F|}{\epsilon}$ 获得相同的失败概率。

Vapnik和Chervonenkis证明的主要观点是，VC dim(F)可以（大致上）替换上述论证中的$\log |F|$，即使当F是无穷大时。原因是，如果VC dim(F) $= d$，则在每个有限样本大小（例如，t）上，F中的函数最多可以用t^d种不同的方式标记这些t点（与平凡的2^t界限相比）。[19]这似乎足以满足联合界限，因为我们实际上将函数对的数量减少到$t^{O(d)}$，这将被捕捉任何单个大对称差异的exp($-\epsilon t$)衰减所淹没。然而，这个想法并不完全奏效，因为手头的函数和所选样本之间存在一种令人烦恼的依赖关系。这个问题通过一个巧妙的论证得到解决，鼓励读者去发现或了解。[20]我留给你的另一个挑战是证明上述关于VC维度的最基本事实。[21]

Lemma 17.5 [Sau72, She72]。*For any integers $d \leq t$, let $F = \{f : X \to \{0,1\}\}$ be any family of functions with $|X| = t$ and* VC 维度(F) $= d$. *Then*

$$|F| \leq \binom{t}{0} + \binom{t}{1} + \binom{t}{2} + \cdots + \binom{t}{d}.$$

17.4.2 Efficiency and optimization

虽然VC维完全决定了PAC学习性*in principle*，但当然，拥有*efficient*学习算法是至关重要的，正如Valiant在他的原始定义[Val84b]中坚持的那样，因为这些是我们唯一希望实现的算法。我们寻求一个在VC维和输入输出大小参数（X元素的表示，以及从F到函数）方面的有效算法。

现在Vapnik-Chervonenkis定理（定理17.4）告诉我们，任何与足够样本一致的假设都实现了PAC学习性。让我们检查一些具体的目标类别。在17.4.1节中，通过查看列表中的例子，可以简单地看出，对于前三个类别，存在找到一致假设的有效算法.[22]另一方面，证明决定一个集合

[19] This ingredient of the [VC15] proof was independently discovered in combinatorics by Sauer [Sau72] and in logic by Shelah [She72]; I state this simple and useful combinatorial fact more precisely below.

[20] Insufficient hint: Partition the random sample into two random parts of equal size to create the required independence.

[21] Insufficient hint: Use the following two steps. The easy one is proving the bound *if* F is "downward closed": every function whose support is contained in that of a function in F is also in F. The harder one is reducing to this case by iterated "shifting": simultaneously remove some fixed $x \in X$ from the support of every function in F for which this removal defines a function outside F.

[22] For example 1, one can compute the minimal enclosing box of all positive examples by finding the minimum and maximum value in each coordinate. For example 2, one can use linear programming. For example 3, note that every positive example points to n of the $2n$ literals which cannot be part of the hidden conjunction, and can thus be eliminated. Repeating for all positive examples, the hypothesis which is the conjunction of all remaining literals will be consistent with the sample.

正负示例的布尔函数具有大小-s DNF，或大小-s布尔电路，都是-$\mathcal{NP}$-难问题。因此，这种明显的有效学习算法的方法在这些情况下不起作用。

注意，PAC学习等同于（更容易理解的）在数据中搜索一致假设，这非常方便。这个结果自然地将学习任务视为优化问题，并增加了这两个领域之间的许多联系。正如在复杂性理论中通常所做的那样，寻找*approximately*一致假设和其他松弛的任务也是这项研究的一部分。

17.4.3 Agnostic PAC learning

框架的一个明显不足是假设目标类别 F 是已知的（即，我们对我们观察和试图分类的现象有大量的结构信息）。在某些情况下这可能成立，但在许多其他情况下则不成立。如果我们完全放弃这个假设会发生什么? 请注意，与 F 相比，假设类别 H (，即我们选择作为观察到的现象的可能解释的模型集合)，在科学上肯定在我们控制范围内，并且也许在考虑生物系统的学习能力时甚至更容易定义或界定。

定理17.4的一个重要扩展是到一个称为*agnostic learning*的模型，其中只有H存在。在许多上述情况下，当目标类F未知或假设类H不一定包含F时，这是有趣且重要的。在这个模型中，学习者的目标可能是什么？在此之前，我们知道某些函数f^*在F中标记了数据，学习者的目标是使用某些假设$h \in H$来近似它。现在f^*可能是任意的。然而，由于只允许H中的假设，显然最好的期望是做得和最佳函数$h^* \in H$一样好，即与数据相关的最佳函数。学习算法能有多接近这种性能?

实际上，上述定理17.4的证明表明，假设类H的VC维在此通用设置中起着相同的作用，并且相同的算法实现了相同的表现。也就是说，任何与该定理中示例数量一致的假设$h \in H$，在任意分布D下，至少以概率$1-\delta$接近最佳假设h^*，ϵ。事实上，这正是Vapnik和Chervonenkis在原始工作中证明的更一般定理（尽管后来才出现了“无偏见学习”这一术语）。

17.4.4 Compression and Occam's razor

一些认知理论认为，学习涉及以有用的方式总结我们的生活经验，其中重点在于*summary*，与仅仅记住所有这些经验形成对比。同样，科学家在积累了一些现象的数据后，试图将其总结成一个更加紧凑的科学理论。让我们探讨压缩与学习之间可能的联系，以及在PAC学习框架中对其可能的正式化。

威廉·奥卡姆的著名引言，一位14世纪的英国哲学家，是 *“entities should not be multiplied unnecessarily.”* 这成为学习的基本原则，更普遍地是科学发现的原则，被称为 *Occam's razor* 。今天，我们解释这句话的意思是 *“among equally accurate explanations of the same phenomena, always prefer the simplest one.”*[23] 虽然这个原则听起来合理，但它值得一个论证，解释简单解释的好处。事实证明，PAC学习提供了一个框架，其中奥卡姆剃刀可以被 *proved* ，从使用它意味着学习的意义上说。这由Blumer等人[BEHW90]提出，我现在来解释。

仅考虑适当的学习($H = F$)。我们将使用的简单性概念是某些规范编码中函数 f 在 F 中的 *description length* (，记为 $s(f)$。以下定理（大致）表明，如果一个算法总能将大量标记示例压缩到一个 *shorter* 一致假设，那么该算法 PAC 学习 F。

[23] A well-known response to the natural question of why Occam's somewhat obscure statement was interpreted this way is that, well ... it is the simplest possible explanation. Luckily, many other thinkers, some as early as Aristotle, articulated this scientific principle more clearly much earlier.

[24] For example, there are simple binary encodings of various classes of circuits, formulas, finite automata, polynomials, geometric shapes, and so forth.

Theorem 17.6 [BEHW90]. *Assume that for some $\alpha < 1$ and $C > 0$, there is a (compression) algorithm A with the following property: on any sample of m inputs labeled by any function $f^* \in F$, with $m \geq s(f^*)^C$, A produces a function $f \in F$ that is consistent with the sample and $s(f) \leq m^{1-\alpha}$. Then A is a PAC learning algorithm for F.*

证明很简单。它使用指数（在 m 中）衰减的概率，该概率表示在给定的分布下错过空间中显著部分的可能性，并结合对具有如此小描述的可能函数数量的并集界限。直观上，这意味着应该优先选择更简单的、一致的解释，因为如果它确实是不正确的，那么它更有可能被标记为不正确。

这是压缩和学习之间的一种正式单向连接，提出了一个相反方向的问题：可学习性是否意味着在某些概念表述中存在压缩？在他们的论文[LW86]中，Littlestone和Warmuth定义了一个任意类F的参数，我们在这里称之为$Comp(F)$，它捕捉了从F中提取的大规模标记样本的可压缩性。简而言之，如果任何样本（任何大小m）的标记示例可以被压缩到仅包含k个标记示例加上额外的k位大小信息的子样本，并且以允许完全恢复$m-k$“删除”示例的标签的方式，那么$Comp(F) \leq k$。

他们的主要结果是，在这里，压缩也意味着可学习性，因为它简单地限制了VC维度：VC dim(F) $\leq O(Comp(F))$。他们询问是否可以在一个方向上证明任何关系。这花了30年才解决；Moran和Yehudayoff [MY16] 证明了$Comp(F) \leq$ exp(VC dim(F))。因此，一个参数的有限性意味着另一个参数的有限性，并且压缩在这个框架中表征了可学习性！下一个定理总结了这些结果。

Theorem 17.7 [LW86, MY16]. *There is a universal constant $c > 0$ such that the following holds. Let F be any class of functions with* VC dim(F) $= d$ *and $Comp(F) = k$. Then*

$$k/c \leq d \leq \exp(ck).$$

这仍然是一个非常有趣的问题，要关闭这个指数差距。

17.4.5 Boosting: Making weak learners strong

boosting 的想法是机器学习中的核心思想之一，其适用范围超越了它诞生的 PAC 模型。粗略地说，它是一种从仅略好于猜测的规则中创建高度准确预测器的方法。这个想法听起来非常荒谬，尤其是如果你考虑使用你最喜欢的股票经纪人或气象学家，他们可能（比如说）以 51% 的概率猜测一些关于明天的信息（适合他们的专业知识），并将他们的建议随着时间的推移转化为一种方法，让你在相同的预测任务中 99% 的时间都能成功。然而，在相当一般的设置中，这是可能的。这首先在 Schapire [Sch90] 的突破性论文[25] 中在 PAC 模型中得到证明，不久之后由 Freund [Fre90] 简化和改进。随后，Boosting 迅速超越了这一模型和计算学习，并应用于优化、统计学、博弈论等领域。（这些应用已在原始论文中提到，包括对学习算法的空间效率的影响，以及学习任务的第一 $\mathcal{NP}$-hardness 结果。）我们已经在本书前面的两个不同上下文中看到了 Boosting 的体现：第 16.2 节中的在线专家预测和第 8.8 节中的结构与随机性。Boosting 是众多实用系统的一部分，并在科学实验中很有用。这个 Boosting 理论及其科学和实际应用在创始人撰写的优秀书籍 [SF12] 中得到了探讨。在这里，我们只限于 PAC 框架。

我们现在将固定一个目标类 F 和一个假设类 H。让我们回到 PAC 学习的定义。PAC 学习算法的质量由两个参数控制：ϵ，即 *accuracy* 和 δ，以及 *confidence*。我们希望研究提高这些参数质量的可能性。

明确定义PAC学习中的角色，让我们称一个（有效）算法A为(ϵ, δ)-学习器，如果对于每个分布D，在足够多的独立示例上，由某些$f^* \in F$返回标签，以概率$\geq 1-\delta$返回一个ϵ-准确度的假设$h \in H$（，即满足$f^* \triangle h$在D)下的测度最多为ϵ的假设。

[25] Yet again, by a graduate student!

我们定义F为使用来自H的假设PAC可学习的，如果存在一个(ϵ, δ)学习器对于*every*选择$\epsilon > 0$和$\delta > 0$。实际上，在标准定义中，ϵ和δ被作为学习算法所需的输入，分别作为所需精度和置信度参数。然而，假设有人为我们提供了一些固定ϵ_0, δ_0的(ϵ_0, δ_0)学习器A。是否有方法使用A来构建具有更好参数的（有效）学习器？如果是这样，在什么条件下？答案是*yes*和*always*；在可能的最弱参数下进行学习意味着在最强的参数下进行学习。

Theorem 17.8 [Sch90]. *For every $\epsilon_0 < \frac{1}{2}$ and $\delta_0 < 1$, an efficient (ϵ_0, δ_0)-learner for F with H implies efficient PAC learning of F with hypotheses H', where functions in H' are efficiently computable when those in H are.*

本小节剩余部分致力于概述此重要定理的证明。我们现在将论证其两部分：首先，*confidence*的简单放大；其次，*accuracy*的深入且根本重要的放大（即提升）。对于这两部分，我们只关注这些参数，主要忽略对样本大小和计算效率的影响（简单原因是控制这些影响相当常规）。我们指出，这些成本将仅在重要参数与平凡界限$\frac{1}{2} - \epsilon_0$和$1/- \delta_0$之间的差距中多项式下降。

让我们一次处理一个参数。达到任何所需的置信度 $\delta > 0$ 容易实现，从任何 $\delta_0 < 1$ 开始。主要思想是，标记数据不仅可以用来训练分类器，还可以用来测试它们的质量。因此，考虑在独立样本上运行 A k 次，使用 $k = O((\log 1/\delta)/\delta_0)$。这产生了 k 个假设，$h_1, h_2, \ldots, h_k$。现在选择另一个（足够大）的独立样本 S，并在 S 上测试每个 h_i 的质量。最后，输出 h_i 对于 $i \in [k]$，它在 S 上犯了最少的错误。不难看出，这个新算法是一个（有效）(ϵ_0, δ)-学习器，置信度参数 δ 已指定.[26]

关于提高准确度参数怎么样？请注意，这远非微不足道。为了清楚地看到这一点，让我们考虑一个“玩具案例”，其中给定的学习器 A 有信心 1（等价于 $\delta = 0$）。[27] 也就是说，在足够多的由任何 $f^* \in F$ 标记的例子中，这些例子是从任意分布 D 中抽取的，给定的学习器 A 总是返回一个与 f^* 在最多 ϵ_0 的 D 度量上不一致的假设 h。我们究竟如何产生一个比 h 错误更少的假设呢?

答案在于充分利用PAC模型。回想一下，虽然我们希望在分布D下学习f^*，但学习器A在任意其他分布D'下都能保证成功学习它。Schapire的巧妙想法是实际上为A提供从其他（精心选择的）分布中抽取的有标签示例，这将有助于识别h出错的地方并对其进行纠正。你现在可能会问：我们如何从其他分布中进行采样？毕竟，生成示例的分布D不在我们的控制之下。Schapire的回答将是，通过*filtering* (或重新加权)来自D的示例.[28]让我们看看这些想法是如何付诸实践的。

最自然的方法是描述和分析Freund和Schapire [FS95] 的AdaBoost算法，这可能是最优雅且实用的提升算法。[29] 然而，AdaBoost在许多来源中都有很好的描述；此外，其分析非常相似（与第16.2节中描述的密切相关）的在线专家预测算法。因此，在这里，我描述Schapire [Sch90] 的原始提升算法，它不太为人所知。虽然效率较低，但其分析更直观。此外，它与电路复杂性中一个非常不同的放大算法相似，即Valiant [Val84a] 为构造Majority函数的短单调公式而设计的算法。

我将展示如何使用 A，它是一个 ϵ_0 学习器，[30] 来构建一个 ϵ_1 学习器 B，使用 $\epsilon_1 < \epsilon_0$。通过迭代这个构建过程，将得到一系列 ϵ_i 学习器，序列 $\{\epsilon_i\}$ 快速收敛到 0。我们

[26] A sentence starting with these words is often false, as is the case here. But morally it is correct. The slightly uglier correct statement is that for any $\gamma > 0$, the above construction can yield a $(\epsilon_0 + \gamma, \delta)$-learner. Moreover, the dependence of k and $|S|$ on $1/\gamma$ are polynomial, making the construction efficient.

[27] Actually, by the amplification of confidence above, this toy case is essentially as general as the general case, as we can drive δ as close to 0 as we wish. Assuming $\delta = 0$ will release us from accounting for the (minor) way in which this parameter deteriorates when amplifying accuracy.

[28] As possibly corrupt casinos might do when making sure that the roulette ball lands in a random slot except for those clients who bet on it in this round.

[29] The name AdaBoost comes from “adaptive boosting”—when amplifying the accuracy, it does not even need to know the initial accuracy ϵ_0 of the given learner A.

[30] As discussed, we disregard δ, as it is set to 0.

不会分析这个简单的收敛性以及样本量和效率分析，直接进行一步放大大分析。

学习器 B 将如下行为。它将从三个分布（如下指定）D_1, D_2, D_3 中独立抽取 A 个样本，A 对这些样本分别响应输出假设 h_1, h_2, h_3。然后 B 将输出函数 h 作为假设，该函数计算 h_i 的多数（即在每一个域元素 $x \in X$ 上，$h(x) = Maj(h_1(x), h_2(x), h_3(x))$）。[31]

分布如下指定。$D_1 = D$，原始分布。D_2重新平衡D中元素的概率，以便给$\frac{1}{2}$赋予相同权重，对于h_1正确和错误的那些。注意，因为我们有h_1在手，这种重新加权很容易。[32]最后，D_3只输出D中h_1和h_2不一致的标记示例（丢弃所有其他示例）。可以验证，给定D的样本，可以从D_2, D_3中有效地进行采样。

主要定量主张是，如果 A 的准确性满足 $\epsilon_0 \leq \alpha < \frac{1}{2}$，那么 B 的准确性满足 $\epsilon_1 \leq 3\alpha^2 - 2\alpha^3$，这严格小于 α 对于 $\alpha < \frac{1}{2}$。细心的读者可能会注意到这个公式很熟悉：当独立抛掷一个正面向上的概率为 α 的硬币三次时，结果将出现多数正面的概率正好是 $3\alpha^2 - 2\alpha^3$。现在考虑当 x 从 D 中抽取时作为三次抛硬币的 $h_i(x)$；很明显，每个都最多以 α 的概率将 x 错误标记。然而，它们可能远非独立（取决于 A 的行为）。尽管如此，以下简单的论证表明，$h(x)$ 错误标记 x 的最大（最大）概率是它们独立时（即最多为 $3\alpha^2 - 2\alpha^3$）。

为此分析，将域 X 分为 4 部分：$X_{CC}, X_{CW}, X_{WC}, X_{WW}$，其中第一个下标表示 h_1 是否正确，第二个下标表示 h_2 是否正确。分别用 a, b, c, d 表示这四个集合的度量（在 D 下）。现在我们使用 A 在任何分布上的承诺精度，特别是 D_1 和 D_2。[33] 由于 h_1 和 h_2 在各自的分布上最多以 α 的概率是错误的，我们得到以下两个不等式（请验证）：

$$d + b \leq \alpha, \qquad\qquad c/(2(1-\alpha)) + d/(2\alpha) \leq \alpha.$$

使用这两个不等式，我们可以如下界定h在D上出错的概率。要使大多数出错，要么x落在具有测度d)的X_{WW} (中，要么它落在h_3出错的部分$X_{CW} \cup X_{WC}$中（这是该集合最多α的分数）。因此，h出错的概率最多是

$$d + \alpha(b+c) \;\leq\; [\alpha(d+b)] + [(1-\alpha)d + \alpha c] \;\leq \alpha^2 + 2(\alpha^2 - \alpha^3) \;\leq\; 3\alpha^2 - 2\alpha^3.$$

17.4.6 The hardness of PAC learning

因此，哪些布尔函数类可以被有效地PAC学习（就其描述大小而言）？[34] Valiant在他的原始论文中表明，合取类（由文字及其否定组成）可以。这很好，因为我们通常通过某些特征的存在和不存在来对对象（动物、植物、首选城市、住宅、朋友等）进行分类（每个特征都可以用一个布尔变量表示）。但显然，这是一种非常原始的对象描述。假设我们提升一个层次，并要求合取的析取，即DNF公式？令人惊讶的是，这个基本问题已经受到数十年的攻击，但仍未解决（并且普遍认为答案是负面的）。[35]

这里我们讨论如何论证PAC学习任务的难度，以及如何处理这种难度。让我们逐一解决这些问题。

[31] This clarifies that the final hypothesis class H' after iterations will be simple formulas of the functions in the original hypothesis class H.

[32] For instance, for each example, first flip a fair coin to choose "correct"/"wrong," and sample labeled examples from D, taking the first for which h_1 agrees/disagrees with the label.

[33] Note that in D_2 we reweigh every element on which h_1 was correct by $1/(2(1-\alpha))$, and every element on which h_1 was wrong by $1/(2\alpha)$.

[34] For natural classes of Boolean functions (e.g., various circuits, formulas, branching programs), their description size also bounds their computational complexity. Additionally, efficient learning can only be done for succinctly described functions.

[35] You may recall that in Section 17.4.2, we mentioned that computing a DNF formula of a given size consistent with a given sample is $\mathcal{NP}$-hard, but of course, this is only a sufficient condition for PAC learning.

第一个考虑硬度结果的是Kearns和Valiant [KV94a]，他们将以下基本联系到密码学硬度。首先，他们注意到随机函数难以学习（即使在均匀分布下）。毕竟，*by definition*看到随机函数在输入样本上的值，对你了解它在任何其他输入上的值毫无帮助！当然，这本身并不是什么坏消息，因为随机函数无法被简洁地描述。但现在回想一下[GGM86]中的密码学伪随机函数（在密码学和随机性章节中提到），在存在单向函数的假设下构建（参见第4.5节）。这是一个高效计算的函数族（每个函数都有一个多项式大小的电路），但这个族中的随机成员无法与真实随机函数区分开来。因此，它们也无法被学习！所以，第一条信息是 *not everything that is efficiently computable is efficiently learnable.*更普遍地，这确立了以下原则：如果一个函数类包含伪随机函数（在一定的复杂性理论假设下），那么它不能被PAC学习，即使在均匀分布下。Kearns和Valiant [KV94a]给出了此类函数的一些其他例子，而Kharitonov [Kha93]将他们的结果扩展到非常低复杂度类。

但是简短的DNF是极其简单的函数，那么这种方法告诉我们它们的学习能力如何？如果我们只关心在均匀分布下的学习，那么有一个简单的算法可以在准多项式（$n^{O(\log n)}$）时间内学习大小为n的DNF公式.[36]相比之下，由于Klivans和Serveidio [KS01]的最佳已知PAC学习算法（对于任意分布），需要轻微指数（大约exp($O(n^{1/3})$）时间.[37]

PAC学习k-DNFs的硬度首先由Pitt和Valiant [PV88] 考虑，他们证明了对于*proper* PAC学习（即，当假设类本身应该是k-DNF公式时），问题是$\mathcal{NP}$-hard（在随机归约下）。这被扩展到[ABF+08]中DNFs的适当学习的硬度。

但是，适当的学习是对学习者不合理的高要求，实际上，最著名的算法都不是适当的（[KS01, Jac94] 都产生了多项式阈值函数的假设）。只有最近，Daniely 和 Shalev-Shwartz [DSS14] 才在自然且合理研究广泛的平均情况难度假设下，最终证明了通用的 PAC 学习难度：没有任何固定的多项式时间算法可以 *refute* 随机 k-SAT 公式，对于增长的 k (（见论文以获取精确假设及其许多支持结果)）。

虽然我们专注于DNFs，但许多其他非常简单的函数类在PAC模型中已知或疑似难以处理（有时在无偏见或有噪声的变体中）。除了PAC之外，还存在许多其他学习模型（具有查询访问等附加功能或对非布尔标签更宽容的错误度量），在许多情况下都是现实的。在某些这样的模型中，对于这些简单问题中的某些，我们确实有可证明有效的学习算法。然而，这些模型与解释人类、动物以及许多机器学习启发式方法在学习和处理更复杂函数方面的经验成功相去甚远，即使在未知（有时甚至变化的）分布下也是如此。这种二分法，一方面是各种启发式方法在“现实生活”实例上的效率差距，另一方面是有可证明保证的算法，远远超出了学习理论，涉及到算法和优化。构建更好地捕捉现实生活输入和问题的模型仍然是整个计算机科学领域的一个主要挑战。但在机器学习领域，特别是最近在训练“深度网络”以学习高度复杂结构方面取得的进展似乎正在推动该领域接受（当然对于工作系统）成功的启发式方法，即使我们不知道它们在做什么以及如何做。这种趋势对领域性质和社会的影响还有待观察！

因此，以一个具体的数学开放问题结束这次讨论是很好的，我认为对于这个问题应该有一个可证明的结果。这是在非常特殊的均匀分布下的DNF学习的一个非常特殊的情况。这是Blum [BL97] 提出的，它捕捉了学习的一个基本问题：消除无关特征。令F_k为n位上的布尔函数的类别，实际上它们最多依赖于它们的k个输入位。能否从均匀分布中有效地检测出相关（以及无关）位?

[36] And in a somewhat more general model that allows the learner to ask queries, Jackson's celebrated Harmonic Sieve algorithm [Jac94] (which cleverly uses boosting) can learn DNFs under the uniform distribution in polynomial time.

[37] Via a clever reduction to learning hyperplanes in these many dimensions, by showing that DNFs of size n can always be represented as the sign of a real polynomial of degree roughly $O(n^{1/3})$. Incidentally, this degree bound is tight by a result of Minsky and Papert [MP69], an important early text in computational learning theory.

随机标记示例? [38] 注意，对于常量 k，可以通过尝试所有可能的子集来有效地学习 F_k，并且对于 $k \leq \log n$ ，F_k 是线性大小 DNF 公式的子类。

Open Problem 17.9. 找到一个在均匀分布下学习 F_k 的多项式时间算法，对于任何随着 n (增长的 k，或者使用一个自然的复杂度假设) 来论证难度。

An interesting trade-off is proved for this problem in [KRT17]: For $k = \log n$, any learner needs a super-polynomial number of samples, and hence super-polynomial time, assuming it has only linear size memory $O(n)$.

[38] Clearly, any learning algorithm will reveal this partition.

18 Cryptography: Modeling secrets and lies, knowledge and trust

密码学是一个真正庞大的领域。它是其更实用的兄弟领域计算机安全的基础，该领域在全球范围内雇佣了众多计算机专业人士，并是计算机产业的重要组成部分。密码学也与密切相关但又有区别的数字隐私领域直接相关，总是因为明显的社会关注而成为新闻焦点，并且最近已经开发出一种美丽的新理论框架（例如，参见调查[DR14, Vad17]）。关于密码学现代基础的全面文本是Goldreich的两卷本[Gol04]。

我们在这本书中讨论了几次密码学的方面，因为它在计算复杂性中提供了重要的概念和研究方向。我在第4.5节中介绍了基于复杂性的密码学的最基本假设，即单向和门限函数。我们在第7.3节中讨论了伪随机性的密码学起源，在第10.2节中讨论了零知识证明的概念。

在以下概述中，我想更多地关注一些更一般性的方面，特别是密码学情境的复杂数学建模及其一些原则。特别是，我将强调现代密码学背后的公理；建模这些任务和约束的多样性、基本原语和设计范式；随机性和计算不可区分性的关键作用；以及复杂性理论和信息理论观点、技术和结果之间的丰富互动。在这份主要涉及经典内容的丰富材料之后，我将简要描述三个在最近取得了令人兴奋进展的密码学任务。在描述了支撑计算机系统安全的这一详尽而美丽的理论之后，我们回到现实，并以对实际密码系统物理攻击而非数学攻击的讨论作为结论。

18.1 The ambitions of modern cryptography

数千年来，*cryptography* (或*cryptology*)，作为一种人类活动，其唯一目的是这些词在希腊语中的字面意义：*secret communication*.[1]该领域的主要目标是设计允许某些当事人交换私密信息的代码，同时阻止其他人理解它们（从另一端来看，主要目标是破解这些代码）。[2]秘密通信在许多情况下都是一个极其重要的目标。我们不会讨论这个古老领域的丰富历史；两本综合性的通俗书籍是[Sin11, Kah96]。相反，我想指出，秘密通信只是实现和保护隐私和完整性的（一个重要）任务之一。古典密码学的目标集中在这一*one*任务上。

但在众多人类活动中，实现和保护隐私和完整性是一个挑战。你希望没有人能阅读你的秘密日记（或文件）。你希望从你的银行账户中取款，但其他人不应能这样做。你希望投票，但不允许重复投票。你希望与朋友进行私人对话，而其他人不应能理解。你希望在你的竞争对手在1秒后少出1美元之前提交出价。你希望在玩扑克时，其他玩家不能偷看你的手牌或袖子里藏着王牌。此类情况层出不穷。现代基于复杂性的密码学的雄心是解决*all*这样的任务（以及更多）。

必须处理所有这些存在上述紧张关系的复杂情况，人类已经发展出大量物理手段（通常是任务特定的），如扑克牌、密封信封、监督投票站、重型保险箱上的大锁、官方银行钞票和身份证。现代密码学不允许任何物理手段。我们想要实现上述所有内容，只是“个人拥有的对象”是信息比特，而“行动”是计算机之间的数字通信（或者，等价地，人们之间的口头通信）！

尽管（或因为）这些严重限制，上述看似不可能的雄心通过现代密码学理论被发现是可能的。关于“要求月亮”的信心在约十年间演变，从20世纪70年代末到20世纪80年代末，当时任务逐渐变化且难度增加

[1]The literal meaning is “secret writing” for cryptography and “the study of hiding secrets” for cryptology.
[2]Often called *cryptanalysis*, which in Greek means “uncovering the hidden.”

在新的框架中发现了可能的解决方案。应强调，这些发展是在互联网出现后发生的 *before*。此外，这一理论 *enabled* 互联网上的电子商务应用，激励了通过私营公司投资其快速扩张！
Modern cryptography is one great
example of the incredible economic consequences of curiosity-driven, theoretical research.

我将很快阐述现代密码学背后的复杂性理论公理，但首先，让我们对比信息论和复杂性理论的观点。

18.2 Information theory vs. complexity theory: Take 1

让我们回到秘密通信问题（我们将反复回到这个问题）。为它发明的所有解决方案都有一个共同特征：两个通信方秘密地 *shared* 一些信息（有时称为 *key*），窃听者没有这些信息。换句话说，总是假设通信方（以某种方式）解决了一个 *earlier* 秘密通信问题，即那个密钥的秘密通信。这是经典密码学的潜在公理，我们将努力消除。

香农在他刚刚发展的“数学通信理论”（我们今天称之为“信息论”）的最早应用之一中，试图精确地研究这一公理的必要性。他的论文[Sha49a]正式定义了*perfectly secure*秘密通信系统的概念，并证明了在任何这样的系统中，秘密发送一个n位消息都需要预先就一个完美的n位密钥达成一致。也就是说，对于你想要在窃听者面前私下通信的每个比特，你必须以某种方式与他们不在场时通信的一个比特为代价。因此，可以证明，这个公理不能被消除！

或者它可以吗？深入探究香农定理的证明，人们会意识到它依赖于以下完全合理（且看似无害）的信息论观点公理：*all representations of a random variable are equally useful.* 更精确地说，如果 X 是某个集合 S 上的随机变量，并且 $f\colon S \to T$ 是从 S 到另一个集合 T 的任何 *bijection*，那么在信息论的所有目的上，随机变量 $Y = f(X)$ 等价于 X。为了做我们必须做的事情（即消除首先需要私下通信以便稍后私下通信的需求），我们必须去除这个信息论公理。第18.3节描述了基于现代、基于复杂性的密码学的公理，这些公理构成了其基础。但首先，为了说明两种观点之间的差异，考虑以下示例。

修复一个大数 n，并通过均匀独立地选择一对 n-位素数 p 和 q 来定义一个概率分布。考虑以下两个由 $X = (p, q)$ 和 $Y = p \cdot q$ 定义的随机变量 X 和 Y。现在，假设我给你其中一个——你会选择哪一个？从信息论的角度来看，这两个随机变量是等价的：一个的值唯一地定义了另一个的值（通过唯一分解）。所以，这是物有所值，你不必在意你得到哪一个。除非 . . . *time is of the essence* 对你来说，当然，这对每个人来说都是如此。从计算的角度来看，这两个选择之间有很大的差异。Y 可以通过乘法从 X *efficiently* 获得 whereas 要从 Y 获得得到 X 等价于分解，这在目前是无法高效完成的。这种 *asymmetry* 是两种观点之间的基本差异，它使得在信息论上不可能的许多壮举在复杂性理论模型中成为可能。

18.3 The axioms of modern, complexity-based cryptography

现代、基于复杂性的密码学，始于Diffie和Hellman的开创性论文[3] [DH76]，通过引入两个互补公理，使不可能变为可能：

- **Axiom 1:** 所有参与加密协议的参与者都是计算上有限制的算法。

[3]The word “seminal” will appear many times in this chapter; no occurrence is taken lightly. This particular paper, besides the scientific and technological revolution it initiated, is an extremely lucid account of basic issues and ideas and their history—a must-read model of scientific writing.

- **Axiom 2:** 存在一些计算问题无法通过此类算法解决。

在继续之前，让我们回忆一下，信息论密码学和复杂性理论密码学都共享一个共同的公理，可以称之为公理0：即所有参与者都能访问完美的随机性：[4]

- **Axiom 0:** 所有参与加密协议的参与者都可以生成无限数量的独立、无偏的掷硬币。

让我们在使用它们之前讨论这些“公理”。当然，这些公理中没有哪一个像一些熟悉的数学公理那样不言自明.[5] 但基于这些公理的加密系统有十亿用户，他们隐含地信任这些公理——这里涉及的不仅仅是抽象真理，而是你的隐私或财富。

公理0至少是良好定义的。它是概率算法背后的同一概念。虽然对于算法的效率来说，它可能不是必要的（如第7章所述），但对于密码学来说，随机性是必不可少的，因为秘密（以及所有其他类似概念）的概念甚至无法在没有随机性的情况下定义。我们已经在前面章节讨论了随机性，特别是处理它不完美的情况（如我们所假设的）。请注意，在密码学中，这比在算法中更令人担忧（因为它可能影响安全性），但我们不会在此处解决这个问题。假设公理0保持不变。

公理1和2需要更多阐述。它们并未精确定义，实际上可以根据实际环境和应用来选择。对于公理1，这种自由实际上是一种建模能力的来源；它允许根据实际情况对不同的资源（可能因用户而异）施加不同的限制。公理2应与公理1相匹配，因为系统的安全性将建立在这种难以解决的问题假设之上。在选择使用哪种假设之前，让我们首先反思一下为什么我们实际上需要这样一个难以解决的问题假设。当然，拥有无条件的安全证明，不依赖于任何未经证明的难以解决的问题假设将是极好的。但就所有实际设计的加密系统而言，破解它们是一个$\mathcal{NP}$问题，[6]，因此证明它们的安全性将意味着$\mathcal{P} \neq \mathcal{NP}$。[7]

因此，我们必须做出硬度假设，而一个困难的问题就是哪些是可接受的。如果能在$\mathcal{P} \neq \mathcal{NP}$上建立密码学那就太好了，但我们不知道如何做到这一点。像分解难题的硬度这样的具体建议似乎很出色，因为这个选择在数学上很优雅，经受住了几个世纪的攻击，[8]并且与应用程序无关。但分解可能最终会变成一个简单的问题。这个问题，即哪些假设是可接受的，是一个不断争论的主题（尤其是在这个领域产生了大量假设的情况下），并且有优秀的论文提出了可接受性的具体标准，如[Nao03, GW11, GK16]。在本章中，我们关注最常见的选择。具体来说，在公理1中，我们将所有参与者限制在多项式时间内计算，并努力[9]使公理2尽可能通用，就像陷阱门函数的存在一样[10] (见第4.5节)。

18.4 Cryptographic definitions

难以精确、数学地定义密码学任务及其密码协议的难度。确实，要真正理解这一点，必须在这个领域工作，或者至少诚实地深入研究

[4] Relaxation of these axioms, when only “imperfect” randomness is available, was explored in cryptography and more generally in computation. We discussed modeling weak random sources and coping with them in Chapter 9.

[5] However, recall that the axiomatic foundations of mathematics itself, on which the “absolute truth” of theorems is based, have been and are still debated.

[6] Simply because one can guess the randomness and private inputs/keys of all participants.

[7] The optimism of the late 1970s is explicit in Diffie and Hellman’s paper; they cite the recent advances in computational complexity, especially the discovery of $\mathcal{NP}$-completeness, as a central motivation for their complexity-based cryptography, and naturally hope that this field will soon produce intractability results!

[8] Gauss is on record for trying hard, but he certainly was not the first. As mentioned in the prelude (Chapter 2), even Euclid’s GCD algorithm is the result of frustration with the difficulty of factoring when simplifying fractions.

[9] In many cases, initial security proofs for concrete systems rested on stronger assumptions, which were relaxed in later works.

[10] Which is of course implied by the hardness of factoring, but is far more general and thus may survive if factoring proves to be easy.

他们的细节在[Gol04]的不同章节中，例如。确实，上面的广告表明现代密码学可以实现看似不可能的目标，这关键取决于选择正确的定义，这些定义通常很微妙，是在发现更天真、初始的定义中存在细微缺陷之后得出的。我将尝试给你一些这些困难来源的思路。首先，与通常由一个具有完整问题输入的算法执行的计算不同，这里有许多相互作用的参与者，每个参与者都拥有组合输入的一部分。这些算法如何通信？是通过成对通信还是广播机制？坏玩家能否窃听？篡改消息？在小组中串通？所有这些决定对于准确建模所需的应用以及协议的可行性都至关重要。其次，与计算任务不同，在计算任务中，几乎总是希望得到正确答案，而密码学任务则伴随着大量的期望目标。此外，由于一些玩家可能执行协议要求的算法之外的算法，定义这些属性需要对坏玩家的不当行为进行嵌套量词交替。我通过一些密码学任务的几个例子来展示这些问题。所有这些在20世纪80年代都被考虑过，正式定义过，并且被认为是可能的，这比互联网出现还要早！尝试为它们的直观要求提出一些半正式的定义是你们的一个很好的练习。请随意使用公理，我再次强调它们如何改变完美保密的概念：玩家无法从她拥有的信息中计算出的内容对她来说是完美隐藏的。

注意，在涉及许多参与者的任务中，必须指定他们之间的通信网络。在整个过程中，我们假设一个广播模型，其中每个人都可以听到任何人说的话（想象他们身处同一房间，或者实际上是在互联网上）；在这个模型中，最难防范信息泄露。以下是一份自然任务的列表。

- **Secret communication:** Alice 和 Bob，他们可能之前从未见过面，想要以完美的忠实度交换信息，以便 Eve，她可以完全访问他们的通信，无法从中提取有关这些信息的 *any* 信息。换句话说，Alice 和 Bob 应该 *publicly* 创建一种 *private* 语言。[11]

- **Message authentication/digital signature:** 爱丽丝和鲍勃，他们可能之前从未见过面，想要以完美的确定性交换信息，以便伊芙，她可以向他们的通信线路中注入信息，无法冒充他们中的任何一个人。

- **Secret exchange:** Alice有一个秘密 S_A，Bob有一个秘密 S_B。必须有一个协议来确保公平交换：当且仅当Bob学习 S_A 时，Alice学习 S_B，并且如果他们都遵循协议，那么两者都会这样做.[12] 在其他任务（例如，联系签名）中也会出现类似的问题。

- **Millionaires' problem:** 爱丽丝和鲍勃希望找出谁更富有，而不让任何关于他们各自财富的信息泄露给对方。

- **Zero-knowledge proofs:** Alice试图说服Bob，她有一个他们俩都知晓的数学命题的证明。如果这是假的，那么Bob会拒绝他们的对话；如果是真的，Bob会接受它，但除了给定命题的真实性之外，他不会从中得到任何东西。

- **Collective coin flipping:** 一组相互不信任的各方希望共同掷硬币并达成公平的结果，他们中的任何一方都无法操纵。相关问题是就集合中的随机“领导者”达成一致。请注意，不良玩家的子集可能会串通！

- **Elections:** 一组相互不信任的玩家，每个玩家都有二进制偏好，希望计算他们的多数投票（即，1的数量是否多于0的数量），同时保持所有投票的完全隐私（除了由多数结果揭示的内容）。[13]

[11] This "impossible" task is what your computer easily performs with, for example, Amazon's computer, over the public Internet, before sending Amazon your credit card number, so it had better work.

[12] Without the last demand, a perfect protocol would instruct them both to hold their tongues.

[13] For example, if three people participate, I voted 0, and the majority outcome is 1, I automatically learn that the other two voted 1.

- **Mental poker:** 一组相互不信任的各方想要在没有牌或其他物理手段的情况下玩一整局扑克（比如，按照德克萨斯扑克规则），从随机发牌和下注回合到最终决定谁赢了以及赢了多少（不透露任何关于玩家手牌或策略的其他信息）。

如果这还不够，如果代理同时参与不同协议和不同集合的代理，则会添加一层巨大的复杂性，但他们可以自由地在他们的互动中使用所有这些信息！

尽管有如此众多的任务和对手，但已经发展出一种优雅的理论，该理论允许为几乎所有合理的加密任务提供形式化的安全性定义以及满足这些定义的协议设计。让我们逐步描述一些导致这种理解的关键思想。 reductions 和 completeness 将发挥重要作用这一点并不令人惊讶。我们从密码学最基本的问题——秘密通信——开始讨论，我们将对此进行一些详细的讨论。

18.5 Probabilistic encryption

没有比Goldwasser和Micali的开创性论文[GM84]“概率加密”更好的地方开始这个解释了。这篇论文首次引入了加密安全的精确数学定义，并在困难性假设下设计了满足这些定义的协议。这篇论文中的语言、框架、思想和技巧（以及坚持最严格的安全定义）成为了随后数以万计（绝非夸张！）加密论文的典范。

本文旨在解决上述列表中的第一个密码学任务，*secret communication*。解决此任务的基础思想，即迪菲和赫尔曼*public-key encryption* [DH76]及其基于RSA [RSA78]和分解[Rab79]函数难度的实例化，已经为人所知。这些方案为爱丽丝提供了一种加密她的消息的方法，以便鲍勃可以解密它们（反之亦然），而伊芙解密这些消息的能力则意味着存在一个针对假设的难题的有效算法（这是不可能的）。但是，正如戈德沃斯和米卡利在他们批判性分析中指出，这些加密方案既不能防止伊芙对*complete*消息的*some*解密，也不能防止伊芙从它们的加密中获得关于*every*消息的大量*partial*信息。本文旨在让伊芙绝对一无所知！如何定义这一点？他们论文的开头最清楚地说明了这一点：

> “本文提出了一种具有以下性质的加密方案：
> *Whatever is efficiently computable about the cleartext given the ciphertext, is also efficiently computable without the ciphertext.*”

此属性在论文中以名称 *semantic security* 进行形式化，并被建议为香农信息论概念 *perfect security* 的计算类比，该概念在他的关于信息论密码学的开创性论文 [Sha49a] 中被提及。简而言之，语义安全要求对于任何有限消息空间上的任何可能分布，以及该空间上的每个函数 f，以下条件成立：在看到随机消息 m 的加密后，Eve 计算出 $f(m)$ 的 *a posteriori* 概率与在原始分布上 $f(m)$ 的 *a priori* 概率基本相同。

当然，在计算设置中，这在上述公理1和2下成立。相比之下，香农证明了在信息论设置中，完美安全意味着Alice和Bob共享信息（因此，在这个设置中，按照定义的任务是不可能的）。实际上，香农使用他最近发展的信息理论证明了，Alice和Bob必须共享一个长度至少是底层消息分布熵的随机序列（这个量也足够了，因为双方可以使用所谓的*one-time pad*）。在计算设置中，不需要共享信息来实现完美安全，正如论文[GM84]所证明的。

本文还提出了另一种安全性的概念，称为*polynomial security*，这更容易进行论证，并且已被证明与语义安全性等价。这里没有对消息进行分布

空间和无需担心任何辅助函数。简单来说，对于*every*两个可能的消息，已知于Eve(!)，她无法以显著优于随机猜测的概率区分它们的加密。

它很好反思这些极其严格的安全要求实际上如何迫使Goldwasser和Micali的加密方案在本质上与所有过去的加密方案截然不同，即成为*probabilistic*，正如论文的标题所宣称的。[14]考虑一个只包含两种可能消息（比如说，0和1）的消息空间。由于Eve看到Alice和Bob之间的所有通信，她可以将Alice的算法应用于0和1，就像Alice自己一样。[15]如果加密是明文消息的确定性函数，就像过去的加密系统一样，那么Eve应该没有问题区分这两种加密。

Goldwasser和Micali的关键思想是，这些加密将随机选择，来自指数级大的空间！在这种情况下，Eve的策略毫无用处。接下来需要论证Bob如何能够识别出信息。这个第一个完全安全的公钥加密方案（省略了许多重要细节）背后的思想很简单。与RSA和Rabin的方案一样，Bob选择两个随机的n-位素数[16]（作为他的私钥）并将它们的乘积M（（在这个系统中是公钥）发送给Alice。为了加密0，Alice向Bob发送一个模M的随机二次剩余，而为了加密1，她向Bob发送一个模M的二次非剩余。Bob可用的M分解使他能够轻松地确定哪种情况，但没有这种*trap-door*信息，Eve无法以比随机猜测更高的概率区分这两种分布（尽管它们有不相交的支持，因此在信息论上很容易区分）。这就是Goldwasser-Micali对公理2的实例化，但后来的论文展示了如何将此类安全的公钥加密方案建立在任何门限函数的基础上。请注意，QRA是关于*computational indistinguishability*的陈述，这在第7.3节中进行了详细讨论，这成为密码学安全定义和证明的基石.[17]

请注意，尽管名为“完美安全”，但它可能并不总是足够（或需要更具体地定义），因为某些应用可能需要更强的概念或额外的属性。这是推动该领域发展的一个因素，要求新的定义和协议。随后开发以应对进一步约束的更强安全定义（以及实现这些定义的公钥加密方案）的例子包括 *deniable encryption* [CDNO97]、*nonmalleable encryption* [DDN03] 和抵抗 *chosen ciphertext attacks* [NY90] 的加密，其名称暗示了这些额外的属性。

18.6 Basic paradigms for security definitions

在关于秘密通信问题及其设定的安全定义标准的长时间讨论之后，让我们考虑第18.4节中提到的其他任务，并开始得出一些一般性结论。这些结论将引导我们到达（相关的）*simulation paradigm*和*ideal functionality paradigm*，它们的组合对于安全定义和证明至关重要。这些范例在20世纪80年代发展起来，现在对密码学家来说已是第二本能。关于这个主题的最近一篇优秀综述是[林17]。

18.6.1 The simulation paradigm

让我们回顾、形式化，然后极大地推广Goldwasser和Micali [GM84] 对其加密方案安全性的非正式表述方式：*Whatever is efficiently computable given the ciphertext is also efficiently computable without the ciphertext*。他们的协议（如上所述）指定了Alice和Bob之间的对话，如果她想要向他发送一个秘密消息（比如说，0或1）。请注意，安全性

[14]Note that while randomness is of course an essential part in many past encryption schemes (in particular, all those mentioned above), it was used only for *key generation*. Once a key is chosen, the encryption of any message using that key is *deterministic*.

[15]This uses another basic principle, already codified by Kerchoffs in 1881 (see, e.g., [Kah74, p. 235]), which states that *the compromise of a cryptographic system should cause no inconvenience to the correspondents*. In plain (or modern) words, this means that for security analysis, each participant of a cryptographic protocol can be assumed to know the algorithms used by all others (but not their private inputs or random coin tosses).

[16]n is called the *security parameter* of the scheme, an auxiliary input, according to which computational complexity is measured.

[17]Recall that, not at all coincidentally, the seminal work on computational pseudo-randomness [BM84, Yao82b] was done at the same time and place—University of California, Berkeley, where Goldwasser and Micali were graduate students.

声明不仅要求Eve不能从对话中学习Alice的秘密信息，也不能推断出Bob的私钥。它要求她在对话之后能做的事情，没有对话她也能做到。我真的是指*everything!*Alice和Bob之间的对话，除了完全保护他们的秘密外，不能帮助Eve解决任何可能无关的问题，比如分解整数$M+2$或找到爱情。如何确保这样的强属性？正如经常发生的那样，证明一般性陈述可能比具体陈述更容易，也更具有说服力。以下肯定可以确保这一点：让我们要求Eve生成Alice和Bob之间的对话*by herself*，而不听他们说话。如果她能做到这一点，那么听他们说话当然不会教给她任何她不知道的东西！

这个要求从Eve那里现在的意义是解决一切问题的关键。首先，回忆一下这个对话是概率性的。也就是说，在二进制序列上存在一个概率分布（比如说，D），它由Alice和Bob的秘密输入和随机性定义，他们之间的实际对话是这个分布的一个随机样本。因此，第一次尝试是要求Eve自己从D中采样。这当然可以，但当然是不可能的（至少和在没有任何信息的情况下猜测Alice的消息位一样困难）。[18]接下来最好的办法（在我们的由公理1设定的计算世界中，这是等效的）是要求Eve从任何分布D'中采样，这个分布是*computationally indistinguishable*从D的。回顾第7.3节关于伪随机性的内容，D和D'在计算上是不可区分的，这里用$D \approx D'$表示，如果没有任何有效的算法能够以比随机猜测更好的概率将它们区分开来.[19]现在这很简单，因为Eve不需要Alice或Bob的秘密来完成这个任务。她可以自己选择两个随机的n位素数，将它们相乘以创建M'，对话的第一部分（到目前为止与实际对话的这一部分在分布上相同）。现在怎么办？很简单。她选择一个随机的二次剩余作为第二部分。也就是说，她只是像Alice一样编码一个0。如果Alice的秘密消息是0，这将很好（实际上，分布相同）到实际对话。但如果它是1呢？没问题！[GM84]协议背后的不可解假设（公理2的选择）正是这两个分布，一个随机二次剩余和一个随机二次非剩余，在两个随机素数的乘积上，在计算上是不可区分的。因此$D \approx D'$。

让我们现在解决一个微妙的问题。我们如何“要求”Eve做任何事情？Eve是一种未知的算法，它肯定不会迎合我们的愿望。这个要求在形式上翻译成以下要求，解释了单词*simulation*。安全要求是对于每个有效的Eve算法，存在一个有效的算法（模拟器），我们称之为Eve′，可以生成D'。这种模拟构成了从加密方案的安全性到困难性假设的降低[20]。具体来说，通过困难性假设，Eve可以用D做到的事情，Eve′也可以用D'做到，而不需要任何对对话的访问。这正是承诺“给定密文，任何有效可计算的，在没有密文的情况下也是有效可计算的”的含义。

秘密通信的任务从几个角度来看都很简单。首先，只有一个（已识别的）可能的对手，即Eve，她不是协议的真正参与者，也没有与之相关的输入或信息。其次，我们已经得到了一个协议，因此很明显要求模拟器做什么；然而，加密任务通常在解决之前就已经定义了。为了处理更复杂情况中的这些问题，我们转向下一个范式。

18.6.2 The ideal functionality paradigm

让我们考虑列表中18.4节的一些其他任务，看看在每种情况下对模拟器进行模拟的自然要求是什么（关注本质，忽略许多细节）。然后我们再次将其推广到一个范例。对于每个任务，我们在讨论模拟之前先回忆其非正式定义。

[18]This follows from the fact that Alice's encryptions of 0 and 1 have disjoint supports.

[19]Again, the most common choices define "efficient" to mean polynomial time in the security parameter n, and "nonnegligible" to mean vanishing faster than inverse polynomial in n. But other choices work as well.

[20]An important issue we will not get into here is how this reduction operates: in most reductions, the simulator Eve′ uses the algorithm Eve in a *black-box* fashion. This was believed to be completely general, until Barak [Bar01] found an ingenious non-black-box reduction. Such simulations allowed proving much stronger theorems in various settings.

- **Zero-knowledge proofs:** (参见第10.2节)。Alice试图说服Bob她有一个证明一个双方都知道的数学陈述。如果这是假的，那么Bob将拒绝他们的对话；如果是真的，Bob将接受它，但除了给定陈述的真实性之外，他不会从中得到任何东西。

  让我们只处理这个任务的零知识方面。（也应该处理这是一个证明的事实，特别是它的正确性：Alice不应该能够欺骗Bob相信一个错误陈述。）零知识要求如果陈述是真的，Bob从他们之间的对话中除了那个真相之外不会学到任何东西。那意味着什么呢？和以前一样，我们要求Bob，给定一个真实的数学陈述（比如说，s），会生成，*by himself*，他和Alice在共同输入s (上会有的对话，这应该是对那个真相的令人信服的论证)。更精确地说，安全性要求对于每个有效的Bob[21]，都有一个有效的模拟器Bob′，对于每个这样的输入s，可以从一个与Alice和Bob在s上会有相同的分布计算上不可区分的分布中进行采样。

  注意，本案例与秘密通信示例有两个区别。首先，Bob有一个输入；当然，Bob′也有权访问该输入。其次，Bob被允许了解一些信息，即声明s是真实的（当它是的时候）。这通过只要求存在这样的输入的模拟器来处理。另一个问题涉及到以下典型的混淆。为什么我们要求为Bob的$every$算法提供模拟器？毕竟，Bob现在是一个参与者，他的行为由协议决定（与上面的Eve不同，Eve是一个“自由代理人”）。答案是，安全要求应该考虑到所有可能的情况，特别是玩家不遵循协议的可能性。请注意，Bob比上面的Eve拥有更多的权力。他实际上与Alice进行通信。因此，如果Bob不好，他可能会决定忽略协议并发送可能导致Alice泄露信息的信息，无论是关于证明的信息还是关于Bob最初不知道而现在想了解的任何其他信息。与之前一样，要求为Bob的每个算法提供模拟器确保这种偏差不会违反零知识属性。

  最后，请注意，我们仅完成了零知识证明的形式安全定义。如何构建一个满足这一看似不可能的任务的协议是另一个完全不同的挑战。在[GMW91]中描述了一个为$\mathcal{NP}$中的数学陈述$every$提供零知识证明的协议，并在第10.2节中进行了讨论。

- **Millionaires' problem:** 爱丽丝和鲍勃希望找出谁更富有，而不让关于他们各自财富的any其他信息泄露给对方。

  我们正在寻找一个交互式协议，其中Alice和Bob在任何（各自的）输入x和y(说，整数在1和M)之间，学习$GT(x, y)$ (一个定义为当x大于y时为1，否则为0的位)，但不再学习其他内容。安全性定义在这里再次类似。让我们明确说明。在这里，每个玩家都应该有一个模拟器。对于每个高效的Alice，我们希望有一个高效的模拟器Alice′，它执行以下操作。在任何输入x和一个位b上，Alice′从一个分布中进行采样，该分布与Alice与Bob在任何满足$GT(x, y) = b$的y上的对话在计算上不可区分。对Bob的模拟器的要求类似。再次，一个协议将指定Alice和Bob的算法。但是，通过要求每个可能的可能“冒充”其中一方的有效算法的模拟器，我们即使在“假的”Bob或Alice偏离协议的情况下也能实现安全性。我们注意到，在这个任务（以及许多其他任务）中，一个有趣的问题（我们在这里忽略了），是如何定义安全性（甚至输出值），如果一方在完成之前终止通信，即在与$GT(x, y)$计算之前。

  如同之前，这仅仅是一个安全定义，找到一个满足这个看似不可能的任务的协议是一个非平凡的挑战，即使对于完全诚实的玩家也是如此！想想看。Yao在[Yao86]中给出了这个任务的协议，我们很快就会讨论它。

- **Elections:** 一组相互不信任的玩家，每个玩家都有二进制偏好，想要计算他们的多数投票（即，1的数量是否多于0的数量），同时保持所有投票的完全隐私（除了由多数结果本身揭示的内容）。

[21] Namely, someone interacting with Alice as Bob, who may not follow the protocol.

这变得越来越复杂，所以我们将会忽略许多问题，比如防止参与者重复投票。我们关注隐私：在多数值对所有参与者公开后，投票的隐私应该具有什么形式意义？例如，假设有五个参与者，其中两个投票0，三个投票1，因此多数值是1。让我们考虑一些场景，以得出隐私的合理定义。假设两个投票0的参与者在协议结束时聚在一起，并互相透露他们的投票。他们立即得出结论，在完美知识下，其他每个投票者必须都投了1。这违反了隐私要求吗？显然没有，因为这即使在理想世界中也会发生，在理想世界中，一个受信任的第三方（每个玩家都有一个安全的私人通道）收集所有选票，计算多数值，并向所有人宣布。相比之下，如果两个投票1的参与者在彼此之间透露了他们的投票，他们所能得出的结论只是，在剩下的三个参与者中至少还有一个1的投票（他们甚至在这个理想世界中也会了解到这一点），但必须不知道更多。即使只有五个玩家，也有许多这样的场景，更不用说更多的玩家了。我们应该在选举协议的隐私要求中列出所有这些可能的场景吗？答案是：不！

存在一种非常简洁的方式来描述所有这些条件，这是由上述假设的受信任方提出的。
*ideal functionality paradigm*断言，在真实协议（其中没有受信任方）中的隐私应该努力匹配理想协议（其中受信任方与每个玩家私下通信并为他们解决问题）中的隐私。对于选举，这推广了Goldwasser-Micali关于秘密通信的最大化到以下安全要求：
What a subset of players learns by participating in the election protocol, it could learn from participating in an election with a trusted party。与原文一样，这个陈述必须对所有可能的玩家输入成立。因此，现在对于任何固定的玩家子集，类似的安全定义如下：对于这些玩家运行的任何有效算法集，存在一个有效的模拟器，该模拟器对于该集合的每个输入值以及给定的多数结果（假设来自受信任方），从（与实际协议记录不可区分的）分布中进行采样。在[GMW87]中开发了一种实现这种安全性的协议，对于所有包含不到所有玩家一半的子集的协议，[22]将被讨论。

我希望您在这里看到一种模式形成——这个范例巧妙且完美地捕捉了我们迄今为止看到的（更简单）其他例子。零知识证明的安全需求是，如果可信方收到了Alice为陈述s提供的所谓证明，检查了它，并告诉Bob这构成有效证明还是不构成有效证明。同样，百万富翁问题的安全需求是，如果Alice和Bob分别向可信方发送x和y，可信方计算$GT(x, y)$并向双方宣布答案。

- **Mental poker**一组相互不信任的各方想要玩一整场扑克牌游戏（例如，按照德克萨斯扑克规则），没有牌或其他物理手段，从随机发牌和下注回合到最终决定谁赢了以及赢了多少（不透露任何关于玩家手牌或策略的其他信息）。

  我们在这里忽略了许多复杂性和细节，并关注高级信息。这个加密任务比投票复杂得多。尽管如此，理想的功能范式应指导我们制定安全定义。我们只需想象一个受信任的实体，它有一个安全通道连接到每个正在按照游戏规则运行的人。根据这个规范，实际协议（对于任何特定的玩家子集）的安全要求是，他们通过参与实际协议所学到的一切，他们已经在那个理想实现中学会了。再次强调，要求为该子集提供模拟器将保证该要求。两方扑克协议来自[GM82, Yao86]，对于任何数量的玩家在[GMW87]。

总结我们所讨论的内容，理想的功能范式与模拟范式共同构成一个极为方便的框架，用于形式化最一般的复杂安全定义。

[22] In a certain sense, this is the best one can hope for.

密码学任务。现在我们已经解决了这个核心问题，让我们转向设计满足这些安全定义的协议的挑战。

18.7 Secure multi-party computation

在20世纪70年代末和80年代初，相继设计了多种不同的密码学任务协议后，姚建议一个单一的任务可能捕捉到其中大部分。他在[Yao82a]中为双玩家任务进行了公式化，并在[Yao86]中描述了相应的巧妙协议。Goldreich、Micali和Wigderson在[GMW87]中很快将这一结果扩展到任意数量的玩家。姚提出的任务被称为“安全多方计算”（SMC），[23]定义如下。

- **Secure multi-party computation:** 一些玩家希望计算他们私人输入上的公共函数 f。他们都知道该函数作为一个布尔电路[24]的显式描述，但电路的每个输入位只被其中一人（玩家可以拥有多个位）所知。这些输入上 f 的值，但其他什么都没有，应该对所有玩家可用。

它完全清楚，到目前为止本章中的大多数例子都符合这种描述（实际上，百万富翁问题和选举被明确定义为函数评估：*GT*和多数）。其他例子只需要简单的修改。例如，集体抛硬币和其他概率函数可以通过允许电路具有随机输入（玩家将不得不共同生成）来建模。还有一些任务，如扑克牌例子和其他“心理游戏”，由于它们的交互性质，需要对SMC进行更实质性的修改。还可以容纳自然扩展，即不同的玩家对相同的输入计算不同的函数（如秘密交换问题中那样）。所有这些扩展都属于SMC，但为了简单起见，我们只会提到上面给出的确定性、静态、单输出定义。

极端通用的SMC问题，几乎捕捉到任何可想象的加密任务，有一个*secure*协议，其中安全性是通过使用上述通用范例来定义的。也就是说，玩家所经历的是将他们所有的输入发送到一个受信任的方，该方评估电路并将正确答案发送回他们。在真实协议中但不在理想协议中可能出现的唯一行为是某些玩家的早期中止（或失败），这可能导致其他玩家永远看不到输出。当一名玩家中止（或更普遍地，如果一半的玩家中止）时，在2人协议中这是无法避免的。但我们将看到，如果大多数玩家遵循协议，我们可以实现*fairness*，即保证将输出交付给所有没有中止的玩家。

该通用协议背后的不可解性假设是最 *standard* 难度假设：陷门函数的存在（与用于安全通信协议的相同，其能力相当接近后者）。[25] 因此，本质上，*if the most basic cryptographic task (namely, secret communication) has a secure protocol, so does the very general SMC task* (我们将回到这个重要观点)。

它甚至更好：*protocol design becomes completely automated*。存在一个有效的算法，当给定要由各方评估的电路作为输入时，输出安全协议的描述（即，参与者算法的规范）。我在这里陈述这个完备性定理的非正式版本。让我们区分（如常见）*fair*安全协议（其中保证将输出交付给所有参与者）和仅仅只是安全协议（其中某些玩家的早期终止可能导致其他玩家永远无法学习输出值。）

Theorem 18.1 [姚86, GMW87]. *Assuming trap-door functions exist:*

[23]Sometimes also called and abbreviated differently in the literature, for example, “secure function evaluation” (SFE) and “multi-party computation” (MPC).

[24]Recall that Boolean circuits are a universal model of computation, as discussed in Section 5.2. Thus, any efficiently computable function can be encoded by a small circuit.

[25]The relationship between the trap-door function primitive and the secure communication primitive is actually quite complex. The advanced reader may want to consult [GKM+00, HO13] on the interrelationship of these and many other cryptographic primitives.

- *SMC has a secure protocol for any number of players* $n \geq 2$.
- *SMC has a* 公平 *secure protocol for any number of players* $n \geq 3$ *, provided a strict majority of the players follows the protocol.*

我现在非正式地、非常高层次地概述了一些设计此类一般问题协议的关键思想。请注意，完整的证明非常复杂和微妙。原始论文缺少许多细节；完整细节可以在Goldreich的第二卷书的第7章中找到。

证明被分为两个概念上独立的部分。为了解释它们，让我们讨论不良玩家可能如何不当行为。在上面的安全定义中，我们没有对对手（除了根据公理1的效率）施加任何限制；每个可能的对手（代表参与者或参与者的子集）都需要一个模拟器。这种对抗性行为在文献中被称为$malicious$，而我们希望对这些最一般的对手提供安全性。一种更为良性的对抗性行为被称为$semi\text{-}honest$。像$honest$玩家（好人）一样，他们逐字遵循协议规范，半诚实玩家也是如此，每个玩家被允许的唯一错误仅仅是崩溃，即停止在协议执行过程中的某个任意点（这可能取决于执行）。直观上，这样的对手应该更容易处理，但请注意，即使玩家完全诚实，上述一些问题看起来也是不可能的（例如，百万富翁问题，零知识证明）。证明的两个部分展示了:

1. 如何设计一个针对（任何数量的）半诚实敌手的SMC协议，以及2. 针对半诚实敌手的$every$协议可以增强以再次成为针对

t

恶意对手，只要后者是少数，就可以变得公平。

两部分的美在于它们都使用了计算复杂性的主要原则，即完备性、归约和计算的局部性。两部分都使用局部性来识别一个简单的$complete$组件（它本身是相同问题的一个非常特殊的情况），为这个特殊情况设计一个协议，然后使用归约（或组合）将其扩展到一般情况。请注意，所有这些都是可能的，部分是因为SMC问题足够通用——对于仅适用于SMC特殊情况（例如，百万富翁问题）的协议之间的归约可能要困难得多。我现在更精确地陈述这两个部分的结果，并概述证明每个部分的想法。

关于加密的说明。两部分都依赖于（不同的）玩家个体输入的加密，这些加密具有特殊性质。我不会指定它们的性质或如何实现它们，只指出这两种类型都可以很容易地从标准陷阱门函数构建出来。我将简单地假设它们存在，并解释它们是如何被使用的。

从第（2）部分开始，这部分在[GMW91]中SMC结果之前就已经得到了。回忆一下，这是第10.2节中提到的论文。这篇论文的一个主要结果是每个$\mathcal{NP}$-陈述都有一个零知识证明。在同样的论文中，他们将这个想法应用于简化协议设计。

Theorem 18.2 [GMW91]. *Every protocol secure against semi-honest adversaries can be enhanced to become secure against malicious adversaries, and it can be made fair as long as the latter are a minority.*

我们将定理简化为一串零知识证明。让我详细说明一下。我们如何迫使一个可能恶意的一方，比如Alice，遵循协议？只需验证Alice发送的每条消息都是根据她（公开已知）指定的算法计算得出的。嗯，该算法指示当前消息应该是一个（公开已知）可高效计算的功能，比如g，应用于一些公开已知的数据x (，例如协议执行中的过去消息)和只有Alice知道的私有数据y（比如她的输入和私有随机数）。因此，Alice发送一些消息m，其他玩家希望验证$m = g(x, y)$，而不让Alice透露y。假设每个玩家在协议开始时公布其私有数据的一个$commitment$（例如，使用其自己的公钥对其加密）。因此，所有玩家都可以访问$z = c(y)$，Alice对其私有数据的承诺y (。

函数 c 当然也是公开的。但现在断言 $m = g(x, y)$ 变成了 $\mathcal{NP}$-陈述！$\mathcal{NP}$-证人仅仅是 y 本身（包括 Alice 的私有随机数）；如果能够访问 y (和公开可用的 x)，就可以有效地验证 $z = c(y)$ 和 $m = g(x, y)$。这意味着根据 [GMW91]，也存在这些方程的零知识证明，Alice 现在可以作为证明者进行，其他玩家作为验证者。这当然需要很多阐述，这里将不提供（例如，其他玩家如何协调，如何避免恶意玩家的无限递归）。

注意，在上面的第（2）部分中，“计算局部性”仅仅是协议执行只是一系列消息的事实。我们现在转向第（1）部分，其中计算局部性将指代每个电路评估都是布尔门评估的序列。

Theorem 18.3 [姚86，GMW87]。
There exists an SMC protocol secure against (any number of) semi-honest adversaries, which is fair as long as a majority of them do not fail before the end of the protocol.

姚的2万SMC证明将任意电路的协议设计简化为单门协议的设计.[26] 关键洞察是在隐藏输入上对门评估的定义，使其与评估任意电路相组合。这个想法很自然：简单地要求输出也隐藏。主要问题是不同输入属于不同的参与者，并且对其他人隐藏。因此，需要以对所有输入和输出的所有门都一致的方式定义“隐藏”。然后，如果玩家可以通过这种方式“在黑暗中”评估一个门，他们就可以迭代这个过程并评估任何电路。姚描述了这样一个巧妙的加密方案（我将不解释），其中门可以由单个玩家评估。第二个玩家只需以这种格式加密其输入，这使用Rabin的 *oblivious transfer* 协议 [Rab81, EGL85] 完成。对于多方情况，[GMW87] 使用分布式加密，其中隐藏值 *shared* 在所有玩家之间共享，他们还进行逐门评估 *jointly*。这些分布式加密的联合构造是通过Shamir的秘密共享方案 [Sha79b] 实现的。实际上，由于 [CGMA85]，我们需要 *verifiable* 变体的秘密共享来防止提前终止。现在，这些加密份额的逐门评估可以由任何多数玩家执行。最后，在这两种协议中，也存在一种方式让（任何多数的）玩家解密电路的最终输出并将其提供给所有人.[27]

让我们通过指出所讨论的通用定理远非最后一言来总结SMC的故事。一个重要扩展[CDD+99]是证明它们对于一个观察协议执行并能根据它腐蚀不同玩家的人来说是安全的*adaptive adversary* (。另一个重要扩展[Can01]，通过通用可组合性范式（该范式适用于SMC之外的协议设计），是在协议不是独立执行而是在与其他协议（相关或不相关）并发执行时证明它们。

18.8 Information theory vs. complexity theory: Take 2

大多数本章内容都强调了这样一个观点：在信息论设置（当不对玩家的计算能力施加限制时）中证明为不可能的任务，在复杂性理论设置（当它们受到如此限制时）中变得可能。现在，我强调一个令人惊讶的教训，即应该根据复杂性理论协议的力量重新思考信息论设置的看似弱点。一个闪亮的例子是安全多方计算（SMC）。回想一下，我们从SMC可以基于门限函数这一事实中得出的非正式结论：执行秘密通信的基本任务意味着能够执行极其通用的SMC。现在可以抽象地考虑这个想法。假设我们每对参与者之间都有完美的通信渠道。这个假设保证了*information theoretically*，即秘密通信的基本任务。在这个模型中，还有哪些任务可以在信息论安全下执行？答案竟然是，就像在计算设置中一样，几乎所有任务都可以！在SMC的计算论文之后不久，就在这个信息论设置中设计了SMC协议。

[26] It may be amusing to realize that this problem, say, for the AND gate, is the very special case of the millionaires' problem in which each of their fortunes is restricted to 1 bit. This problem is still highly challenging, even if no one cheats!

[27] Again, in the case that half or more of the players abort, some others may not get the output (which is unavoidable, as e.g., for two players).

这是在两个独立的并行工作中完成的，使用了不同的技术，一个是由Chaum、Crépeau和Damgård[CCD88]完成的，另一个是由Ben-Or、Goldwasser和Wigderson[BOGW88]完成的（详见[AL11]中的完整细节）。

Theorem 18.4 [BOGW88,CCD88]. *Assume private communication channels between every pair of players. There is a protocol for SMC that is secure against computationally unlimited adversaries as long as more than 2/3 of the players are honest.*

请注意，我们只能容忍 $< 1/3$ 作弊玩家，而计算设置可以容忍 $< 1/2$。然而，正如那个设置一样，在这个设置中，常数 $1/3$ 是最好的。当然，安全保证要好得多，并且假设在许多具体设置中可能是合理的。此外，协议比计算设置中的协议更简单，效率也更高。最后，请注意，这些协议提供了对（公平）计算SMC结果（以及较弱的常数）的完全不同的证明，本质上是通过用计算加密方案替换假设的安全私有通道来实现的。[28]换句话说，如果信息论协议早些时候被发现（当然，它可能已经被发现），它将只使用密码学秘密通信就得出计算结果。这两个设置之间的联系，信息论和复杂性理论，既重要又迷人。这些联系以其他方式和地点表现出来，[29]并将在未来的发展中发挥重要作用。

18.9 More recent advances

直到现在，我们讨论了塑造该领域的一些非常古老的结果。然而，密码学是一个极其动态的领域，它不断地被现实世界的变化需求所重塑，以及新思想和新技术的出现所推动，这些新思想和技术解决了非常古老的问题和挑战。在本节中，我们讨论了这些令人兴奋的最近发展。

18.9.1 Homomorphic encryption

同态加密是一种公钥加密方案，允许任何拥有公钥的人在不访问解密密钥的情况下对加密数据进行操作。在20世纪70年代末，当此类系统出现时，已知允许进行加法或乘法（在适当的环上）但不是两者都允许的初始公钥系统。当时已经提出了创建一个同时可能进行这两种操作的系统的挑战，称为
*fully homomorphic encryption (FHE)*方案[RAD78]。当时也很清楚，FHE将是一个强大的密码学原语，随着时间的推移，出现了更多关于它的应用。

作为一项应用，您可以将敏感数据的计算委托给他人。例如（这可能对某些读者不相关），您可以同态加密您的财务信息，让您的会计师在“黑暗中”为您报税，然后解密。这种能力的重要性随着（计算能力较弱）的移动设备和（计算能力强大）的云服务的出现而变得巨大，这些设备通常为他们执行计算。FHE保证了这些计算输入的完全隐私（以及更多，我们将在讨论委托时看到）。作为另一个应用，SMC，它捕获了许多加密任务，可以使用FHE相当简单地实现，并获得巨大的好处。具体来说，已知的协议，它们非常通信密集，除了预处理和后处理之外，可以由一个最小的通信协议所取代：所需的唯一通信是在计算前后分别发送加密的输入和输出。

的确，FHE看起来太好了而不可能实现，但像其他看似不可能的密码学任务一样，它也取得了成果。构建这个构造需要40年，这是在Gentry的开创性博士论文中发展的[Gen09a, Gen09b]。Gentry的巧妙初始构造非常复杂且成本高昂，其安全性

[28] Of course, they give nothing for $n = 2$ players.

[29] For example, recall one beautiful example we have seen in Section 9.2: the construction of *information-theoretic* extractors from *complexity-theoretic* pseudo-random number generators by Trevisan [Tre99].

基于某种非标准版本的格问题的硬度。正如预期的那样，随后进行了大量工作来简化构造，使其更高效，并将硬度假设弱化为更标准的假设（以及在实际中实现FHE）。其中一项进步是论文[BV14]，它提供了相当大的简化，并依赖于[Reg09, Pei09]中更标准的错误学习假设。将FHE建立在通用假设（如门限函数的存在）之上是一项挑战。

18.9.2 Delegation of computation

让我们详细说明“云计算”应用，它今天有众多表现形式。计算能力较弱的Alice（例如，一部手机或一台笔记本电脑）希望执行一个计算量大的任务，超出了她的计算能力。她可以将任务委托给一个能力更强的（但仍然可行）的机器Bob（云或超级计算机），它提供执行此任务的服务。问题是Alice不相信Bob，Bob可能因为懒惰、错误或恶意而给出错误答案。Bob能否比Alice自己计算答案更快地让Alice相信答案是正确的?

这显然是第10.1节中讨论的交互证明设置的版本，但在效率参数上有两个基本差异，现在“缩小”以适应委托应用。以下由Goldwasser、Kalai和Rothblum [GKR08]提出的*doubly efficient*交互证明要求：我们希望有*efficient*个证明者和*highly efficient*个验证者。更具体地说，假设Alice想要计算$f(x)$，输入x的长度为n，其中f由一个已知的大得多（但仍然是多项式大小）的电路描述。证明者Bob应在多项式时间内运行[30]（，而不是在标准交互证明中拥有无限时间。验证者Alice应在（几乎）线性时间内运行（与标准交互证明中的多项式时间相比）。这个设置与标准交互证明模型不同的另一个特点是，一些协议的交互轮次非常少，通常只有一轮：Alice向Bob发送一条消息，Bob向Alice发送一条消息。

在初始进展 [GKR08, TKRR13] 之后，该问题在 Kalai、Raz 和 Rothblum 的美丽论文 [KRR14] 中得到了完全解决。我们下面陈述他们的结果，并指出更多近期研究其他 $\mathcal{PSPACE}$ 以下类别的交互证明中证明者能力的工作发展了 *doubly efficient proofs*（框架；例如，参见 [RRR16, GR18])。在另一个方向上，这个想法被用于验证量子计算（在适当的加密假设下）在 [Mah18]。

Theorem 18.5 [KRR14]. *Assuming fully homomorphic encryption, there is a 1-round doubly efficient interactive argument for every computation.*

这个（密码学）定理18.5中引人注目的成分之一是标准（非密码学）交互证明模型中的一个看似无关的结果。回想第10.1节，在多证明者交互证明（$\mathcal{MIP}$）中，验证者可以与几个证明者进行交互，而这些证明者不允许通信。本文介绍了一种这种模型的变体，称为*no-signaling* $\mathcal{MIP}$、[31]，并完全确定了它所证明的语言：可解在确定性指数时间内的$\mathcal{EXP}$问题类。将此与定理10.4进行比较，定理10.4对原始版本证明了$\mathcal{MIP} = \mathcal{NEXP}$。

回到原始的委托问题，请注意，还有一种情况下Alice可能不相信Bob——她可能想要保护她的输入x不被他看到，但仍然希望他为她执行计算。这是私有信息检索（PIR）的模型，在[CKGS98]中引入。鼓励读者发现有关PIR的激动人心的进展和剩余问题，我们在这里将不讨论。另外请注意，委托定理18.5只需要假设PIR，而不是（看似更强的）同态加密。

18.9.3 Program obfuscation

现在这里有一个非常强大的加密原语，每个软件公司都希望拥有：一个 *program obfuscator*。一个允许是概率性的混淆器 O（是一个有效的算法，

[30] Such interactive proofs are sometimes called *arguments* in the literature.

[31] This no-signaling was first studied in the context of quantum interactive proofs—it is a notion that comes from physics, motivated by the fact that information cannot travel faster than light.

将布尔电路映射到布尔电路，以完全隐藏它们除了功能以外的 *every* 方面。更精确地说，如果它将电路 C 映射到电路 $O(C)$，那么 $O(C)$ 计算的函数与 C 完全相同，但关于 C 的信息不会比通过黑盒输入输出访问 C 提供的更多。例如，一家基于其秘密技术和专有知识设计游戏或其他复杂软件的公司可以在不担心其竞争对手的情况下发布混淆代码。程序混淆是一种奇妙的加密方式，允许你使用程序代码，但无法理解其工作原理。其存在将意味着许多其他加密原语，包括本章讨论的大部分内容。

很遗憾，程序混淆器无法存在。这一点在Barak等人[BGI+01]的同一篇论文中被发现（通过一个高度复杂的证明），该论文正式定义了这个概念。鉴于这个令人悲伤的消息，他们提出了一个替代定义，称为 *indistinguishability obfuscator* (或IO)。IO也是一个从电路到电路的有效概率映射，它保留了功能（即，$IO(C)$计算与C相同的函数），它满足以下（不可区分性）属性。对于大小相同的两个*functionally equivalent*电路C, C'，它们的混淆形式$IO(C), IO(C')$在计算上是不可区分的。

两点在此尚不明确。在合理的难解性假设下，能否构造 IO (？这有什么用？这些问题在十多年里都没有得到解答，然后，从Garg等人突破性论文[GGH+13]开始，这个主题又重新活跃起来。确实，关于这个主题的活动带来了优秀惊悚片的全部激动人心的元素，包括基于新的、不熟悉的硬度假设（主要是群论和数论）的巧妙构造IO，新的算法攻击反驳了这些假设中的某些（例如，参见[CGH+15, HJ16]），改进和修复了免受过去攻击的假设，关于幸存假设合理性的激烈争论，等等。这仅仅是关于第一个问题，即IO的存在。在第二个问题，即IO的效用方面，也有大量的活动，揭示了它的潜在力量。它被证明意味着一系列其他极其有用的密码学原语，其中一些之前没有构造（例如，参见[SW14, GGHR14]）。此外，IO被证明等价于[GR07]中“最佳可能的模糊化”定义，等等。

非常有趣的是，当尘埃落定后，看看这个主题会如何发展！

18.10 Physical attacks

本节应降低前节所讨论的奇妙数学理论带来的振奋感。众所周知，加密系统无处不在。它们是否安全？嗯，我们知道它们在数学上是安全的（例如，据我所知，还没有人找到有效的分解算法）。[32] 那么，我们为什么还不断读到黑客入侵最安全的计算机系统，如五角大楼的，以及从银行和信用卡账户中进行电子盗窃的新闻呢？当然，答案是这些攻击在上述数学模型之外起作用。安全系统的*implementation*提供了许多途径和漏洞，允许攻击者入侵系统而无需违反其依赖的任何数学、严格的安保证明！通常，这些弱点是由于智力上无聊的原因产生的，比如愚蠢和疏忽地发送密码或其他识别信息，留下大门敞开和电脑开启，等等.[33] 然而，过去十年或二十年的研究表明，即使在看似滴水不漏的情况下，也存在微小的弱点，允许巧妙的攻击完全破坏加密系统。

这样的微小弱点可能是什么？计算机系统是物理对象，当它们运行时，会发出各种类型的众多物理信号。此外，这些信号与它们执行的操作相关联。正如伟大的密码分析师Adi Shamir所说，
computers are telling us what they are doing, and we just have to listen。将这个口号字面化，Shamir及其合作者[GST14]最近设计了一种*acoustic attack*，其中在执行操作的笔记本电脑附近放置一个廉价的麦克风

[32] And the efficient *quantum* factoring algorithm is still waiting for a quantum computer that can execute it—see more on quantum computing in Chapter 11.

[33] One popular jargon word for obtaining sensitive information this way is *phishing*—the reader can read about such attacks and protecting against them by searching on this word.

RSA解密可以完全提取其秘密RSA密钥！这只是密码分析领域快速发展的一个例子，或者简单地说，*physical attacks*），它向安全系统设计者发送不同形式的唤醒信号，让他们重新思考设计方案，这是一场永无止境的猫鼠游戏。这类攻击包括（这些攻击并非我凭空捏造——所有短语均来自实际论文）时间分析、电磁攻击、光学攻击、功耗分析、漏洞分析、微波攻击、缓存攻击、热分析、冷启动攻击等等。其中一些攻击也适用于硬件！许多攻击不仅需要独创性和洞察力，还需要复杂的数学和物理分析。有趣的是，许多对密码系统的攻击都是以一种可能类似于疾病治疗方法发现的方式被发现的：观察者在各种条件下观察一个系统，直到出现一些强烈的信号.[34]

实际上，目前来看，大多数此类物理攻击无法大规模进行，即同时应用于数百万台计算机，比如说。从数学上讲，此类物理攻击的丰富性（其中许多仅通过实验发现）提出了理论抽象和建模的挑战，这可能会为更好的安全系统设计提供信息。

18.11 The complexity of factoring

为了给上一节中令人清醒的新闻再添一桶冷水，我们回到数学攻击。如今，许多计算机安全系统，[35] 支持庞大的电子商务和金融交易网络，依赖于大整数分解的难度假设。我相信，大多数人，以及大多数决策者，都没有意识到一个 *single*，一个简单明了的数学问题，明天可能被数学或计算机科学本科生解决，是潜在经济崩溃的 *only* 障碍。事实上，很难想到任何其他具体的潜在 *mathematical* 发现，能够造成类似规模的灾难性破坏。我们准备好应对这种可能性了吗？当然，我们（非常）少数的替代陷阱门函数在原则上可以替换分解作为这些安全系统的基础，但这些似乎效率要低得多，可能需要数月或数年才能实施。在这期间会发生什么？如果你觉得这种末日般的沉思太过压抑，这里还有一个问题值得深思。假设你发现了一个高效的分解算法——一个如果公布可能会给你带来名誉和财富的惊人发现，但也可能对电子商务造成破坏。你会怎么办?

[34] For example, the paper [GST14] describes how viewing sonograms of a computer when performing a variety of different arithmetic operations revealed patterns of frequencies that correlate with the binary representation of the secret RSA keys as these are processed during the execution of a decryption algorithm.

[35] Perhaps the majority still, even with the rise of elliptic curve cryptography (see e.g. [HMV06]), for which the questions we raise here can be studied as well.

19 Distributed computing: Coping with asynchrony

分布式计算捕捉任何许多计算机或进程旨在以需要交互的方式实现某些目标（共同或分离）的情况。这当然包括服务器农场；超级计算机；传感器网络；组织内部网络；以及互联网；但也包括你的单个笔记本电脑，其中许多组件具有一定的独立性。此外，它还可以作为自然科学中许多现象的模型。显然，这是一个庞大且异质的研究领域。

分布式计算在很多方面与密码学相似，包括多方的互动、异构性和众多目标，以及形式建模的复杂性，但每个问题的核心问题相当不同。确实，有很多互动；密码学被用于某些分布式算法中，而分布式算法思想被用于密码协议中。

在这一章中，我们关注一个在分布式计算中存在而在本书讨论的所有其他主题中缺失的主要问题，即*asynchrony*。异步分布式计算中的各方[1]不能假设一个*common clock*来保持时间并协调它们之间计算的全局进度。从另一方期望的信息可能会无限期延迟，因此无法确定这样一个方是否只是运行缓慢、完全损坏，或者恶意拒绝合作。即使所有来自其他方的消息最终到达，它们的到达顺序也可能是任意的。但是生活必须继续，每个方都必须根据其局部观点做出决策并继续计算。请注意，异步性是一个真正的头疼问题：它可能源于各方之间的计算差异、它们之间底层通信的性质或其他原因。必须处理它。事实上，这个问题如此核心，以至于异步分布式系统的子领域也非常庞大。关于这个主题的一些优秀文献包括[Lyn96, AW04]，它们特别将下面讨论的高级语言形式化。

我们将讨论异步系统需要解决的两个最基本问题，一个是在各方有共同目标的情况下，另一个是在他们有冲突目标的情况下：*coordination*和*sharing resources*，或者用该领域的形象语言来说，是*Byzantine generals problem*和*dining philosophers problem*。我们将看到，在非常一般的设置中，这两个问题[2]实际上是完全*impossible*的。在这里，与本书的大部分内容不同，我们关注的是简单与困难，"不可能"实际上意味着不可能，就像图灵对停机问题的结果一样。也就是说，这些问题的算法*finite*是不存在的！与图灵的工作一样，不可能性结果非常有用——真正重要的问题需要得到解决，不可能性结果关注的是需要添加假设或放宽要求，以适应各种实际设置。在分布式计算中，不可能性证明非常微妙，对“异步”的确切定义以及通信介质允许的内容极为重要——我们将探讨这些建模问题背后的某些原则。我们还将看到，一个事先相当意外的数学领域——*algebraic topology*——似乎与这种异步环境中的分布式问题密切相关，并且在证明不可能性和可能性结果方面非常强大。最后，我们将讨论的不可能性结果将适用于*deterministic*程序。正如我们在其他章节中看到的那样，随机性在这里也非常强大：在异步设置中，它可以使不可能变为可能！

19.1 High-level modeling issues

这里我们非正式地讨论了一些必须正式定义的异步分布式计算的主要问题。本章后面讨论的问题和结果将使其中一些问题更加具体。本节主要是为了给读者留下该领域建模复杂性的深刻印象。再次强调，上述教科书和引用的参考文献提供了更精确和详细的信息。

[1] It must be a coincidence that parties, processors, players, participants, philosophers, P_i, and so forth, all start with the letter “p.” We shall use these terms interchangeably in this chapter when referring to the interacting entities in distributed systems.

[2] Hundreds of other impossibility results, as the title of these papers proclaim, can be found in [Lyn89, FR03].

Atomicity 在所有玩家都无法参考的全球时间不存在的情况下，*simultaneous*动作可以破坏信息（或者更糟，在某些并发算法的应用中，甚至可能破坏您的资产甚至生命）。这可以通过引入*atomic*动作来解决。我知道，这听起来对人类来说更糟，但你知道我这里所说的“原子”是什么意思：瞬间发生，没有干扰。更准确地说，处理器对对象（从内存寄存器读取或写入、检查资源可用性、获取可用资源控制权等）的某些访问原语，在现实中从未瞬间发生，但被当作瞬间发生处理：一个处理器对对象的访问必须完成，另一个处理器才能访问它。访问原语的选择自然取决于应用，以及它们的物理可实现性和成本（硬件或软件）。研究它们的相对能力是这个理论的重要组成部分（并且占据了大量的文献）。我将指定我们讨论的问题所做出的选择，但为了让你感受到多样性，这里有一些为访问内存寄存器而发明和研究的原子*read-modify-write*原语名称：*test-and-set, fetch-and-add, compare-and-swap*和*load-and-store*。

Program 并发程序（或分布式协议）将每个参与者分配一个程序，其步骤在本地计算和对共享变量的某些原子操作之间交替。通常没有对处理器的计算能力、内存或通信的限制.[3] 在某些但不是所有的问题中，在计算开始之前会向参与者提供一个输入。

Symmetry 一组研究的重要问题包括 *symmetric* 问题，在这些问题中，正确行为或合法输出的规范对所有玩家（被认为是相同的，例如，没有唯一的ID）都是相同的。对于这类问题，希望每个玩家使用的程序都是 *identical* (这样的并发程序有时被称为 *anonymous*)。在这里我们也要求它，因为我们考虑的问题在这种意义上是对称的。

我们现在讨论异步环境中并发程序的性质 *execution* (或 *run*)。

Asynchrony 给定动作的原子性，让我们阐述“异步”的含义。我们（以及一般理论）通常考虑最坏的情况：一个（调度）对手控制并发程序执行。它可以随意“唤醒”不同的参与者。它被允许以任意方式调度和交错原子动作，并完全了解迄今为止的全局系统状态。这个对手的唯一限制是它必须按照它们执行顺序保持每个处理器的自身动作；这保证了每个局部视图的一致性。然而，例如，从A到B发送的两个连续消息可能会以相反的顺序到达。

Faults 一个理论的重要方面是必须能够容忍的 *faults*。正如在密码学中，该理论通常区分两种主要的故障。良性的故障是处理器在计算过程中的简单终止（有时称为“失败停止”或“崩溃”故障）。这可以通过让调度对手无限期地延迟该处理器的所有未来动作来建模。恶意的故障（通常称为“拜占庭”）允许处理器任意行动，完全忽略它应该执行的程序。我们将在此关注第一种类型，再次使不可行性结果更强。

Communication medium 在分布式系统中研究了相当多的通信模型。最常见的是 *message passing* 和 *shared memory*，每个都包含许多变体和子模型。在第一个模型中，参与者是无向图中的顶点，它们只能通过与其直接邻居交换消息来进行通信（例如，参见 [AW04] 了解形式定义和背景）。在第二个模型中，有一组共享寄存器支持某种原子操作。共享资源

[3] This is especially convenient for our focus on impossibility proofs, as it makes them stronger. In actual algorithms, one of course cares about minimizing all of these, see, for example, [EGSZ16].

经常提供这样的通信手段——它们要么是“锁定”的（在使用时），要么是“空闲”的（在不使用时），并且可以访问这种状态并提供外部信息给处理器。在讨论共识时我们将使用的变体允许每个共享寄存器都能被所有人读取，甚至可以同时读取（如在广播设置中），但只能由一个指定的玩家写入。

共享内存模型通常比消息传递模型为参与者提供更全局的视角。例如，在共享内存模型中，一方无法向其他人发送不同的消息；她所写的内容都可以被其他人看到。实际上，在共享内存模型中（使用读写寄存器）模拟消息传递算法很容易。因此，设计并发算法更容易，而对于共享内存模型来说，证明不可能性结果更难。然而，该论文[ABND+87][5]给出了相反方向的模拟（即消息传递系统中的读写寄存器），前提是大多数参与者没有崩溃（或者，如果少于三分之一是拜占庭式的）。

Properties of programs 考虑到并发程序必须承受的复杂性（特别是调度对手的力量），定义一个“成功”的程序并非易事。当然，这取决于它应该解决的问题或应该诱导的行为。我大致提到了许多研究属性中的几个；我们很快就会看到一些实际应用。关于这些属性及其关系的更精确信息可以在[HS11]中找到。本质上，所有这些属性都需要系统在任意延迟和某些组件可能失败的情况下持续运行（在多种意义上）。在一个 *wait-free* 程序中，每个保持存活（没有失败）的处理必须在有限步骤后计算其输出。在一个 *deadlock-free* 程序中，除非所有处理器都已停止，否则 *some* 处理器将能够访问请求的资源。在一个 *starvation-free* 程序中，请求某些访问的 *every* 处理器最终会得到它。在一个 *fair* 程序中，授予一个处理器的 *every* 访问最终会被授予请求它的另一个处理器。请注意，某些属性是有序的：例如，公平性特别意味着没有饥饿，而饥饿反过来又意味着没有死锁。

值得强调的是 *a property is satisfied by a concurrent program if it is satisfied in every possible execution of that algorithm (for all possible scheduling of actions and allowed failures).*

19.2 Sharing resources and the dining philosophers problem

埃德加·迪杰斯特拉，他奠定了分布式计算和并发编程[6] (以及其他计算机科学领域的基础)，喜欢发明故事来捕捉该领域出现的许多问题。其中*dining philosophers problem*是最著名的这些故事之一，自那时以来一直作为并发控制问题的模型问题。迪杰斯特拉在[Dij71]中描述了它，豪尔在[Hoa78]中将其形式化。这个问题抽象了在对称环境中的资源分配困难。我们重复迪杰斯特拉的非正式版本，当然这需要将其翻译成处理器和资源的枯燥语言。它在图20中有所描述。

想象有 n 个哲学家围坐在一张桌子旁。这些哲学家是 *identical*，即没有独特的名字或ID。每两个哲学家之间有一把叉子。每个哲学家面前有一碗（无限量的）意大利面。碗是私人的，但叉子是共享的。哲学家们在两种活动之间交替：思考和进食。思考不需要（外部）资源。要进食，哲学家必须使用左右两边的叉子。一个原子操作是请求相邻的叉子，如果可用，就取走它。[7] 获得两个叉子的哲学家会吃一段时间，然后依次释放叉子。

原始挑战要求一个 *deadlock-free* 程序（其中至少有一位饥饿的哲学家能够吃到食物）。我们还将讨论一个 *starvation-free* 程序（其中所有饥饿的哲学家都能吃到食物）。由于情况是对称的，我们要求并发算法也是对称的：即每个

[4]This is of course a generality; each of these models has numerous submodels, and the relationship is more tricky; a comprehensive paper on these relationships is [AFR06].

[5]Written after most results described below.

[6]His paper [Dij65] on *mutual exclusion* marks perhaps the birth of the field.

[7]So, access to a fork is *mutually exclusive*.

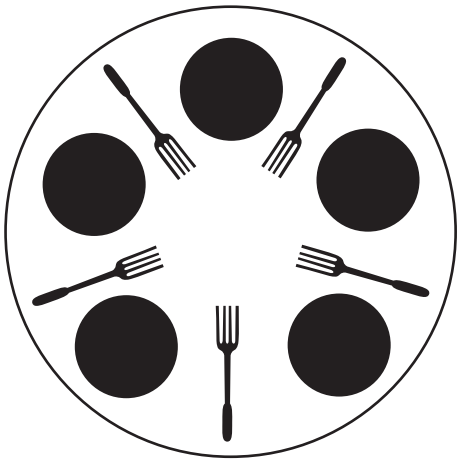


图20. 哲学家就餐桌。

哲学家有相同的程序。此要求还意味着哲学家本身具有完全的对称性；他们是相同的，特别是，没有独特的名称或ID。一个简单程序 P 来说明该问题如下。

1. *Try to obtain the left fork.*
2. *Try to obtain the right fork.*
3. *If unsuccessful in either attempt, go to 1.*
4. *Eat.*
5. *Release left fork.*
6. *Release right fork.*

显然，这个程序 P 不是无死锁的。想象一个调度器只为每位哲学家激活一步（任意顺序）。之后，每位都拿着她的左叉，并且这个状态将永远不会改变，无论未来的激活顺序如何。

当然，可以避免死锁：只需考虑尝试抬起左叉然后释放它的程序。当然的问题是没有人能吃到东西。你能找到一个无饥饿的程序吗？有许多建议的程序，但所有这些程序都超出了上述模型，要么打破了对称性（例如，通过给哲学家不同的ID），要么改变了问题（例如，通过使哲学家“卫生”[CM84]，或者添加一个“服务员”）。但Dijkstra的原始问题仍然悬而未决，直到Lehman和Rabin [LR81] 证明了解决这个问题是不可能的。

Theorem 19.1 [LR81]. *There is no deadlock-free deterministic, symmetric program for the dining philosophers problem.*

一个证明必须为每个这样的程序提供一个调度器，该调度器会导致所有哲学家都饿死。证明的想法很简单（尽管将其形式化并不容易）。调度器简单地以循环顺序激活每个哲学家一步，并永远重复这些轮次。关于这些轮次数量的归纳主张是，在每一轮开始之前，系统的状态是对称的（即，所有哲学家都处于相同的状态）。这是在开始时的情况，程序的对称性确保在每个轮次中，任何哲学家的（归纳上相同的）步骤的结果都是相同的。为了得出结论，假设在某个轮次中，某个哲学家正在吃饭。归纳主张以及每个哲学家在每个轮次中执行完全相同的步骤并获得相同的结果的事实，意味着在这个轮次结束时，所有哲学家

正在吃饭。但这不可能，因为每个人都需要两把叉子来吃饭！请注意，异步性对于这个不可能的结果至关重要，但在这里它被以一种相当温和的方式使用。调度器本质上阻止任何玩家在其他玩家之间各执行一步之前执行其程序的连续两个步骤。

但是，论文[LR81]的主要信息是，如果允许概率程序，那么就有 *is* 一个解决方案！本文表明，上述程序 P 的自然概率变体会挫败调度对手，某个哲学家将以概率 1 得到食物。事实上，在莱赫曼和拉宾称为“自由哲学家算法” R (的新概率程序中，本质上唯一的改变是，当一个没有叉子的哲学家试图得到一个叉子时，她会掷一个公平的硬币来决定是先尝试左叉还是右叉。正确性证明相当不平凡。它关键地使用了（一个自然的）假设：每个非故障哲学家必须无限次地被调度器激活。杜弗洛特、弗里堡和皮卡龙尼 [DFP04]（他们还开发了一个非常优雅的形式框架来表达此类概率程序的分析）移除了这个假设。

Theorem 19.2 [LR81, DFP04]. *There is a probabilistic symmetric program for the dining philosophers problem that is deadlock-free with probability 1.*

我们以[LR81]中的另一颗宝石作为结论。这些作者观察到，对于程序R，存在一些调度方案可以防止除了一个哲学家之外的所有哲学家进食。虽然这个程序没有死锁，但它远非令人满意（至少对于饥饿的哲学家来说）。他们设计了一个新的概率性程序R'来解决这个问题，因此它是*starvation-free*！一个关键因素是允许叉子的状态告诉我们不仅仅是它是否在使用。回想一下，只要它是有限和局部的，就没有限制让状态携带更多信息。新的算法仅将一个比特添加到叉子的状态：与它相邻的两个哲学家中哪一个最后进食。（初始状态可以是任意的。）然后，他们称之为“有礼貌的哲学家算法”的程序R'简单地阻止最后进食的哲学家在下一个时刻访问它。这种修改保证了在每一个调度中，每个被无限次激活的哲学家将以概率1无限次地得到进食的机会!

Theorem 19.3 [LR81]. *There is a probabilistic symmetric program for the dining philosophers problem, which is starvation-free with probability 1.*

19.3 Coordination: Consensus and Byzantine generals

在上一节中，我们研究的问题反映了参与者之间的*conflicting*情况，其中他们为了共享资源而竞争，主要问题是对称性破缺。我们现在转向一个反映*cooperative*情况的问题，其中玩家共同尝试计算一个函数（或执行一个更通用的任务），该函数依赖于他们的私有输入，并且异步性是主要障碍。我们将在本章中较早描述的*shared-memory*模型和*message-passing*模型中考虑它。描述此类协作任务的并发程序的一个常见形式化模型，在两种类型的通信媒体中，是Lynch和Tuttle的输入-输出自动机[LT89]（但自然地，我们将保持讨论的非正式性）。我们将关注的程序属性，在共享内存模型中称为*wait-freedom*（有时在消息传递模型中简单地称为*fault-tolerance*），是“好”参与者无论所有其他参与者的行为如何都能在有限时间内正确完成任务。在这里，“好”简单地意味着在故障停止故障模型中是“活着的”，对于恶意（或拜占庭）故障模型则是“诚实的”。

我们将关注的问题是 *consensus* (或协议)问题。它在许多可能发生组件偶尔故障的实际应用中出现的非常早。其中最常提到的应用包括 *transaction control* (例如，当从一家银行向另一家银行转账时，两台计算机必须承诺)，*consistency control* (例如，从崩溃中恢复，计算机如何从保留相同信息副本的其他计算机更新其内存)，*flight control* (例如，空中交通管制，等待的飞机必须同意谁获得着陆许可)，以及 *coordination* (例如，一组军队将军必须同意他们的部队是否应该进攻或撤退)。[8]

[8]The name “Byzantine generals problem”, coined by [LSP82], is a common synonym for the consensus problem with potentially malicious faults.

显然，在这些（以及其他）例子中，共识是一种在系统中不存在（但需要）的同步原语。

更普遍地，考虑到参与者所处的异步、故障环境，很明显，拥有一个共识原语将非常有用。例如，如果他们同意一个领导者，那么这个领导者可以成为时间守护者，从而成为计算进度的守护者；可能收集所有其他输入并自行执行计算；等等。事实上，共识感觉像是一个自然的“完整”问题。当然，这并不简单，因为这样的领导者可能会失败，或者更糟——是恶意的。但尽管不简单，共识任务确实是完整的，至少在共享内存模型中是这样！Herlihy [Her91] 的基础论文，[9]，建立了无等待程序的结构理论，研究了它们之间的相对能力，形式化了它们之间的归约，并证明了此类任务中共识的完整性。总之，出于直观、实用和理论的原因，共识是一个基本问题——让我们现在定义它并思考其可解性。

The consensus problem 在词义上，“共识”意味着玩家必须都同意*one*的值，这个值最初至少由他们中的一人持有。让我们以二进制值的简单情况稍微正式地陈述一下。当然，某些应用需要大的值域，但我们的主要关注点将是不可行性结果。

想象 n 处理器 $P_1, P_2, \ldots, P_n$。玩家 P_i 有一个二进制输入值 $x_i \in \{0, 1\}$。其任务是提交一个二进制输出值 y_i，以满足以下属性：

- **Consistency** 所有已提交的值 y_i 均相等。
- **Validity** 每个已提交的值 y_i 必须等于某个 x_j。

再次，挑战在于找到一个并发程序，其中每个处理器都在有限时间内提交（或失败）。请注意，有效性约束防止了一个简单（且无用的）解决方案，即无论它们的输入如何，它们都提交为0。

Fischer、Lynch和Paterson的突破性论文[FLP85]表明，在最强想象中的消息传递模型中，共识是不可能的.[10]

Theorem 19.4 [FLP85]. *For every $n \geq 2$, there is no program for consensus in the message-passing model that tolerates even one fail-stop fault.*

此外，即使所有发送的消息都保证在有限时间内被送达，甚至如果我们要求只有一个活动处理器在有限时间内提交，这个定理的证明仍然成立！此外，尽管问题具有对称性，该结果不仅适用于对称、相同的程序（例如，即使每个处理器都有并使用唯一的ID，它也成立）。

我建议阅读原始论文[FLP85]中清晰、详细的证明。在这里，我将突出证明的一些高层次元素，因为我们稍后会回到它。证明分为两个概念部分。首先，它使用一个假设的有限程序来分类在运行时可以到达的系统所有配置。然后，它使用这个分类来证明无限执行的存在。让我们详细说明这两部分。

要开始，假设为了矛盾，一个给定的程序具有以下性质：在每次执行中，[11] *some*处理器在有限时间内提交一个二进制值。取第一次发生这种情况的时间，并注意由于一致性，其他处理器不能在之后提交任何其他值。在每个执行中提交的第一个值现在被用来分类所有*global configurations* (，即程序可能到达的所有处理器状态和所有消息队列)的状态，无论是*univalent*还是*bivalent*。如果一个配置*univalent*的所有后续操作都产生相同的提交值，我们称它为*univalent*。如果某些执行导致提交，我们称它为*bivalent*。

[9] Playing a similar role for distributed shared-memory wait-free algorithms, as the Cook-Karp-Levin papers on $\mathcal{NP}$-completeness (see Section 3.8) played for sequential polynomial-time algorithms and verification.

[10] Prior to this result, impossibility for 3 players and one *malicious* failure was proved in [LSP82]. An informal argument for impossibility in the special case $n = 2$ players was given in (the appendix of) [AEH75]. A good challenge for a motivated reader is to write down a formal proof for this simple case, if only to realize the need for precise definitions of what the processors and the scheduling adversary can do.

[11] Recall that each ordering of processor actions and messages delivered yields an execution, or run of the program.

0，其他人承诺为1。显然，没有二价配置是最终的。这种分类为寻找无限运行建立以下归纳论证。基础情况是证明存在一个输入，其配置是二价的。归纳论证（复杂）表明，从*every*二价配置，存在一个有限的时间表，导致另一个二价配置。这个组合论证，仔细使用模型的性质，是证明的核心。现在通过归纳，对于某个时间表，程序将永远运行，从而证明不存在无等待程序。

bivalent 的概念以及更一般地，*indecisive* 配置（例如，对于非二进制输入）对未来不可能性结果产生了极其重要的影响。事实上，定理19.4被扩展到证明在共享内存模型中也无法达成共识，由Loui和Abu-Amara [LAA87] 以及Herlihy [Her91] 完成。

Theorem 19.5 [LAA87, Her91]. *For every $n \geq 2$, there is no wait-free algorithm for consensus in the shared-memory model that tolerates even one fail-stop fault.*

Probabilistic programs 当然，这些结果为实践中使用的各种额外假设或对问题的放松提供了依据，这实际上导致了这个基本共识任务的有限程序。但就像就餐哲学家问题一样，如果允许随机化，原始问题是可解的。[12] 此外，即使在恶意故障的情况下，这个积极的结果仍然成立。

第一个概率算法，用于消息传递模型，[13] 由 Ben-Or [BO83] 发现。[14] 它有两个部分，根据故障类型而定；它可以处理少于 $n/2$ 个故障停止故障，以及少于 $n/5$ 个拜占庭故障（后者由 Bracha [Bra84] 改进到 $n/3$）。

Theorem 19.6 [BO83, Bra84]. *For every n, there is a probabilistic concurrent program for the consensus problem that terminates in finite expected time under either of the following assumptions:*

- *less than $n/2$ of the players may fail;*
- *less than $n/3$ of the players are malicious.*

一个关键是将共识问题降低到 *joint coin-flipping* 问题。这些原始解决方案虽然有限，但以指数时间运行，现在正在努力提高其复杂性。大多数这种发展，导致一些预期的多项式时间算法，在 Aspnes 的调查 [Asp 03] 中以历史和技术细节给出。但自 Ben-Or 的原始论文以来，一个核心挑战一直悬而未决。他的算法可以容忍一个“强大”的对手，即选择在（和根据）执行期间哪些玩家被腐化的对手。如前所述，算法需要常数比例的恶意故障的预期指数时间，但如果坏蛋的数量低于 $\sqrt{n}$，则复杂性将提高至预期的多项式时间。剩余的挑战是存在一个可以容忍常数比例恶意故障的预期多项式时间程序。这最终在首次提出 30 年后，由 King 和 Saia [KS13] 在肯定方面得到解决。

Theorem 19.7 [KS13]. *For every n, there is a probabilistic concurrent program for the consensus problem that terminates in expected polynomial time even if a (strong) adversary can corrupt up to $n/500$ players during the execution.*[15]

19.4 Renaming, k-set agreement, and beyond

我们现在回到确定性程序，以及拓扑学与异步分布式计算之间美妙联系的诞生。除非另有说明，所有所述结果均基于共享内存模型，这是后续大多数工作所关注的。

[12] And “wait-freedom” is now defined as termination in finite *expected* time.
[13] And hence also for the shared-memory model, which can simulate it.
[14] The title of this paper refers to the Lehman-Rabin paper [LR81]. This is not surprising, as Ben-Or was Rabin’s PhD student at the time.
[15] No attempt was made to optimize the constant—the proof is complex enough without it.

共识的不可能性，以及理解定理19.4证明边界的尝试，推动研究人员定义相关问题，以阐明异步环境中基本协议任务的可能性和不可能性之间的界限。一个可能的线索是数值巧合：共识需要one共同输出值，即使有one故障也是不可能的。论文[ABND+87]研究了扩展共识的一些一般$renaming$问题，其中可能的输出值集合必须小于可能的输入值集合。在计算机系统中，当许多计算线程试图使用少量共同资源时，缩小名称空间是一个非常普遍的问题。论文[ABND+87]给出了一些依赖于两个集合大小差距和允许的故障数量的可能性和不可能性结果。然而，对边界的精确描述仍然悬而未决。

为了在阅读本文后获得紧密的理解，Chaudhuri [Cha93] 定义了一个特定的重命名问题，即 k-$set\ agreement$ 问题，如下所述。

再次，我们有 n 名玩家，$P_1, P_2, \ldots, P_n$。玩家 P_i 从大小为 $k+$ 的集合中有一个输入值 x_i，例如，$x_i \in \{0, 1, \ldots, k\}$。其任务是承诺一个输出值 y_i，以满足以下属性：

- **Consistency**集合 y_i 的基数最多为 k。
- **Validity**每个提交的 y_i 必须等于某个 x_j。

再次，挑战在于找到一个并发程序，其中每个处理器在有限步骤内提交（或失败）。请注意，共识问题是 k -集合一致性的特例，其中 $k = 1$。另外，有效性约束保持不变。

第一个（正面的）结果，由Chaudhuri [Cha93]获得，扩展了共识在0个故障的情况下是可能的这一明显事实，并再次暗示了相同的数字巧合。

Theorem 19.8 [Cha93]. *For every n and k, there is a wait-free protocol for k-set agreement tolerating less than k faults.*

为了扩展[FLP85]并在故障数恰好为k时证明其不可能性，Chaudhuri [Cha93]试图模仿并推广其证明技术到这种情况。她定义了二值配置的自然类比，即$indecisive$ (或$(k+1)$-$valent$)配置，这些配置不能是最终的（因为仍然有持续的执行可以导致第一个提交玩家的所有可能的提交值）。剩下的是给出类似的归纳证明，即这样的不确定配置可以通过一个恶意的调度强制持续永远。论文只证明了归纳的基础情况，即存在一个输入使得初始配置$indecisive$！记住，对于$k = 1$来说这很容易。对于$k > 1$来说则不然，这篇论文的一个关键贡献是证明基础情况的论证：使用我们很快将要讨论的工具$Sperner's\ lemma$。论文进一步表示希望类似的论证可以证明归纳步骤，完成一个不可能性的证明。

“斯佩纳引理”提示足以让三个独立团队迅速实现这一希望。Borowsky和Gafni [BG93]、Saks和Zaharoglou [SZ00]以及Herlihy和Shavit [HS99]证明了k-集合一致性（即使在k故障的情况下）的不可能性。

Theorem 19.9 [BG93, HS99, SZ00]. *For every n and k, there is no wait-free program for k-set agreement tolerating k faults.*

论文[SZ00, HS99]中的证明明确使用了$topology$。这种联系为使用强大的数学理论来理解分布式系统中的异步性打开了大门。特别是，它具有将$dynamic$对象（如程序、调度和执行）表示为$geometric$或$topological$对象，从而允许通过静态对象的结构推理来反映动态对象属性的极具吸引力的特点。描述这些发展的优秀书籍是[HKR13]。

定理19.9在共享内存模型中得到了证明，因此，正如讨论的那样，也适用于消息传递模型。然而，请注意，对于那个较弱的模型，后来发现了一个更简单的不可行性证明[BRS11]，这个证明完全不含拓扑；实际上，这个证明使用了从共识到k-集合一致性的这种模型中的降低，这使得作者可以直接应用消息传递不可行性结果定理19.4。

让我们现在回到本节剩余部分的共享内存模型。上述三篇证明定理19.9的论文在明确程度上使用了拓扑。[16]虽然[BG93, SZ00]只证明了定理19.9，但[HS99]中的证明是从一个更一般的定理推导出这个定理的，我将在结尾处提到这个定理，它使用了*algebraic homology theory*。但是首先，我给出一个非常高级的证明描述——从选择的高度来看，它们看起来都一样。然而，我主要遵循[SZ00]中的证明结构，我在这里隐藏的细节山脉可以在那篇论文中找到，该论文还提供了大量的直觉和仔细解释的特殊情况。

19.4.1 Sperner's lemma and Brouwer's fixed-point theorem

首先，让我们陈述Sperner引理，它是一个*combinatorial*陈述。然后我将陈述Brouwer不动点定理，它是一个*topological*陈述。然后我们将讨论一个约简，展示第二个是如何从第一个中得出的。这将为定理19.9的证明做好准备。k-集合一致性的不可能定理，它是一个*computational*陈述，本质上以与Brouwer不动点定理相同的方式简化为Sperner引理！实际上，不可能定理可以被视为一个不动点定理。我们将限制我们的注意力到k-集合一致性的$k = 2$，这意味着我们只需要Sperner和Brouwer来处理欧几里得平面$\mathbb{R}^2$，这在纸上很容易说明。同样的想法可以相当简单地应用于一般的k，使用这些数学结果在$\mathbb{R}^k$中的适当术语，我们将避免使用（例如，用单纯形替换三角形，用三角剖分替换单纯形细分）。你可能想模仿下面的证明来处理一维情况$k = 1$（其中只有线段）。在这种情况下，Sperner引理和Brouwer的不动点定理非常简单，但如前所述，共识($k = 1$)不可能结果不是。

我们回到 $\mathbb{R}^2$。我们需要几个基本定义。参见图21。设 T 是平面上的一个（几何）三角形，其顶点为 v_1, v_2, v_3。*triangulation* D 的 T 简单地是将 T 划分为有限个（较小的）三角形的划分。*vertices* 的 D，记为 $V(D)$，是这些较小三角形的所有顶点。如果一个 3-着色 $\chi : V(D) \to \{1, 2, 3\}$ 将 D 的顶点着色，且用不同的颜色着色 T 的顶点，即 $\chi(v_i) = i$，并且用其端点颜色之一着色区间 $[v_i, v_j]$ 上的每个顶点 u，即 $\chi(u) \in \{i, j\}$，则称其为 *Sperner*。最后，如果一个三角形在 D 中的顶点具有不同的颜色，则称其为 *rainbow* 三角形。

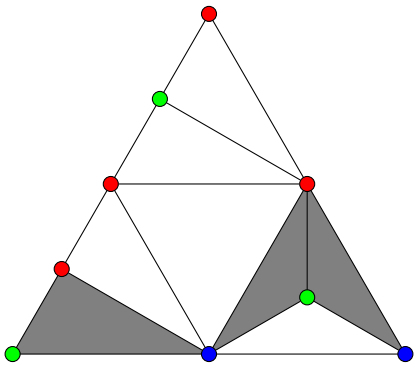


图21. 三角剖分和斯佩纳着色。彩虹三角形被着色。

显然，如果 D 是空三角剖分（即，它单独留下 T），那么只有一个 Sperner 着色，并且它使 T 成为彩虹三角形。一旦 D 非平凡（即，添加顶点），T 就不再是 D 中的三角形，并且有几种 Sperner 着色。然而，无论你如何做，都无法避免彩虹三角形。这就是 Sperner 定理。

Theorem 19.10 [Spe28]. *For every triangulation D of T, and for every Sperner 3-coloring χ of $V(D)$, there is a rainbow triangle in D.*

[16] A later proof of Attiya and Castañeda [AC11] is completely combinatorial but clearly states the value of the topological viewpoint leading to their proof.

[17] A lot of hard work is hidden by this word.

这个引理的证明很简单，本质上源于每个无向图都有偶数个奇数度顶点的事实（如果你从未见过这个证明，请尝试证明它）。

这里是有固定点定理的Brouwer，表明另一个对象是不可避免的：从T到自身的任何连续映射中的固定点.[18]

Theorem 19.11 [Had10, BJ11]. *For every continuous function $f: T \to T$, there must be a point $x \in T$, such that $f(x) = x$.*

让我们从Sperner引理推导出这个定理。固定一个连续映射 $f: T \to T$。这个约简有三个部分：首先，从 f 中获得 T 中所有点的 3-着色 χ_f。其次，证明对于 *every* 三角剖分 D，χ_f 是 $V(D)$ 的Sperner 3-着色。第三，将这个论点应用于更细和更细的三角剖分，并证明，在极限情况下，χ_f 下的彩虹三角形是 f 的一个不动点。让我们首先看看 f 如何在 T 中的 *all* 个点上诱导出 3-着色。

1. χ_f 定义如下。设 w 是 T 中的任意一点。那么，$w = \lambda_1 v_1 + \lambda_2 v_2 + \lambda_3 v_3$，其中 $\lambda_1 + \lambda_2 + \lambda_3 = 1$。同样，由于 $f(w) \in T$，$f(w) = \mu_1 v_1 + \mu_2 v_2 + \mu_3 v_3$，其中 $\mu_1 + \mu_2 + \mu_3 = 1$。设 $\chi_f(w) = i$ 为最小的 $i \in \{1, 2, 3\}$，使得 $\lambda_i \geq \mu_i > 0$。请注意，这样的 i 总是存在的。

2. 注意到每个 i 都满足 $\chi_f(v_i) = i$，并且每个 $u \in [v_i, v_j]$ 都满足 $\chi_f(u) = \min\{i, j\}$。因此，特别是对于每个三角剖分，这种着色是 Sperner 的。3. 假设 $\{D_m\}$ 是一个无限序列的三角剖分，其中 $D_0 = T$，并且 D_{m+1} 是 D_m 的一个细化，使得 D_m 的每个三角形都被分割成面积最多为原始面积 1/2 的三角形.[19] 显然，$T = T_0$ 是 D_0 中的彩虹三角形。根据 Sperner 定理，如果 T_m 是 D_m 中的彩虹三角形，那么在 D_{m+1} 中必须存在某个彩虹三角形 T_{m+1}。因此，存在一个无限序列的 *nested* 彩虹三角形。设 x 为它们的交集。根据 χ、x 和彩虹性质的定义，如果 $x = \alpha_1 v_1 + \alpha_2 v_2 + \alpha_3 v_3$ 和 $f(x) = \beta_1 v_1 + \beta_2 v_2 + \beta_3 v_3$，那么对于所有 i，我们必须有 $\alpha_i \geq \beta_i$。但这样 $f(x) = x$。

19.4.2 Proof sketch of the impossibility theorem

考虑到这个证明，我们最终得到了不可能定理。让我们首先回顾一下关于共识的不可能性定理19.4的证明概要（以及将其适应共享内存模型定理19.5的修改，这两个定理类似），并看看我们如何将其扩展到2集协议。给定一个假设的无等待程序{v*}，证明通过归纳生成了一个无限序列的{v*}配置{v*}。首先，它证明了某些初始配置{v*}是不可决定的（回想一下，Chaudhuri [Cha93]已经为每个{v*}做了这件事）。然后，它证明了，如果任何配置{v*}是不可决定的，那么存在一个有限的调度将{v*}引导到另一个不可决定的配置，{v*}。这是{v*}1（它适用于{v*}1）的难点。有足够的先见之明，注意到在不可能证明中由{v*}生成的无限序列的{v*}配置{v*}的句法相似性，以及将不动点定理简化为Sperner引理的{v*}三角形{v*}的无限序列，这些三角形是{v*}。啊，如果我们能只给这个句法类比加上正确的语义！让我们现在更仔细地尝试，更广泛地简化[SZ00]（但保留其直觉的骨架）。再次，我们在这里为{v*}2做这件事。并且如上所述，你可能想用对更简单的{v*}1（共识）情况的论证来模仿下面的{v*}2，这可能对某些定义有所揭示.{v*}

一个关键思想是暂时忽略配置（其描述取决于我们没有的程序）并专注于调度。这样做实际上并不会造成任何损失，因为固定一个初始配置 C_0 和一个调度（比如，s），程序 π 就可以确定它在执行过程中生成的配置序列。调度与程序无关，并且它们的结构简单。在共享内存

[18] Why is this a *topological* statement? Simply because the same statement holds for any subset of the plane that can be continuously deformed to a triangle. To see this, compose the deformation with the given function, find a fixed point, and invert the deformation.

[19] The refinement and area conditions on the sequence D_m are not necessary for this reduction—we do it to facilitate the next one.

[20] Both papers [SZ00, HS99] give a detailed account of the case $k = 1$ before starting the proof for $k > 1$.

模型，它们可以被视为在$S = \Sigma^*$中的无限序列，其中字母表Σ是所有非空玩家子集（即调度器允许在本步骤中将*write*存储到内存中的玩家子集[21]）S是一个紧集，我们可以想象通过一个连续映射将其映射到三角形T。[22]

问题在于：我们如何做到这一点？下一个关键思想是，在上面的约简中，嵌套三角剖分的顺序是任意的，因此让我们方便地选择它以适应这种映射。使每个细化看起来如图22所示，这可以被视为第一次三角剖分[23]，D_1。在其中，每个区域都由三个[24]玩家中的非空子集标记。现在想象以相同的方式递归地三角剖分每个三角形（使用我将不描述但自然的一致标记）。这产生了无限序列$\{D_m\}$。

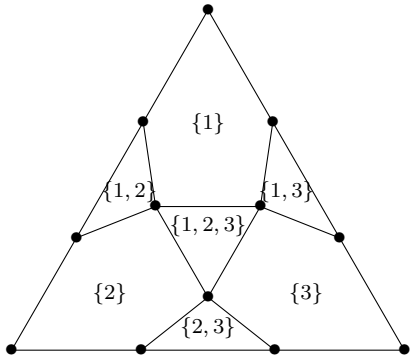


图22. 基本三角剖分，即 D_1。

嵌套序列 $\{D_m\}$ 现在为我们提供了一种自然规范映射 μ，将序列 S 映射到 T。如果 $s = \sigma_1, \sigma_2, \ldots$ 是 S 中的一个序列，它以自然的方式定义了一个三角形序列 $t_1, t_2, \ldots$，其中 $t_m \in D_m$ 以自然方式——如果你在 t_m 中，那么 σ_{m+1} 是 t_{m+1} 内部下一个区域的标签 t_m。令 $\mu(s) \in T$ 定义为所有 t_m 的交集。现在观察这个映射 μ 的一些重要性质。

1. 地图 μ 是 *onto*，因此我们用 S 的元素标记了 T 的点。

2. 虽然映射 μ 是 *not* 双射，但为了简单起见，我们将假设它是，即，μ^{-1} 是良好定义的。

3. 映射 μ 是 *continuous*，在 S 上的自然度量下：如果两个序列 s, s' 在长度为 ℓ 的前缀上相同，那么 $\mu(s)$ 和 $\mu(s')$ 之间的距离至多为 $|\Sigma|^{-\ell}$。

4. 地图 μ 将每个 *face* (a 顶点、边或三角形) 在 D_m 中分配给每个 *finite* 调度 $s \in \Sigma^m$。每个接触此面的三角形中都有 s 的延续。

现在是我们对三角剖分进行三色标记并最终应用Sperner引理来完成证明的时候了。实际上，让我们定义一个对T中所有点的3色标记χ_π，使用程序π，我们假设它是无等待的。取任意的$u \in T$，并让$s = \mu^{-1}(u)$是映射到u (的调度，我们假设它是良好定义的)。根据假设，某个玩家在有限时间内对s上的π做出承诺；这个值是$\chi_\pi(u)$。这定义了

[21] Reads also have to be incorporated, but we ignore this here.

[22] We ignore here a crucial point, which is elaborated in [SZ00], that does not affect our level of description. Namely, that one actually needs to define equivalence classes of schedules in Σ^* to define S, and this gives a different topology than the simple product topology on Σ^*. Roughly speaking, two schedules are *equivalent* if they induce the same sequence of contents of all shared registers.

[23] Imagine that the three pentagons in the "triangulation" are triangles, too. It is another detail swept under the rug, which is better omitted for the intuition. A simple fix is triangulating each pentagon by chords from the appropriate apex of the big triangle.

[24] Which suffices for proving impossibility when $k = 2$.

χ_π 在所有 T 上，特别是在所有三角剖分的所有顶点上。容易看出这种着色是斯佩纳的.[25]

最后，我们可以理解在这个几何图中如何建模配置，并定义 *indecisive* 配置。考虑任何 *finite* 调度 $s = \sigma_1, \sigma_2, \ldots, \sigma_m$，并假设 $\mu(s)$ 位于 D_m 中三角形 T_m 的内部。将程序 π 应用到初始配置 C_0 和调度 s，得到某个配置 C_m，我们可以将其与这个三角形关联。如果这个三角形是彩虹色，则定义 C_m 为不确定。其理由很简单：根据 μ 的定义和三角剖分的标记方案，s 可以以三种不同的方式扩展，因此每种方式都会到达不同的顶点。因此，从 C_m 有三个可到达的配置，分别到达三个不同颜色的顶点，即不同的承诺值。这正是不确定性的表现！

总结来说，因为我们从彩虹三角形 $D_0 = T_0$ 开始，Sperner引理允许我们选择一个嵌套的无限序列彩虹三角形 T_m。它们的交集定义了一个无限调度，由程序 π 从初始配置 C_0 开始并遵循 s 定义的无限序列配置 C_m 都是未决的！因此，没有处理器可以在有限时间内对这个调度 s 进行提交，这与 π 是无等待的假设相矛盾。

19.4.3 General input-output problems and simplicial homology

我们在此继续关注具有故障停止错误的共享内存模型。如上所述，Herlihy和Shavit[HS99]的论文不仅证明了k-集合一致性的不可行性。他们的论文考虑了所有此类分布式任务，即所有*input-output tasks*，这是一个定义极为丰富的集合，大致如下。直观上，它允许以任何方式指定每个输入的合法输出，同时包括参与者失败的可能性。

更精确地说，固定任何输入值的字母表 Σ_I 和任何输出值的字母表 Σ_O。我们假设每个字母表都包含一个特殊的符号 $\perp$ (，它本质上表示一个玩家可以变得不活跃，或者不参与)。将 n 固定为玩家的数量。一个 *input-output task* 是任何满足两个自然封闭性质的关系 $\Delta \subseteq I \times O$：$I \subseteq \Sigma_I^n$ 和 $O \subseteq \Sigma_O^n$ 是在 $\perp$ 符号下的 *monotone* 集合（即，将集合中任何向量的任何组件的值替换为 $\perp$，它仍然在集合中），并且 Δ 是在 $\perp$ (下的 *consistent*，即对于任何 $(i, o) \in \Delta$，如果 $i_k = \perp$，则也 $o_k = \perp$）。

Herlily和Shavit的主要结果是对于每个此类任务，一个无等待并发程序的可能性或不可能性的精确 *characterization*。这种描述完全采用拓扑形式，不涉及任何协议。这使得可以使用拓扑工具来确定每个任务的可能性。以下是一个展示这种能力的简单示例:
A wait-free algorithm for 2-processor consensus exists if and only if there is a continuous onto map from a connected set in Euclidean space to another, disconnected set。后者显然是不可能的，因此前者也是不可能的。

更普遍地，该论文建立了自然拓扑空间，称为*simplicial complexes*、$\hat{I}$，对应可能的输入I和$\hat{O}$对应可能的输出O，使得每个输入/输出向量对应于输入/输出复形中的一个单纯形。现在的问题变成了两个空间之间是否存在足够好的映射。更精确一点（使用未定义的术语）：*A wait-free algorithm for the distributed task Δ exists iff there is a simplicial map from a subdivision of the input complex to the output complex that respects Δ*。

我不会更精确，也不会在这里定义这些术语。但请注意，复形之间的单形映射是拓扑空间之间连续映射的类比，并且提到的细分是之前讨论的三角剖分的扩展。实际上，细分的概念正是捕捉这种语言中的协议，我们在k-集协议不可能性的证明中看到了这种类比。事实上，当将这个一般设置专门化到k-集协议问题时，单形映射的不存在性很容易从Sperner引理（定理19.10）得出。在更一般的情况下，可以使用来自同调理论的工具，这些工具提供了更复杂的拓扑空间*invariants*（如更高维度的连通性和“洞”，同调群等）来证明不存在性。

[25] A vertex v_i of T corresponds to a schedule where only player i ever gets to write, and so π must let it commit to its own value. Similarly, every point on the interval $[v_i, v_j]$ corresponds to schedules in which only these two players participate, and so the committed value must be one of theirs.

所需映射，正如原始论文中确实所做的那样。如前所述，关于拓扑方法强大功能的更多示例和扩展可以在书籍 [HKR13] 中找到。

19.5 Local synchronous coloring

本章以一些好消息结束。在异步模型中，有许多看似不可能的任务，当玩家拥有一个共同的时钟时（特别是，共识和哲学家就餐问题就是这样的问题）变得可行。我们给出了另一个例子，*local coloring*，它非常适合我们，因为它将（一种）重命名问题和处理器环中的计算结合起来，这两个问题我们之前都已经讨论过。但选择这个亮点有两个更好的理由，除了这个便利之外：*locality*和*determinism*。首先，它展示了非常松散连接的分布式系统可以多么出人意料地强大和高效，尽管受限于局部性，单个处理器的视野是短视的。其次，它展示了尽管随机性是我们之前看到的在分布式环境中打破对称性的一个惊人的强大方法，但仍然存在具有类似性质的自然任务，可以通过确定性来解决。

这里是这个设置。我们再次有一组就餐的哲学家 $\{P_i\}$ 围坐在一张桌子旁。每个哲学家只能与其左右邻居进行交流。每个哲学家都有一个初始输入值 x_i，来自集合 $[n]$，这是一个合法的循环着色（即，每个哲学家的初始值与其两个邻居都不同）。请注意，哲学家的数量没有上限；实际上，它可以无限大！尽管如此，输入值允许他们以固定的、有限的速率进食，每个哲学家都确保每 n 步得到一顿饭。怎么做到的呢？因为他们现在有一个共同的时钟，并且以 n 为模进行轮次进行。在时间步 t，每个具有 $x_i = t \bmod n$ 的哲学家 P_i 会拿起两个（可用的！）叉子，吃一个时间单位，然后把叉子放回桌子上。

到目前为止一切顺利，但他们真的很饿，n也可以很大，等待下一餐的时间太长了（更严重的是，在许多应用中，利用共享资源的速率很重要）。能否提高这个速率？当然可以！毕竟，每个周期都可以合法地三色标记，所以知道这种标记的外部方可能已经给出了从{1, 2, 3}的值，这些值可以让每个人每走3步就能吃一次。然而，这是一个分布式环境，没有这样的外部方——他们必须自己保护自己。一个*local algorithm*，其中他们在每一步与邻居交换信息，能让他们以从上述可能的n-色标记开始的速度找到这样的合法三色标记有多快？这正是当前问题的$n \to 3$ *coloring*[26]！

不清楚这样的重命名是否可以在有限步骤中找到。注意，在一个 T 步的局部算法（现在忽略消息大小），每个处理器都可以了解连接它们的通信网络中距离它 T 的每个其他处理器的所有信息（在这个例子中，是循环）。因此，网络的直径是一个上界，对于许多任务，也是一个下界（例如，考虑一个（甚至）循环的 2-着色问题 [Lin92]）。但是，这里我们对网络的大小没有任何限制（对于循环来说，这基本上也是直径）！因此，一个有限算法将不得不依赖于一个小邻域中的值。

一个自然想法确实是使用随机性：让每个 P_i 从 $\{1, 2, 3\}$ 中选择一个随机颜色，并检查其邻居的值。期望中只有常数比例（最多 5/9）的处理器会冲突（即，有相同颜色的邻居）。现在，那些没有冲突的处理器保持原位，其颜色的选择永远固定。处于冲突中的每个处理器再次投掷（在其当前颜色和其邻居未选择的颜色之间，至少留下 2 种可能颜色）。再次，我们期望只有常数比例的它们仍然有冲突，因此它们可以重复，直到没有冲突。这个想法似乎打破了直径界限：如果有 N 个处理器在循环中，那么我们期望在约 $\log N$ 步骤后终止，期望中（甚至实际上）以高概率。但是这个算法有几个缺点。首先，N 可能非常大。其次，它们没有明显的办法知道没有剩余的冲突，也就是说，哲学家知道过程已经结束并且他们可以开始吃饭的时间。最后，他们没有使用他们的输入——这能有所帮助吗?

[26] It is a renaming problem (as the processors choose different names) from a smaller palette, but now the consistency requirement is of a different nature.

[27] One can, and we will, consider $n \to m$ coloring for any $m < n$.

一个由Cole和Vishkin [CV86]提出的巧妙局部算法展示了他们可以做得更好，*deterministically*。实际上，所需的步骤数量几乎是一个常数！

Theorem 19.12 [CV86]. *There is a deterministic local algorithm for the $n \to 3$ synchronous renaming problem that takes $O(\log^* n)$ steps.*[28]

想法很简单。让我们看看处理器如何在一步中将颜色集合从 n 缩小到 $2(1 + \lceil \log n \rceil)$。将名称视为 k-位序列，对于最小的 k，使得 $n \leq 2^k$。考虑任何具有输入值 x 的处理器 P，其（不同于它）的相邻值在左侧是 y，在右侧是 z。让 i 是 $x_i \neq y_i$ 和 j 的最左侧位，其中 $x_i \neq y_i$ 和 j 是 $x_j \neq z_j$ 的最左侧位。现在，这些信息足以让每个处理器创建一个更短的价值，同时保持这些值形成一个合法着色。新选择的价值 P 是 x'，简单地定义为 $(i, x_i), (j, x_j)$，长度为 $2(1 + k)$，正如所需。我将验证这是否是一个合法着色的工作留给您（稍微微妙）。现在，迭代此过程 $2 \log^* n$ 步，结果得到某种 C-着色，具有某种 $C < 20$。在 C 更多步骤中，进一步将其减少到 3-着色是另一个练习。

此结果对于上述应用很重要，同时也是一种通用的技术，可以在多个步骤中确定性地在网络及其直径几乎独立的情况下实现某种局部（对称性）破缺。

一些由这项工作引发的自然问题由林尼亚尔[Lin92]解决。首先，是否可以比上述算法做得更好，也许在常数步数内解决这个问题？其次，是否可以处理除了循环以外的其他通信网络？第三，是否可以使用随机性来改进这些算法中的任何一个？我们以林尼亚尔的回答（否，是，否，分别）作为结论：

Theorem 19.13 [林92]。

- *Any local algorithm for the $n \to 3$ coloring problem on the cycle requires at least $\frac{1}{2} \log^* n$ steps.*
- *There is a local algorithm solving the $n \to O(d^2)$ coloring problem on any graph of maximum degree d in $O(\log^* n)$ steps.*[29]
- *If a probabilistic local algorithm solves a coloring problem in T steps in expectation, then there is a deterministic T-step local coloring algorithm for the same problem.*

[28] All logarithms are base 2. Recall that the $\log^*$ function is the inverse of the tower function, namely, $\log^* n$ is the smallest number h such that a tower of 2's of height h exceeds n, or equivalently, the smallest number of times h of logarithms we need to take, starting at n, to get below 2. This function grows extremely slowly. For example, if n is the number of atoms in the universe, estimated by 10^{80}, then $\log^* n = 4$.

[29] And thus, also an algorithm that in $O(d^2 + \log^* n)$ steps solves the $n \to d + 1$ coloring problem.

20 Epilogue: A broader perspective of ToC

这本书主要关注计算复杂性，既作为一个主要的数学领域，描述其丰富的内部结构和其广泛影响其他数学领域，又作为计算理论（ToC）的核心、关键领域。本章致力于对ToC的全面概述，它远远超出了数学和计算机科学的交集。技术对社会的影响往往掩盖了ToC所依赖的智力基础，有时甚至从专家那里也是如此。即使对于许多了解和理解它们的人来说，这些理论思想通常只被视为技术发展的仆人。在本章中，我详细阐述了ToC作为一个独立的知识学科的属性，总结如下。

The theory of computation, since its inception by Turing in 1936, is as revolutionary, fundamental, and beautiful as the great theories of mathematics, physics, biology, economics... that are regularly hailed as such. Its impact has been similarly staggering. The mysteries still baffling ToC are as challenging as those left open in other fields. Moreover, the ubiquity of computation makes its theory central to all other disciplines.

In creating the theoretical foundations of computing systems, ToC has already played, and continues to play, a major part in one of the greatest scientific and technological revolutions in human history. But the intrinsic study of computation transcends human-made artifacts, and underlies natural and artificial processes of all types. Its expanding connections and interactions with all sciences, integrating computational modeling, algorithms, and complexity into theories of nature and society, is at the heart of a new scientific revolution!

本章各节从高层次讨论了目录（ToC）的各个方面，主要涉及智力方面，但也包括社会和教育方面。本阐述旨在描述该领域的性质和范围、迄今为止的演变以及我期望它在智力和社会领域发挥的重要作用。我希望它能为新来者和有志之士，以及计算机科学家提供一个有用的鸟瞰视角。本阐述显然受到了我的个人经验、观点和有限知识的偏见。

Organization 第一几个部分讨论了目录与其他许多学科之间的联系和相互作用。我首先简要讨论了它与“母”领域，即计算机科学和工程（CS&E）以及数学的相互作用。然后，我详细描述了与“邻近”领域，如优化和编码理论的相互作用。最后，我详细讨论了与更“遥远”的科学，如生物学和经济学，以及与哲学和技术的相互作用。

在相邻学科和远程学科之间的章节中，我们将进行两次重要的偏离。一次将讨论一个非常*general definition of computation*，这将阐明ToC的广泛使命以及它如何与所有科学的基本使命自然对齐。另一次将给出*central tenets of ToC methodology*的一般描述，这将阐明ToC可以为这些与科学的联合合作带来哪些新方面。

在讨论了联系和相互作用之后，我将转向ToC的一些高级挑战，然后简要讨论ToC研究的作用和重要性以及研究人员对K-12教育的影响。最后，我将谈谈对该领域社会学术特征的几点个人观察，我认为这部分解释了其迄今为止的显著成功，并应保留以促进未来的成功。

对科学领域的历史、演变、思想、贡献和挑战进行盘点，在存在数百年的领域中非常熟悉。从这一观点出发，已经撰写了大量半普及书籍，涉及数学、物理、生物学和其他领域。从图灵的论文开始计算，ToC已经存在了80年（其中我亲眼见证了最后40年）。我相信，关于ToC的书籍，将详细描述和论证其成就、目标和在学术领域的独立知识地位，显然有很大的空间。此外，我相信，对领域成员、学术界和公众进行此类教育对于该领域的未来发展至关重要。在这里，我将我对该领域的处理方法浓缩为一章。[1] 除了简短之外，这项调查也只是一个时间胶囊。我完全

[1] Twenty years ago, Oded Goldreich and I gave a similar treatment [GW96] of the nature and future of the field, with the same conclusions as to its success, promise, and independence, but one that was much shorter and had far less detail. The two decades that have passed, and all the additional research ToC has created since, make it easy for me to add much more supporting evidence for these claims.

预计角色计算在我们生活的各个方面所扮演的巨大扩张率可能会改变并扩大ToC的范围、目标和使命。由于计算无处不在，这种扩张的潜力是无限的。

在整个本章中，“ToC研究”将指对理解所有形式的计算的追求，而不是指执行该研究的学者的学科归属；尽管这些学者通常与ToC有关联，但此类显著的贡献也由应用计算机科学家、数学家和所有学科的科学家做出。

20.1 Close collaborations and interactions

很难在几页内公正地展现ToC与其“邻近”学科的广泛和多样的互动，这些学科都远大于ToC。我不尝试全面描述这一庞大的工作体系。相反，我试图捕捉这些互动在每个不同学科中的不同性质和作用，以及它们如何反过来塑造和重塑ToC及其影响范围。我从最接近的学科开始，即数学和CS&E（计算机科学与工程），然后继续进入邻近领域。在这里，以及当我们转向与更遥远领域的互动时，覆盖范围和详细程度并不一致。

Meet the parents: CS&E vs. mathematics 如第1章所述，ToC诞生于并生活在两个截然不同的传统中，这些传统塑造了其独特的性格。我认为这种基因和环境组合极其幸运！以下是一些例子。CS&E是一个年轻且急躁的领域，而数学则是古老且耐心。CS&E的研究方向主要受实践和实用性的驱动，而在数学中，它们主要受美学和抽象的驱动。CS&E要求研究真实、运行的计算机系统，而数学要求任何研究都必须完全严谨，但允许想象力自由驰骋，不受现实的限制。单独来看，两者结合在一起为ToC产生了无数建模和算法挑战。在这两方面，我都会相对简短地讨论。

20.1.1 Computer science and engineering

计算机行业围绕我们的巨大技术成就使许多人盲目，不仅包括普通人和政治家，还包括科学家，他们被纯粹的智力奇迹所吸引。除了创建这些技术背后的许多理论之外，ToC 还继续为所有计算机专业人士及其解决的问题构建概念基础设施。最基本的是，这包括精确地模拟各种计算环境、正式定义资源以及在算法组织和处理信息时分析其利用的能力。ToC 为研究用户获取知识（从而可以评估隐私和保密）提供了新颖、有用的定义和方法；随机性和交互的力量和限制；具有许多组件的系统组织、利用和分析，以及许多其他计算环境，其中一些在本书中详细讨论。这些概念贡献，其中一些具有革命性的实际影响，已经成为基础本科教育中根深蒂固的思维方法，以至于它们常常被视为理所当然。

理论对CS&E活动的具体贡献得到了很好的记录，因此我将简洁地陈述，仅限于列举。ToC创建了并仍在发展基础理论，这些理论是各种计算系统发展的重要基础。在许多情况下，理论先于实践，并促进了系统的发展.[2] 在许多情况下，为特定技术而产生和发展的理论思想，在那种技术过时后仍然有用.[3] 这里是一个理论的部分列表，其中每个成员都深刻、广泛，占据了大量的教科书（人们应该研究哪些以全面了解）

[2]Examples abound, and I give only two primary ones. The first is the early theoretical blueprints of the computing machines of Turing and von Neumann, which unleashed the digital age. The second is the theoretical foundations of public-key cryptography, which enabled potential e-commerce applications, that in turn unleashed the rapid expansion of the Internet.

[3]Again, examples abound, and I give two representatives of two phenomena. First, theories: The theory of communication complexity, was designed primarily to understand area-time trade-offs in VLSI chips and has gone on to be key in numerous other areas, as we discussed in Chapter 15.2. Second, algorithms: The algorithm for pattern matching, designed initially for text processing, was (together with its extensions) essential for the Genome project.

欣赏其背后的理论基础）。它从图灵的计算和算法定义开始，接着是细胞自动机理论、有限和无限自动机、编程语言、系统验证、数据库、计算复杂性、数据结构、组合算法、密码学、分布式系统、软件工程、随机性和伪随机性、计算学习、近似算法、并行计算、网络、量子计算、在线和实时算法，等等。不用说，计算机科学和技术快速发展的世界不断为建模提供许多不同的情况以及需要解决的问题，例如针对大型数据集的最近子线性算法；科学和公共数据库的差分隐私；云计算中的委托；传感器网络中的群体协议；当然，还有对机器学习程序（如深度网络）的理论解释的斗争。未来的技术和应用无疑将导致新的理论、更多的互动以及更好地理解计算的实质、适用性、力量和限制。

20.1.2 Mathematics

数学中，交互作用的本质发展不同。我还会简短地总结，因为这本书中有很多例子，特别是第13章中断章专门讨论了其中的一些。早期，除了与逻辑的早期亲密合作，ToC主要使用数学技术和结果。主要处理离散对象，因此最初的交互大多与组合数学有关。但随着ToC的扩展和深化，其多样化的理论需要来自不同数学领域的工具，包括拓扑学、几何学、代数、分析、数论和代数几何等。这些工具往往并不容易获得，甚至根本不存在，因此不得不开发。这形成了许多合作，导致了这些领域的纯数学结果。新的问题来源（以及进一步工具的需求）来自为不同的计算现象构建的全新的数学模型，正如本书的一些章节所展示的。这些活动与将数学算法化的元挑战相结合；即*efficiently*寻找通过某些间接或非显式论证证明其存在性的对象。这些问题已经渗透到数学中，在某些情况下已经持续了几个世纪。追求这些问题，特别是使用新发展的算法和复杂性工具，导致了重新思考和扩展许多领域，发现了新的结构，并提出了许多新问题。这些与数学许多领域的快速扩展合作极大地丰富了ToC，并阐明了其作为重要数学领域的独立角色。看到大多数年轻数学家对计算复杂性的基本概念、结果和主要未解决问题了如指掌，就像他们对任何其他非专业数学领域一样，令人欣慰。我毫不怀疑，算法和复杂性思维方式将进一步渗透到数学的所有领域，并将创造许多与其他领域的新交互。

Meet some neighbors: Optimization, coding and information theory, statistical physics 我现在提供更多关于ToC与三个与计算机科学自然非常接近的、日益增长的大型学科之间的互动的细节和参考文献：优化、编码和信息理论，以及统计物理学。这些联系已经非常多样化，在某些情况下，它们的起源也令人惊讶。我发现这一系列联系在广度和深度上令人惊叹。再次强调，这里提供的主题列表和参考文献只是部分内容。

20.1.3 Optimization

大型优化社区与ToC的联系远非令人惊讶，因为高效算法是两者共同研究的核心对象（尽管最初每个社区关注的课题和工具类型略有不同）。然而，可能没有预料到的是，主要突破，这些突破大多源于*computational complexity*考虑和结构，将如何丰富和振兴对算法的功率和限制的研究。本书讨论了许多这些内容，包括：

- PCP定理，导致近似难度理论，类似于（尽管比$\mathcal{NP}$-完备性理论更难、更精细）的近似难度理论。里程碑包括[FGL+96,AS98,ALM+98, Kho02, Rag08]，其中一些发展在4.3节和10.3节中进行了讨论。值得注意的是，PCP定理的起源和导致的方法与优化相去甚远：它们包括密码学、交互式证明、编码理论、程序测试和平均情况复杂性。

- LP和SDP层次结构的功率、限制和连接，存在于优化文献中的强大算法范式，这些范式（并且仍然是）随着像[Gri01b, Gri01a, ABL02, BS14, LRS14, CLRS16]这样的工作而变得更加清晰。导致这些发展的某些起源和联系包括证明复杂性、计算学习理论和基于电路复杂性的随机限制。

- 相关且更具体于上述项目的是线性规划的“扩展”，即添加变量以减少不等式的数量。这种技术的确切效力在[Yan91]中得到充分描述，其局限性首次在[FMP+15]中确定（然后在后续工作中进一步发展，如前一项所述）。有趣且令人惊讶的是，一些关键见解和方法来自看似无关的领域，包括通信复杂性和量子信息理论。

- 连续优化方法，可能从Karmarkar算法[Kar84]开始，用于线性规划。最近，动量显著增加，椭球法、内点法、交替最小化、乘性权重以及各种一阶和二阶下降方法（以及它们的应用）得到了极大的扩展：[ST04a,CKM+11,AHK12, Mad13, LS14, KLOS14, AZO14, GGOW15, AZH16]。在许多情况下，这些方法打破了数十年来保持的效率记录，其中一些导致了近线性的算法。联系和工具范围从电流量和随机游走到永久近似、谱分解和泛函分析。

- 指数时间假设（ETH）。这个假设被提出作为一种比$\mathcal{P} \neq \mathcal{NP}$更自然但更强的困难假设，如[IPZ01, IP01]中所述。随之而来的活跃理论，通常被称为*fine-grained complexity*（，因为它引入了更精细的优化问题之间的归约概念），能够预测类$\mathcal{P}$中问题的精确复杂性，并且特别排除了线性时间算法（例如，参见综述[Wil15]）。

- 平滑分析。在[ST04b]中引入，它提供了一种根本新的分析启发式方法，与所有先前的平均情况模型非常不同，并用于以新的方式解释它们在“典型”输入上的成功。

20.1.4 Coding and information theory

目录与广泛的编码和信息理论领域的联系极为广泛，从普通到非常意外的都有。大多数预期的联系都与计算（在计算机、计算机网络、数据库等）处理受噪声和错误影响的信息这一事实有关，因此必须设计成容错。这自然促使使用纠错码和信息理论。不太预期的联系包括将*locality*引入纠错码（以几种方式）、复兴几个旧的想法和模型（如列表解码、低密度奇偶校验码以及编码和信息理论中交互作用的作用），并提出新的问题、应用、技术和结果。以下列出并引用了一些这些内容（许多内容在书中讨论过，尤其是在讨论交互作用的章节，特别是第15.3节和第10章）。

- 列表解码（即，如果解码被放宽以生成一系列潜在消息而不是唯一的消息，则代码可以容忍更多的噪声）的想法可以追溯到编码理论的早期阶段[Eli57]。四十年后，从Sudan [Sud97] 的突破性工作开始，包括[GS98, PV05, GR08b] 的一系列工作迅速导致了具有高效编码和解码算法的最优列表解码码，以及针对许多不同代码的列表解码研究。

- 与之一致，研究 *local decoding* 的想法源于密码学、去随机化和 PCPs 中的复杂性理论考虑。在局部解码中，应仅从原始消息的噪声编码中恢复一个（指定的）比特，但只能检查其中的一小部分（随机），并且允许存在小的错误。在 [GL89,STV99,KS09,Yek12,Efr12,DSW14b,DG16] 和许多其他论文中出现了显著的结构和应用。作为一个应用的例子，[GHSY12] 和其追随者的相关 *locally repairable codes* 对互联网数据分布式数据中心的设计产生了巨大影响，在这些规模和速度下，从故障中恢复和多个副本之间的一致性成为了一个全新的挑战。局部解码（通常与列表解码相结合）的许多复杂性理论应用可以在综述 [Sud00] 中找到，以及许多其他应用，在代数复杂性、组合几何和信息检索的文献中有所体现。

- *Local testing* 代码是另一个从程序测试、交互式证明和PCPs的计算复杂性考虑中产生的新编码问题。在这里，人们简单地询问一个给定的单词是否在给定的代码中，或者非常远离它（在汉明距离上），再次仅通过检查它的一小部分（随机）即可。再次，可以在[BFL91, BLR93, GS00, RS97, AS03, GS06, IKW12, KMRZS17, DK17]中找到令人瞩目的构造和应用。

- 在反向方向，编码理论中为构建显式、高效的码而发展的 *concatenated codes* [For66] 的思想，启发了PCPs和相关系统中的许多证明组合构造。其他鲁棒性的概念以及实际构造也被直接和间接地从鲁棒码中的错误恢复借用到鲁棒证明中的错误恢复。

- “基于图”或LDPC（低密度奇偶校验）码最初在开创性工作[Gal62]中提出并研究。然而，这个方向在30年后才因兴趣和成果的激增而重新活跃起来，其中很大一部分是由于第8.7节中讨论的*expanders*（的兴起），这些图在ToC中有多种应用。这些早期工作包括[Spi95, SS96, LMSS01, RU01, Lub02, CRVW02]。

- 交互计算一直是信息论的一部分。尽管如此，通信/信息复杂性和交互编码理论已经出现并发展成为两个成熟的理论 *within* 计算复杂性，在领域内有重大应用。这些理论极大地丰富了与编码和信息论之间的互动。这两者都在第15章中进行了描述。

20.1.5 Statistical physics

似乎令人惊讶，我将统计物理学作为ToC的邻近领域，但两个领域之间很早就发现了许多自然联系和共同兴趣，并导致了显著的协作（这些协作自然包括离散概率和组合数学）。此外，统计物理学中的许多问题和方法是自然算法性的，正如我们马上将看到的。我将简要地概述这个共同基础，以便更好地欣赏新的相互作用（这使得本节比前两节略长）。

统计物理学的中心主题是理解大系统的全局性质是如何从其部分之间的局部相互作用中产生的。一个常见的例子是物质的 *global* 状态：气体、液体、固体以及它们之间的转变（例如，冰变成水然后变成蒸汽）在温度变化下，这会影响分子的运动和相邻分子之间的 *local* 相互作用。另一个常见的例子是在电场影响下的磁化。一个系统通常由一个图来描述，其节点代表相互作用的部件（每个部件都可以处于有限数量的状态之一）以及其边代表局部相互作用结构[4]。这些给出了每个全局

[4]This (discrete) formalism captures a large number of models of matter of all types, including spin systems on glassy and granular matter, electromagnetic systems (both classical and quantum), “hard-core” interactions, and percolation in porous materials. Many other variants we will not discuss include continuous settings like heat baths, Brownian motion and other diffusive systems, and nonlinear interactions, for example, the Maxwell-Boltzmann ideal gas model (which gave birth to statistical physics), and billiard models.

声明一个权重，或 *energy*，自然诱导系统全局状态的概率分布，称为 *Gibbs distribution*。确定系统的全局属性（平均能量、长程相关性、相变等，都具有物理意义）通常被简化为根据吉布斯分布的 *sampling* 系统状态。这个问题是一个自然的计算问题！然而，请注意，状态的数目与系统的大小（部分数目）呈指数关系，因此是非平凡的。

不难看出，这种离散形式主义自然地捕捉并确实推广了*constraint satisfaction problems* (CSPs)，例如3 $-SAT$，这在第4.3节中进行了讨论。此类系统（交互图和局部能量函数）的描述构成了许多算法优化问题的输入数据。但通常，人们不是要求采样随机状态，而是找到（精确或近似）优化能量的状态。同样，在组合数学中，相同的数据被视为许多枚举问题的输入，即计算具有最优（或特定）能量的状态数量。结果发现，所有这些计算问题*sample, search, count*都是相关的，这自然促进了互动，我们将在后面讨论。

采样方法起源于20世纪50年代的冯·诺伊曼和乌拉姆，部分是为了曼哈顿计划而开发的，这就是所谓的 *Monte Carlo*算法，后来进一步发展。在上述框架中，一个人在系统的状态上设置一个指数级大的、隐式描述的*Markov chain*，其平稳分布是吉布斯分布（这被称为*MCMC method*，对于马尔可夫链蒙特卡洛）。然后，从一个任意状态开始，在链上进行自然的随机游走，希望快速收敛到一个典型状态，如所需。这种类型的自然算法包括Metropolis算法和Glauber动力学。为了确定许多模型的特征，进行了许多此类链的模拟。然而，对于少数系统，几乎没有工具可以严格量化这些算法的收敛速度，因此使用了启发式论证（以及资源限制）来截断模拟时间。不用说，如果随机游走没有收敛，关于系统全局状态的信息可能是完全错误的。当然，MCMC可能不是唯一的采样方法！当该领域在20世纪70年代开始研究*probabilistic algorithms*时，与目录的交互开始，MCMC是一个完美的例子。这导致了关于上述局部相互作用系统问题集的重大进展，我简要总结如下：

- Valiant的计数问题复杂度理论[Val79b, Val79c]引入了复杂度类$\#P$，并建立了布尔矩阵的永真值是该类完备。这使我们能够证明几乎所有自然枚举问题的困难性，以及与它们相关的吉布斯采样问题中计算概率的困难性（实际上，已经建立了区分困难计数和容易计数的二分法——参见综合书籍[CC17]）。

- 为了使事情变得容易接受，Jerrum、Valiant和Vazirani [JVV86] 证明，对于大多数有趣的问题，*sampling* 等价于 *approximate counting*。这提供了一个关键的联系，使得采样算法可以应用于解决枚举问题。

- 一系列由Jerrum和Sinclair [JS89,SJ89]撰写的论文提供了*canonical paths*和*conductance*的一般方法来界定一般马尔可夫链的收敛性，这些作者使用这些方法严格证明了统计物理中诸如伊辛模型和单体二聚体模型等问题的多项式收敛性，以及计数图中的匹配等枚举问题。这些论文引发了一系列后续研究，发展了这些和其他方法，包括使用各种耦合形式的严格方法。许多早期例子在[Jer96]中得到了总结。

- Jerrum、Sinclair和Vigoda的重要工作[JSV04]给出了一种多项式时间概率算法来近似非负矩阵的永真值，通过完备性（上述）捕获了许多其他枚举问题。尽管只有指数因子精度[LSW00,GS14]，但用于永真值的有效*deterministic*算法揭示了这些问题与矩阵缩放和双曲多项式之间的联系。这些方面以及更多内容在Barvinok关于组合数学和分函数复杂性的书籍[Bar16]中得到了解释。

- *spatial*与局部系统的结构特性（例如，吉布斯分布中的长程相关性、相变）以及*temporal*，复杂性理论特性（例如，自然马尔可夫链（如Glauber动力学）的收敛时间）之间的联系自20世纪70年代以来已被物理学家在自旋系统中研究。这一联系通过Weitz [Wei06] 对*hard-core*模型的研究得到了扩展；他开发了一种*deterministic*算法（与马尔可夫链方法非常不同），该算法在相变之前是高效的。这通过Sly [Sly10] 在相变之上的一项硬度结果得到了补充。这引发了进一步的协作和对空间和时间混合之间这种深层关系的更好理解（参见[DSVW04]以获取综述）。

- *Lovász local lemma* (LLL)使我们能够建立罕见“全球”事件的发生率。LLL的有效算法版本由Beck [Bec91] 提出，从Moser [Mos09]（以及随后[MT10]）的工作开始，导致了罕见事件的近似计数和均匀抽样版本（例如，参见[GJL16]）。这些分析*directed, nonreversible*马尔可夫链的技术是许多更多应用的有力新工具。Moitra [Moi16] 在LLL区域提出的完全不同的确定性算法承诺了更多应用；即使在解空间（以及因此自然马尔可夫链）不连通的情况下，它也能工作。

- 最后，像Metropolis算法这样的马尔可夫链，在能量等优化目标的引导下，已被用作启发式算法中的优化工具，如*simulated annealing*，以生成有利于高目标值状态的Gibbs分布。其中一些技术允许分析经典特定优化问题的收敛时间（例如，参见[JS93]关于上界和[Jer92]关于下界）。

20.2 What is computation?

随着我们现在从与邻近领域的交互转向（看似）更遥远的领域，我们采取更高的视角。在这个时候，我应该首先解释一下，因为我们讨论计算理论，我们所说的术语*computation*是什么意思。定义计算的最广泛方式之一——实际上，支撑著名*Church-Turing thesis* (的观点，稍后还会讨论到，是这样的。

Computation is the evolution process of some environment,
by a sequence of “simple, local” steps.

如果这个定义在你看来几乎捕捉了你所知道的任何自然过程，这正是我希望它听起来那样！

当然，这个定义要求指定演化的环境和区分局部与全局、简单与复杂的概念粒度。最基本的设置（其中“*computation*”一词最初产生），是当*bits*在图灵机或布尔电路中演化时；这是粒度概念的一个具体例子，以及演化的规则/步骤。另一个，仍然是一种实际的计算，但具有完全不同的粒度和基本步骤，出现在网络中处理器*states*的演化中，在（比如说）成对通信的情况下。当然，还有许多其他例子，这些例子捕捉了（现有或想象中的）计算系统。

主要观点是，将过程视为计算的观点扩展到许多其他环境，这些环境与计算机相去甚远。在每个环境中，许多不同的粒度选择和简单本地规则（这些规则可能由自然给出或由我们创造），将导致不同的进化过程。所有这些（即使是物理的）都值得研究，从计算的角度来看，使用我们在第20.3节中阐述的ToC方法，即*information processes*。

以下是一个具有此类相互作用部分的环境的部分列表，在所有情况下，它们都可以摆脱其物理特性，被视为纯粹信息的转换：

- 计算机中的比特。
- 计算机在网络中。

- 物质中的原子。
- 大脑中的神经元。
- 蛋白质在细胞中。
- 组织中的细胞。
- 细菌在培养皿中。
- 减法在证明系统中。
- 市场价格。
- 个体在种群中。
- 星星在星系中。
- Facebook上的朋友。
- 量子纠缠态中的量子比特。

所有这些地方都是计算理论家[5]可以发挥作用的地方！对所有这些过程所消耗的资源、它们“计算”的内容以及这种演化的其他属性进行计算建模和定量研究，是计算理论（ToC）的日常工作。

这些以及许多其他类似例子阐明，计算的观念远远超越了其对（自然和本质）计算机技术的相关性，并证明了计算理论的需求，即使计算机根本不存在也是如此！

20.3 ToC methodology

我们已看到，许多自然和人为现象向我们展示了我们希望理解的进程，而许多实际和智力重要性的问题则向我们提出了需要开发有效方法来解决它们的必要性。ToC 已经创建了一种强大的方法和语言，用于研究这类问题。以下列出了一些其（相互关联的）重要原则，我们在本书的前几章中反复看到它们在行动中的表现，并应在本章所讨论的计算的一般背景下加以考虑。让我强调，这些原则中的大多数并非新颖：它们在科学和数学的许多研究中已经使用了很长时间。然而，我认为在 ToC 的背景下，它们被以更系统的方式进行处理，在某种程度上，这些原则本身也成为该领域的研究对象。我预计，在数学和其他科学中更系统地使用这些原则将会很有成效，因为现有的相互作用（其中一些将在以下章节中讨论）已经揭示了这一点。

1. **Computational modeling**: *Uncover and formally articulate the underlying basic operations, information flow, and resources of given processes*。这本书有很多关于具有各种基本操作（如布尔或算术门）的计算过程的例子，这些操作可以是确定性的、随机的或量子化的。例如，我们已经看到了在证明中的几何、代数和逻辑推理。在所有这些过程中，我们都研究了时间、空间、通信、随机性和其他资源。在计算机科学和数学之外，存在着大量使用不同资源的自然过程，其中许多可以被看作是信息过程。通过建模它们的基本步骤和资源 *as* 计算，并应用计算语言、方法和结果，在科学领域可能极为有益。此外，对自然过程的抽象、计算理解可以通过整合自然界使用的算法和硬件反馈到计算机技术中，正如纳米计算、量子计算、DNA计算、群体计算等初始尝试所承诺的那样。

[5]And these theorists can come from every discipline.

2. **Algorithmic efficiency**: *Attempt to minimize relevant resources used by computational processes and study their trade-offs.* 首先必须强调，这一原则同样适用于由人类设计的用于解决问题的算法、在大量数据上训练以自我改进的深度网络，或经过数亿年进化以指导生物体行为的算法。总的来说，节约资源是首要的（甚至无生命的物理对象似乎也更倾向于低能量状态）。经验表明，研究效率的局限性和权衡是问题和过程的极好分类指南。此外，开发通用的分析工具以在一个计算环境中找到效率的局限性和权衡，在其它环境中可能极具威力。

3. **Asymptotic thinking**: *Study problems on larger and larger objects, as structure often reveals itself in the limit*。有一种自然、实用的倾向，即关注理解（或开发针对）我们真正关心的具体、特定对象（例如，人类基因组、Facebook图、美国路线图、鲁比克魔方、费马最后定理的证明）的算法。然而，将这些具体对象视为无限家族的一部分，其中相同的进程或算法适用，可能会导致处理这些特定对象更加高效和更通用的方法。该领域的许多主要成功源于应用渐近观点，并且许多算法、缩减和复杂度类清楚地证明了这一点。这种方法在其他领域也有类似之处。在编码理论中，香农的渐近方法彻底改变了数字通信和存储（尽管所有技术应用的参数都非常具体有限）。在物理学中，这种方法通常被称为 *thermodynamic limit*。在数学中，渐近视图在离散设置中很自然，如集合系统和图（尽管连续结构的某些参数也被视为渐近的，例如维度、 genus 和连续性本身）。使离散结构连续并研究各种极限可以揭示隐藏的结构；也许可以更多地应用于计算的研究。组合对象极限理论的最近发展（例如，参见全面书籍[Lo v12]）是一个有希望的例子。

4. **Adversarial thinking**: *Prepare for the worst, replacing specific and structural restrictions and constraints by general, adversarial ones—more stringent demands often make things simpler to understand!* 在目录表中反复出现的一个主题是，当手头的模型允许更强的（而不是较弱的）对手时，会发现令人惊讶的算法和协议。[6] 虽然看似反直觉，但这种劣势（或最坏情况下的观点）往往能揭示可能被具体细节所掩盖的想法。显然，如果存在这样的普遍性，正面的结果（即上界和算法）是非常受欢迎的。当然，同样经常的是，在一般的对抗性设置中找到下界。这样的负面结果要求制定更具体的假设，在这些假设下，我们关心的这个问题确实有解。但即便如此，如何限制对手和避免困难，从理解算法在更普遍的对手存在时如何以及为什么失败中，往往可以找到指导。特别是密码学理论，其中对手通常仅限于他们的计算能力，由于这种方法非常成功，这与计算复杂性理论完美契合。

5. **Classification**: *Organize computational tasks into (complexity) classes according to the amounts of various resources they require in various models*。分类在科学中是自然的，但通常是基于对象的结构属性。在计算复杂性中看似令人惊讶的是，大量且种类繁多的问题如何紧凑地适应相对较少的由资源需求定义的复杂性类别，以及由不同资源定义的类别之间的强烈联系。当然，该领域的核心开放问题是大多数关于证明某些类别的成对实际上是不同的。可以想象这种基于计算复杂性的分类在其他科学（例如，自然分布式系统中的同步或协调过程，或在数学中群和代数以及流形上的词问题和对偶问题）中的潜在力量。

[6] These adversaries can be inputs, distributions on inputs, schedulers, noise, eavesdroppers, and so forth, depending on the context.

6. **Reductions**: *Ignore your ignorance, and even if you can't efficiently solve a problem, assume that you can, and explore which other problems it would help solve efficiently*。尽管哲学家们就“还原论”在理解世界方面的力量展开了激烈的争论，但通过将复杂现象分解为更简单的部分来理解，通常会导致重大的科学发现。在计算机科学中，一种标准的编程实践要求在解决更复杂问题B的算法中使用子程序作为某个已解决的问题A的一部分。算法技术对这种简化的重要应用。ToC的主要转折在于使用简化来证明*hardness*以及易用性：在上面的例子中，不仅A的易用性意味着B的易用性，反之亦然，B的难度意味着A的难度。这些关系产生了各种难度概念下的计算任务的偏序，这通常也关联了不同的模型。在密码学和伪随机性等领域，简化（本身也是算法）的复杂性和精致性被提升到了一种艺术形式，其中除了效率之外，许多其他属性也限制了过程A和B。在近似难度简化和PCP构造中，出现了不同的、数学上丰富的精致性。我坚信，当系统地使用简化的语言来关联不同的问题和模型时，在数学和生物学等领域可以揭示出更多的结构和联系。

7. **Completeness**：*Identify the most difficult problems in a complexity class.*[7] 结合上述第5和第6项，以下现象最初令人惊讶，而现在已成为一种自然预期:
whole complexity classes of problems that can be solved in certain limited resources and environments, can be "captured" by a single "complete" problem in the class。这种*completeness*承诺（通过缩减）表明，针对该单个问题的更好算法将立即带来对该类中所有其他问题的相同改进。反之，要区分一个类别与另一个类别，只需证明该单个问题的难度即可。对于这两个方向，当研究整个类别时，可以自由关注任何完整问题。一些完备性的概念，特别是$\mathcal{NP}$-完备性，在许多学科中表现出非常普遍的现象。在数学和科学中寻找更多此类现象的例子，特别是其他概念，将非常有趣且有用。

8. **Hardness**: *Prove intractability results—these are useful!* tractability结果（即有效算法）的潜在效用是显而易见的。然而，知道一个任务困难（对于某些模型和资源没有有效算法）同样有用！它可以在众多实际和科学应用中建议改变模型和任务的定义。而且正如我们所看到的，困难问题可以直接用于产生积极的应用，如在密码学和伪随机性方面。最后，尝试证明困难失败的尝试已经提出了令人惊讶的算法，这些算法可能在其他情况下无法被发现（一个著名的例子是Barrington的算法[Bar86]）。当然，困难通常是很难证明的，条件困难是次优选择；在这种情况下，人们自然会努力最小化和简化所需的假设。

9. **Barriers**：*When stuck for a long time on a major question, abstract all known techniques used for it so far, and try to formally argue that they will not suffice for its resolution*。内省一直是计算复杂性理论的优势，对领域内主要问题进展障碍的正式研究已成为该领域本身的一部分。有趣的是，这些研究往往导致对证明技术（尽管这些是我们研究人员开发的，而不是我们研究的计算模型）的*computational*特征描述。这导致了一些证明和计算之间意想不到的联系，以及一些令人惊讶的无条件结果，例如，不存在对整数分解困难性的*natural*[8]证明。障碍不仅作为解释（或借口）失败解决主要开放问题的原因——它们还可能引导我们开发新的、不同的技术来绕过它们。这已经在过去发生过，无论是下界障碍（如对角化）还是上界障碍（如线性或半定规划的对偶间隙）。对于证明，看到数学上的障碍类比将会很有趣，对于

[7] Namely, those which all other problems in the class reduce to in the sense above.
[8] In the formal sense of Razborov and Rudich [RR97].

示例，使用当前技术，即与我们在计算复杂性中证明 $\mathcal{P} \neq \mathcal{NP}$ 所遇到的障碍在精神上相似的技术，来处理黎曼猜想。

10. **Play:** *Forget reality, ask for the impossible*。尽管与计算机技术的实践（和需求）紧密相连，但ToC的一些最大进步和创新却来自完全忽视现实约束和关于可能性的直观偏见，以及构建玩具模型和问题来玩耍和探索。在认真彻底地探索在引入时看似不合理甚至荒谬的观点和概念方面，这种做法在智力和实践上都得到了丰厚的回报。例如，非确定性机器、不一定总是可靠的证明、将输入放在玩家的额头上、通过电话玩扑克、在$n \times n$棋盘上玩象棋和国际象棋、定价无政府状态、无计数器计数、无知识而确信、量子后选择、完美随机性、匿名所有权、将数独和定理证明置于同等地位，以及许多需要更多文字来描述的其他内容。这些努力导致了不断为Alice和Bob、Arthur和Merlin、拜占庭将军和就餐哲学家、多臂老虎机和许多不那么吸引人的名字的玩家创造新的游戏、规则和任务。当然，许多这些游戏受到了外部、现实考量的启发，以及发明者强烈的直觉，但它们抽象和自由的探索对于理解原始应用以及常常是完全出乎意料的应用往往是至关重要的。无畏地做出假设并探索其后果，或者提出不可能的事情并调查使其成为可能的最低假设，一直是该领域的持续推动力，并将无疑继续下去。

20.4 The computational complexity lens on the sciences

我现在要讨论一个在书中几乎未涉及的话题，但有两大重要例外：$\mathcal{NP}$-完备性和量子计算。首先，我在第3.10节已经详细讨论了$\mathcal{NP}$-完备性的概念是如何以及为什么侵入所有科学，并在所有科学中变得至关重要和无处不在。让我在这里总结一下。“无处不在”可能是在描述这一罕见科学现象时的巨大低估。所有领域的科学家都发现有必要理解和使用一个非常*technical*的计算机科学概念，并且他们发表了无数[9]文章，将NP-完备性与他们正在研究的任何事物联系起来。现在转向量子计算（第11章专门讨论），到目前为止，这是一个令人兴奋、发展成熟的领域，它激发了显著的科学研究互动，并在构建量子计算机的技术开发中引发了数十亿美元的投资。这个领域是第20.3节中描述的ToC方法应用于物理学家[Ben80, Fey82, Deu85]关于构建量子力学计算机的最初建议时的完美示范。

我们现在已经远远超出了这些重要例子。

Eugene Wigner [Wig60] 的著名文章，其标题概括了其精髓，是 *The unreasonable effectiveness of mathematics in the natural sciences.*[10] 好吧，在科学中 ToC 的有效性 *is* 非常合理、预期和自然，我们将在下面给出许多示例。这种观点与大多数科学理论背后的机械论观点完全一致。它被称为 *computational lens on the sciences*，有时，当对有效资源使用的定量评估是首要任务时，被称为 *computational complexity lens on the sciences*。

我们将讨论ToC与众多科学学科的众多联系和相互作用，这些学科将算法和计算复杂性的考虑作为*modeling*和了解自然本质的重要组成部分。这种关注远远超出了计算在科学中作为*tool*的广泛使用，正如所提到的，这本身具有重大意义。我认为这些活动将在许多现有理论中带来根本性的变化，并在更好地理解许多自然现象方面取得重大进展。

[9] You can amuse yourselves with Google Scholar, to see that “numerous” may be an understatement as well. Searching when the phrase “$\mathcal{NP}$-completeness” occurs *concurrently* with *each* of “physics,” “biology,” and “chemistry” in scientific publications gets tens of thousands of hits. To make these numbers even more meaningful, these pairs *each* occur very roughly as often as natural pairs as “energy, physics,” “atom, chemistry,” “life, biology,” and “function, mathematics.”

[10] This article was debated and interpreted by many after being published.

在开始之前，让我指出，科学理论和模型中的计算元素始终存在（尽管通常只是隐含的）。这遵循了既包括哲学也包括实践考虑，这些考虑远远早于计算理论。首先，虽然哲学家和科学家们几个世纪以来一直在争论科学理论的精确性质，但所有人都同意它应该是*predictive*，即能够在实验进行之前提供（或很好地猜测）实验的结果。如果没有这种预测能力，理论通常是无用的。此外，这种预测能力对于*falsifiability*的要求是必要的：存在潜在的实验可以证明理论是错误的。但预测显然是一项计算任务！初始数据作为输入输入到模型中，必须提供预期的结果作为输出（例如，“明天太阳或月亮将在纽约什么时候升起？”）。如果这项计算任务是不可能或甚至仅仅是难以处理的，那么理论再次是无用的。解决计算问题以进行科学预测的需要是伟大科学家们努力工作的一个基本组成部分（以一个著名的例子，牛顿的*Principia*包含用于此类预测任务的巧妙高效的算法）。随着TOC和计算复杂性的到来而发生变化的是，这些计算任务的可处理性能够以数学形式化并评估。

再次，该领域的先驱图灵和冯·诺伊曼已经清楚地认识到自然过程是计算性的，并意识到这种观点对于理解自然至关重要。正如这些巨匠的典型做法，他们并未处理琐事。在生物学中最常引用的论文之一，“形态发生的化学基础”中，图灵[Tur52]提出了一个“胚胎模型”，用以解释*structured*非对称性和不均匀性（在所有生物体中占主导地位，例如斑马的色彩图案或左右手性）的出现，这个过程始于对称、均匀的初始条件，并遵循对称的进化规则。冯·诺伊曼甚至更加大胆。在他的著作*The computer and the brain* [VN58]中，他建立了计算机科学和神经科学之间第一个重要的桥梁。在这份极具远见的文件中，冯·诺伊曼比较和对比了计算机和大脑，并利用图灵的通用性、神经元输出的数字性质以及当时（年轻）两个领域的最新知识，展示了它们通过互动可以获得多少收益。在他的著作*Theory of self-reproducing automata* [VNB+66]中，他接受了终极挑战：比较机器和生物体，并模拟生命、进化和繁殖。他的*universal constructor*，一个29状态的（通用）细胞自动机，能够自我复制，本质上使用了图灵的程序和数据双重性进行复制，早于即将发现的[11]双链DNA结构，其中自然繁殖也使用了同样的双重性。对这些详细而雄心勃勃的开创性作品进行一句话的总结似乎几乎是一种罪过，我希望读者能够了解更多，特别是欣赏自然计算建模对复杂生物系统的适用性。

计算理论经过几十年才真正回归，继续跟随这些巨人的脚步，并将计算方法整合到更广泛的科学学科中。这些十年对于计算理论（ToC）的内部发展至关重要：研究了众多新的计算模型和类型，计算复杂性的焦点使我们对各种算法有效利用各种资源的理解更加深入，并能够论证这种效率的限制和权衡。这与自然建模完美契合，其中资源的有效利用是基本法则。在过去的三十年里，算法思维和计算建模在自然科学和社会科学领域的许多领域都迎来了繁荣。在许多情况下，这些领域的有远见的领导者已经认识到计算视角的重要性，并建议将其研究整合；著名例子包括赫伯特·西蒙的*bounded rationality* (参见[Sim57])在经济学和理查德·费曼的*quantum computing* (参见[Fey82, Fey86])在物理学。在许多其他情况下，无论是跟随这样的线索还是独立于它们，算法和计算复杂性的有远见的领导者，凭借这个领域的强大方法和知识，以及对理解另一领域的热情，提出了将计算视角和经典科学问题相结合的建议。

令人兴奋的是，其中一些已经转变为真正的互动，包括不断增长的协作、联合工作、会议和实验。当然，文化、语言差异，知识障碍以及简单的谨慎和保守使得这些互动的进程比其他一些互动慢。此外，与ToC相比，这些领域通常非常广泛，这通常限制了互动仅限于特定的子领域。

[11] This book was published long after von Neumann's death.

和 问题区域。然而，已经有 plenty of stories t o 告诉。

*There is no doubt in my mind that the algorithmic lens on sciences is the global scientific paradigm shift of the twenty-first century*我们只见证了其开始。所提出的计算模型将发展，新的模型将诞生，它们将具有更多互动，提出新颖的实验，并吸收计算方法。这将自然地使实验科学更加理论化，与数学建立更进一步的联系，并与自动化科学发现的互补革命更好地互动。这种发展将需要教育每一位未来的科学家了解ToC。

以下故事展示了在不同领域和问题中，计算和算法建模在众多科学学科中的某些示例。这个选择基于我的偏见和有限的知识。我主要关注ToC研究人员对这些领域的推广。这些故事应该足以让读者了解这种互动的范围及其潜力。没有理由认为所提出的任何模型或理论是“正确”或最终的。相反，重点是计算复杂性是它们的一个基本组成部分，这通常是一种新颖性，为每个领域的重大问题增添了新的视角。随着这些合作的扩展，我认为这里的故事只是填补巨大桶中几滴水的存在。然而，在我看来，这种互动在短短几十年内产生的综合影响是惊人的!

请注意，我将讨论所有许多重要的方式，其中算法、计算机系统和技术的直接使用如何相互作用并影响科学，包括分析和解释大量科学、社会和医疗数据以及模拟科学模型和过程。这种交互是另一场革命，超出了本章的范围。

20.4.1 Molecular biology

也许，生物学家和计算机科学家之间最快、最大、最重要的互动增长始于千禧之交的人类基因组计划。将新的测序技术与分析这些产物的新高效算法相结合，产生了*computational biology* (或*bioinformatics*) (，例如，参见[Pev00, JP04]关于这一合作早期工作的内容)。这些领域发展壮大，涵盖了分子生物学和遗传学的各个方面，疾病发现和药物设计，类似的合作结合了建模和算法，包括人工和机器学习的类型。这些合作，在学术界和工业界，可能比我提到的其他所有事情加起来还要庞大!

在一个反向转折中，计算机科学家[Adl94,Lip95]启动并研究了*DNA computing*，即使用像DNA这样的大分子来更快地解决复杂的计算问题。尽管与电子过程相比，分子过程的速度较慢，但它似乎以极低的成本提供了巨大的（尽管是恒定的）并行性。这确实在Len Adleman的开拓性论文[Adl94]中得到了实验室的证明。请注意，这个计算模型的通用性并不明显，但[BGBD+04]通过体内实验证明了其通用性，并具有在细胞水平上进行诊断和治疗的潜在医学应用。

除了在通常意义上计算函数之外，Ned Seeman [See04] 还设想了从初始DNA和其他分子设计并物理构建纳米材料。图23展示了我们编程DNA类似分子以在特定位置具有某些原子，这些原子将以乐高式与其他分子相互锁合，从而产生许多不同的结构。这些结构与自然界自身所做的一切都完全不同（例如，图23中由精心设计的DNA链制成的立方体）。有关更多算法技术和结果结构，请参阅[RPW04, BRW05]。能够为各种功能和结构医疗目的编程这种纳米级有机设计具有极大的前景（例如，参见[KSZ+10]），尤其是当人们不仅能构建刚性结构，还能构建移动机器（例如，参见[GV08]）。对于其他材料（例如，基于碳的；参见[TPC15]第11章）的纳米自组装也是如此。确实，生物学家和计算机科学家在“可编程物质”的模型和算法方面的联合活动越来越多（例如，参见最近的例子，工作坊报告[DK16]和论文[Der17]）。在所有技术中，容错性是一个关键问题，容错自组装是这个努力的主要动力——参见综述[Win06]。

它很清楚，这些令人兴奋的能力需要一种关于底层“编程语言”的通用理论。

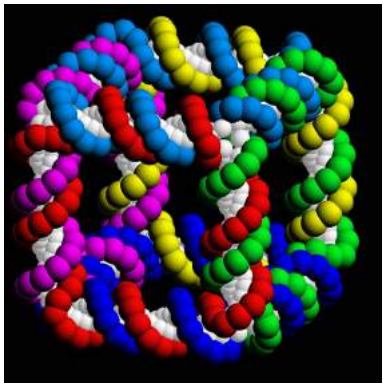


图23. DNA链构成的立方体。经Nadrian C. Seeman许可使用。

其基本构建块是纳米材料，其基本操作使用化学或物理键。研究此类语言（以及最终的设计）的表达能力（即，哪些架构是可能的以及成本如何）无疑是正在路上，正如上述例子所示，与技术开发并行。这些模型为ToC研究人员创造了新的挑战。一个想到的类比是*origami*，折纸艺术。几个世纪以来，聪明的艺术家通过简单地折叠一块不同颜色的纸张，找到了越来越复杂的图案设计。然后出现了一群计算几何学家，他们在[DDMO04]中证明了*every*合法设计是可折叠的，并且还给出了一种有效的算法来实现这一点！Demaine和O'Rourke的奇妙书籍[DO08]描述了对“物理”算法的这种完整理解，其中一些与上述生物和化学问题非常相关。

20.4.2 Ecology and evolution

自然模型的计算复杂性至关重要这一事实，查尔斯·达尔文在他的*Origins*中就已经很清楚，这是科学史上最有趣的争议之一的核心！达尔文不遗余力地估算地球的年龄，以便检查他的进化论是否与地球生命达到如此多样性的时间相一致。在他那个时代，（宗教）教条仍然是地球有6,000年历史，对达尔文来说，这远远不足以让突变、繁殖和自然选择从共同起源达到那样的多样性。幸运的是，地质科学正是在那时得到发展，达尔文利用它来估计地球的年龄为几亿年，这对他来说似乎是令人满意的.[12]但当他得知伟大的物理学家威廉·汤姆森（后来成为开尔文勋爵）根据当时最好的物理理论，仅估计太阳的年龄为几千万年（大约是达尔文的10倍*lower*）时，他非常沮丧（正如他在给华莱士的信中所承认的）。这个时间估计似乎有可能使达尔文的理论失效。当然，正如我们所知，汤姆森非常错误（因为核力是在很久以后发现的），地球的年龄大约是达尔文估计的10倍。但这个故事的信息完全清楚。任何自然过程都消耗资源（如故事中提到的*time*），因此“计算复杂性”必须是相关理论一致性和可证伪性的一个组成部分。事实上，对进化过程及其复杂性的更好的计算建模可能导致对这一著名理论的修订和更好的理解。

构建一个定量、计算理论，该理论将解释通过变异和选择从简单机制进化而来的复杂机制（例如，在细胞中），这是Les Valiant在过去十年中的追求，始于他的论文“Evolvability”[Val09]。他认为进化是他PAC *learning*方法的限制形式，我们在第17章中对其进行了考察。而不是深入细节，让我向您推荐Valiant令人兴奋且极具可读性的书籍*Probably, Approximately Correct* [Val13]。

[12]Needless to say, this time estimate and its sufficiency for Darwin's theory relies on many assumptions, which he gave in detail.

他在其中阐述了他的理论，特别是关于 *ecorithms* 的概念：与环境交互的算法。

从这个非常一般性的进化问题和模型，我们转向讨论两个具体难题，我相信其解决将涉及计算复杂性。在这方面已经提出了一些初步建议。

著名的“性别问题”（由格雷厄姆·贝尔在进化生物学中称为“问题女王”）要求解释在自然界中，尽管有能量成本和遗传物质（可能通过选择而改善）的损失，性繁殖为何如此普遍。毕竟，无性繁殖在能量上似乎要便宜得多，并且可以持续提高适应性。当然，这些简单的考虑只是漫长历史辩论的开始，以及许多远未解决的众多理论。在这里，我只想再次指出，不深入细节，最近由进化生物学家和复杂性理论家组成的联合团队[LPDF08]提出的定量、计算和非常不同的提议，在以下（和其他）论文中得到了进一步探索和扩展：[LPPF10, CLPV14, MP15, LP16]。

进化生物学的另一个谜团是“冲突问题”。大致来说，它询问的是，经过数百万年进化经验优化的生物，如何因为无法解决不同选择之间的冲突而“停滞”。这种现象在许多物种和情境中都有观察到（人类对此非常熟悉），并且与上述问题一样，引发了大量的辩论和不同理论的提出。另一项协作工作[LP06]提出了一种复杂的计算模型，其中系统向最优行为的进化可能会在其内部自然地创造出两个可能相互冲突的子系统！计算限制在这个模型中起着至关重要的作用。

本节总结，让我从非常不同的角度讨论理解自然界中一些非常不同的算法的非常一般的问题。

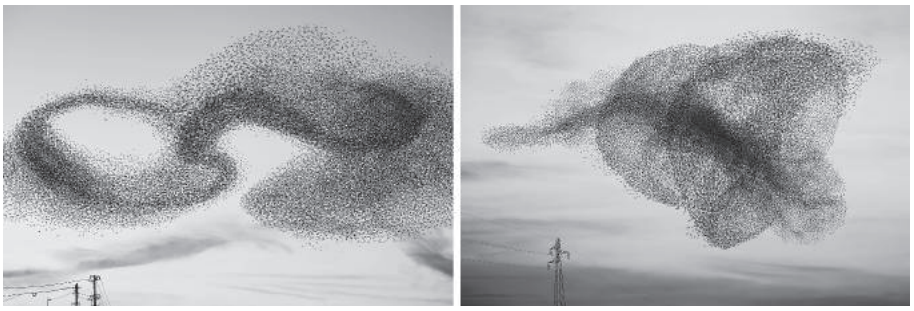


图24. 星星群聚（图片由Alain Delorme提供）。

图24中的静态图片甚至无法开始展现你通过互联网搜索“星鸦聚群”所能找到的惊人视频的影响力。观看成千上万鸟类的视频几乎是不可能的，人们在难以置信和敬畏之后，会寻求理解这种分布式系统产生如此壮观、复杂但协调一致的行为的机制。确实，几十年来，科学家和数学家们一直在研究这些（以及许多其他物种的众多其他形式的非凡协调和行动）。我指出了一种非常一般的离散模型，称为*influence systems*，由Bernard Chazelle提出（参见他的调查[Cha12,Cha15]）。它试图捕捉一个分布式网络的动态，其中动态本身会改变网络拓扑。这个模型不仅与自然系统和算法相关，而且与许多动态社会情况相关。Chazelle的工作发展了研究此类非线性模型的技术，并给出了鸟类集群和类似系统中收敛时间的第一个定量上下限。

20.4.3 Neuroscience

与本章讨论的许多其他科学问题不同，理解大脑（尤其是人类大脑！）的探索始终具有显著的计算成分，因为大脑通常被认为是一个（尽管极其复杂）的计算设备。大脑计算建模的早期工作包括 McCulloch 和 Pitts 的关于 *Nerve nets* (的工作 [MP43]，这是一位神经科学家和一位逻辑学家的合作)，以及 von Neumann 关于该主题的上述书籍 [VN58]。生物技术和计算能力的巨大进步导致了神经元、神经元之间的连接以及神经元在各种行为中的活动理解的不断提高。反过来，这导致了大量（神经网络）大脑模型和大脑部分模型的出现，试图以不同层次的具体性解释各种认知和其他功能。有趣的是，许多此类模型的本性和分析中的挑战吸引了物理学家和生物学家的兴趣和合作，这已经持续了几十年。所有这些都代表了浩瀚且仍在迅速增长的工作。在众多关于该主题的文献中，Dayan 和 Abbott 的 *Theoretical neuroscience* [DA01] 是一本出色的全面书籍，概述了所有层次的大脑描述性、机制性和解释性计算模型。

在以下内容中，我描述了一些最近的研究，这些研究强调了将计算复杂性整合到大脑模型中的重要性。我预计这些研究将不断增长，并导致更多的合作与理解.[13]

关于大脑的一般计算模型，Les Valiant [Val00] 从计算复杂性的角度开创了全面的设计。它对神经元之间的互连模式和单个神经元的计算能力提出了精确的数学假设，这些假设与已知的生理结果一致。然后，它继续描述了可以在该模型中实现的详细算法，并执行一些基本的认知功能，如记忆和学习。这些细节包括信息和（集合中的）神经元如何以及存储什么，以及这些算法消耗的物理和计算资源的仔细核算。对我来说，与其说是一个实际的大脑模型（这可能需要几十年才能弄清楚），不如说是一个建模大脑的模型。特别是，它建议进行进一步实验，并且具有极高的可证伪性。奇怪的是，它甚至预测了某些生理现象（例如，强突触，神经发生）的需求，这些现象是某些算法所必需的，而这些预测在书首次出版后确实在一定程度上得到了验证。Valiant在[Val06]中进一步阐述了大脑一般建模的原则。

其他作品试图在计算上解决大脑的更具体区域，并专注于更具体认知功能的实现。例如，海马体和记忆在[Val12, MWW16]中进一步研究，皮层和学习在[Val14, PV15]中，以及神经组织中信息的压缩在[AZGMS14]中。虽然这些论文大多由计算机科学家撰写，但它们出现在神经科学家阅读（并且可能由他们审阅）的期刊上，并承诺未来的合作。

神经生物学中最大的计算和算法挑战之一源于*connectomics*，这个领域（在某些类比于人类基因组计划，难度大得多）试图绘制出整个大脑，包括所有突触连接。即使部分成功，也希望能够模拟其在计算机上的活动，并更好地理解大脑的过程和疾病。这种跨学科的*blue brain project* [Mar06]可能代表了今天最大的此类协作尝试。即使在决定收集哪些数据、保留哪些数据、如何组织它以及如何从中提取所需结构等方面，也出现了“大数据”挑战，这些挑战很大，并且远未达成共识（当然，这些挑战在许多其他领域也存在，并且越来越多，在这些领域中可以轻松收集到前所未有的大量数据）。神经科学家和计算机科学家的联合工作[LPS14]试图阐述指导这项任务的原则，并可能构建连接组学的计算理论。

20.4.4 Quantum physics

与第20.4.3节不同，在那里我关注了复杂理论学家对神经科学的贡献，这里我关注的是物理学家，他们使用复杂理论的结果和方法来提出解决

[13] See, for example, `https://simons.berkeley.edu/programs/brain2018`.

物理难题的解法。*As we shall see, the physicists' quest for understanding the fundamental structure of physical space and time may need the fundamental understanding of the computational resources of space and time!*

我在这本书的第11章中全部致力于量子计算，这一领域由物理学家发起，并由计算机科学家继续发展。这一领域导致了两个群体之间非常紧密和富有成效的合作，并在量子力学和量子信息理论中产生了新的结果和基本问题。这种互动的一个副产品是物理学家对复杂性理论概念和结果的整合，这允许我提出以下方案。我将专注于量子引力，即长久寻求的理论，它将使量子力学和广义相对论相结合，朝着爱因斯坦关于一个*unified theory*的梦想迈进。我只会讨论几个（相关）例子。我完全预期在物理学的所有领域内，计算考虑与理论之间的整合将更加深入，并且不同领域之间的互动将更加频繁。

它全部关于黑洞。如果你觉得以下内容过于神秘和牵强，要知道我和一些物理学家有同样的感受。我们讨论的是在可预见的未来难以检验的物理理论，而关于它们的论点很大程度上依赖于思想实验。此外，许多这些想法都是最近的，并且被广泛讨论。我会放松并非正式。对于渴望细节的勇敢者，Harlow [Har16] 写了一篇技术但非常易读的概述，其中包含大量直觉，详细介绍了以下讨论的大部分方面。它还包含了所有相关参考文献。一篇更非正式的概述，这些问题的解决方案（具有更多计算复杂性背景）是Aaronson [Aar16a] 的讲义。现在，准备好听两个充满术语的故事，你应该像对待幻想一样接受并继续阅读，直到（计算）笑话的结尾。

引力与量子力学的基本联系是 *Hawking radiation* (，虽然霍金是第一个将其完全数学化的，但他使用了泽尔多维奇和贝肯斯坦的早期想法)。这些物理学家发现，尽管黑洞如此之重，以至于吞噬周围的一切，包括光（这就是我们无法直接看到它们的原因），并且似乎注定要永远增长，但量子效应仍然导致它们泄漏粒子，尽管这种效应的速度极其缓慢，但黑洞最终会完全蒸发！

霍金辐射，这是一个被广泛接受的现象，使得将广义相对论（例如，黑洞中心的奇点，黑洞事件视界的时空平滑结构）对它的看似冲突的约束与量子力学（例如，辐射的幺正演化，互补性，量子态的非克隆性和纠缠的单向性）对它的约束结合起来变得极其困难。这些约束之间的潜在冲突（并非所有这些约束都享有相同的接受度）通过巧妙的思想实验得到了探索。其中核心的一个，被称为*the firewall paradox*，由Almheiri等人[AMPS13]提出。简单来说，这个悖论可以大致总结为：一个坐在事件视界之外的观察者能够从泄漏的霍金辐射中提取一个量子比特的信息，然后跳入黑洞，在越过视界后提取那个*same*量子比特，这是一个矛盾（这将违反上述提到的量子信息理论的基本定理*monogamy of entanglement*！）。

许多解决悖论的建议被提出，但Harlow和Hayden [HH13]（参见[Aar16a]中的详细说明和简化，第6讲）的独特之处在于引入了计算观点，这对于许多关于此类过程的其他问题都相关。它本质上将产生辐射的黑洞以及从该辐射中提取悖论性比特的观察者都视为有效的（多项式时间）量子算法。这种观点导致了一个非平凡的计算机分析，其结果是如果*quantum statistical zero knowledge is hard for quantum algorithms*,[16]，那么*the black hole will evaporate long before the observer can compute any paradoxical bit*。简而言之，计算复杂性为这个悖论提供了一种解决方案：时间太短，以至于

[14]The "boundary" of the black hole; anything inside it is trapped and falls into the black hole.

[15]I did not make this up!

[16]No need to understand what that means, other than that it is a computational assumption about the intractability of a natural problem, like $\mathcal{P} \neq \mathcal{NP}$. But I stress that this strange computational assumption, merging cryptography and quantum computing, has made it to the consciousness of physicists.

实验将进行，仅仅因为观察者的任务是计算上困难的。[17] 这种解释（本身也是条件性的）仍然与其他解释一起被讨论，但它的计算精神已经进入讨论，并且随着它，将自然和观察者视为算法的理解变得清晰！

理解黑洞内部的进程，这些进程受广义相对论支配，当然在定义上就有问题——无法窥视其内部。从其边界观察到的现象中推导它不是很好奇吗？Maldacena [Mal99] 提出了一个引人注目的建议，有点类似这种想法：一种通用的方法，通过研究量子力学系统在其边界上的动力学，来获取关于量子引力理论（如弦理论）在时空区域“主体”（或内部）中粒子行为的有关信息。这种理论已成为 *AdS/CFT correspondence*，其中“AdS”代表反德西特空间，[18]，“CFT”代表共形场论。共形场论已被完全接受，并且它们的演化参数的计算在某些情况下已成为标准。使用这些来推断AdS宇宙（或实际上，反过来，使用AdS中的广义相对论来绕过复杂的CFT计算），需要一本将AdS与CFT量相关联的词典。确实，Maldacena提供了一份广泛的此类词典。虽然还没有证明这种对偶性，但它已成为理论物理许多领域的极其有用的工具，并允许对主体行为做出可能被检验的预测。

现在，让我们关注*the Susskind puzzle*，它要求一个称为*wormhole length*的AdS量的类比。这种“虫洞”的想法可以追溯到爱因斯坦和罗森，根据Maldacena和Susskind的另一个令人兴奋的新理论，虫洞应该用来解释纠缠的本质。无论如何，我们只需要知道，在适当的AdS宇宙中，虫洞的“长度”随时间线性增长。Susskind的难题是在字典中填充对应CFT的类比量。任何物理学家都很难想象在量子演化中与时间成线性关系的*any*量！但是Susskind通过考虑计算复杂性找到了答案。在给定的CFT中离散时间和空间，你得到一个在某些n量子比特上的量子电路，该电路在每一步迭代（例如，使用某个固定的初始状态ψ_0，例如所有零）时进行迭代。经过t步后，这种迭代产生了一系列量子状态，ψ_t。Susskind提出，在CFT中的ψ_t量子电路复杂性是AdS演化t步后虫洞长度在字典中的类比。换句话说，量子引力理论的一个基本、*physical*组成部分根据这个提议对应于CFT中的一个*computational*参数。再次强调，尽管这个观点仍然新鲜且备受争议，但将这些理论中的过程视为量子电路，并研究它们的计算复杂性，正在进入理论物理学的讨论。

20.4.5 Economics

一个与“远程”学科互动的ToC巨大成功是经济学，尤其是博弈论。这一领域诞生于20世纪40年代末，约翰·冯·诺伊曼和奥斯卡·莫根施特恩的著名巨著*Theory of games and economic behavior*，以及约翰·纳什关于均衡的基础论文。在接下来的几十年里，博弈论发展了复杂而精细的市场和理性代理人在其中的战略行为理论。这些理论显然旨在具有预测性，并为战略实体的决策提供信息（从冯·诺伊曼最初想要玩好（令人恐惧的）冷战游戏的动机，包括核攻击等可能策略，到不同公司之间（可能恶劣的）市场份额竞争，其中策略可能仅仅是产品价格，一直到众多情况中的个人和社会互动）。在该领域的许多理论和论文中，直观上很明显，真实的人或组织，与

[17] In the same way as in the section about Lord Kelvin's calculations of the age of the universe could have killed Darwin's evolution theory, if they had been correct.

[18] To stress that the theory of gravity we employ happens on a negatively curved space, namely, has a negative cosmological constant. As an aside, notice that this certainly is *not* the case in our universe, whose expansion is accelerating, and so has a positive cosmological constant; this does not stop physicists pursuing first the full understanding of this "toy model."

[19] Namely, a quantum field theory equipped with a strong symmetry called "conformal invariance." Importantly, there is no gravity in CFT!

[20] In general, these require quantum computation.

[21] Namely, the size of the smallest quantum circuit generating that state from the zero state.

计算限制，实际上必须实施所建议的策略。确实，已经提出了某些算法，但相对而言，对它们的性能进行正式调查或更一般地将计算复杂性纳入这些模型的工作做得很少，部分原因是缺少相关的计算理论。

这个空缺始于20世纪80年代，当时算法和复杂性理论相当发达，将复杂性理论应用于经济学的兴趣增长。自20世纪90年代互联网出现以来，这项工作得到了加速，创造了新的市场，其中参与者的处理时间成为主要关注点。与复杂性理论的互动以及将计算和算法元素整合到经济理论和模型中涵盖了该领域的许多方面。对我来说很幸运，书中出现了一本优秀的调查集 *Algorithmic game theory* [NRTV07]，因此我将在下面简要回顾其中的一些主要主题（现在，是全面的研究领域）；更多细节和缺失的参考文献可以在该书的章节中找到。自本书出版以来，还有更多研究和合作发生，我将用一个（非常不同）的例子来结束，这个例子展示了这些领域之间的互动。

- *Complexity of equilibria*平衡在多种战略情况（游戏、市场）中体现了理性解决方案的概念：它们保证在考虑其他方的行为后，每个个体对其收益的满意度，因此被视为稳定性的概念。博弈论的核心定理证明了在非常一般的情况下存在平衡。但如何实现它们很少被讨论。Papadimitriou [Pap94] 的开创性工作观察到，这种证明中使用的论点是纯粹存在性的（例如，拓扑不动点定理），而计算平衡的自然算法需要指数时间。他为经济学和这类数学问题发展了一种复杂性理论，即解决方案由各种存在性原则保证。他提出了计算纳什平衡可能是PPAD类（即与找到Brouwer不动点一样困难）的完备问题的可能性。这15年后得到了肯定 [DGP09, CDT09]，甚至对于两人游戏，这是对纳什平衡困难性的第一个迹象，可能限制了其作为经济解决方案概念的价值.[22] 证明这个结果需要将*graphical games*族引入到[KLS01]中的博弈论。对有效平衡概念的“需要”导致了纳什平衡的近似版本，这些版本确实有近多项式时间算法 [LMM03]。此外，困难性结果和算法随后扩展到各种其他平衡，最著名的是Arrow-Debreu市场平衡。这类问题的近似算法将这个领域与优化联系起来，导致了许多“现实生活”情况（限制效用、偏好等）中，平衡*can*可以有效地计算或近似。

- *Price of anarchy*这是对均衡的全新理解，本可以成为经济理论的一部分，但不知为何并未如此。由Koutsoupias和Papadimitriou在开创性论文[KP99]中提出，*price of anarchy*是“分布式”解决方案对社会成本之比，由个体（自私且不合作的）代理策略实现的次优均衡给出，与由（仁慈且完全信息）实体分配给它们的策略实现的最佳“集中化”成本之比。许多需要代理在利用共享资源（例如，道路上的汽车、互联网上的数据包）中做出决策的情况会影响性能（从而影响个人和社会成本），并引发了一个问题：这种选择的自由度（“理性无政府状态”）的成本是多少。这篇论文导致了非常令人惊讶的结果，表明在许多一般情况下，无政府状态的价格远低于预期。这类主要一般结果之一是[RT02]，它确立了在任意大小的*any*网络上的流量流量比最多为4/3，其中链路上的延迟是其拥塞的线性函数。这个界限惊人地与经济学中几十年来已知的紧约束示例Braess悖论相匹配（表明向网络添加链路实际上会增加拥塞），这是在4节点网络上实现的。另一个在一般性和提供的无政府状态价格界限上同样令人惊讶的例子是

[22] Further hardness results based on various (so far, nonstandard) *cryptographic* assumptions have recently been pursued (see, e.g., [BPR15]). These results heavily rely on the reductions and completeness foundations of [Pap94] and their subsequent development.

发现了与 Walrasian 平衡完全不同的设置 [BLNPL14]。我们强调，虽然分布式与集中式解决方案成本的问题本质上是非算法性的，但它们以及它们的解决方案都自然地来自 ToC 方法论（这里主要是指使用约简和近似保证，这在经典博弈论中并不常见）。本书展示了 ToC 方法论在数学中其他非算法性问题与答案中的强大作用。这些重要工作自然也导致了许多算法性问题，再次与优化、在线算法和分布式计算相联系。这些工作与博弈论中的研究问题相联系，并在不同领域之间产生了新的合作。

- *Mechanism design* (有时被称为“逆博弈论”)。博弈论的主要目标是理解参与游戏的理性、策略代理的行为，这些游戏已经*pre- specified*。机制设计是博弈论的一个领域，它提出了一个元问题：如何指定（或设计）这些游戏、激励和行为规则，以便迫使类似的理性、策略代理实现某些全球目标。例子比比皆是：政府制定法律和分配资源以实现各种社会利益，拍卖旨在销售物品并最大化利润，投票系统用于最好地代表公众意见，以及平衡服务和成本的医疗保健和其他社会系统。在这个研究领域中，发现了许多基本原理和定理，例如解释了如何诱导代理的行为具有各种属性，如真实性、最大化利润或某些全球社会结果。再次，互联网的出现为机制设计增加了许多新的应用，这些应用也比许多先前考虑的更为复杂，因为参与者的潜在巨大数量、行动速度和互动的分布式性质（在线广告市场是一个完美的例子）。现在算法是代理，必须考虑它们的优缺点。这极大地丰富了机制设计理论。例如，使用通信复杂性和其他工具允许得到限制机制能力的困难结果。另一方面，将（分布式）算法和协议视为由理性代理执行的观点为算法理论增添了全新的成分；其设计应确保在自私理性玩家实施时，行为是正确的、高效的、公平的等等。在这个领域，ToC与博弈论之间的合作可能是最广泛的，这在工业界也得到了体现，即使仅关注由数十亿美元搜索引擎广告行业驱动的广告拍卖的理论和实践，也可以观察到这一点。

- *Cryptography and game theory*. 我在上面提到了由于计算复杂性的限制，在发现和实施所需策略（例如，在均衡中）后对经济学的负面影响。然而，从密码学中产生了大量积极新闻，关于实施在（通常假设的）博弈论信息论环境中看似不可能或难以实现的战略。作为一个例子，考虑Aumann的*correlated equilibrium*概念，与纳什均衡相比，它易于计算，*if*所有关于所有玩家效用的信息都为某个中心可信玩家所知。这种“不切实际”的假设使得Aumann的概念在不存在此类可信权威或玩家希望保护其隐私的情况下几乎无用，直到你意识到第18章中我详细阐述的*secure multi-party computation*恰好解决了这个问题。确实，论文[Yao86, GMW87, BOGW88]及其追随者（在不同的设置中）这样做，消除了在任何此类情况下需要可信方的需求，并使Aumann的解决方案概念（以及许多其他概念）变得极其现实和有用！这两个领域的公理的许多差异（在密码学中，隐私是一个目标，而在博弈论中，它通常是手段；在密码学中，玩家可能是诚实的或恶意的，但在博弈论中，它们被假设为理性的等）保证了当我们把一个领域的观点整合到另一个领域时，会产生巨大的协同效应和合作。

让我用一个不同领域之间互动的例子来结束，这个例子解决了一个完全不同、根本的问题，该问题检验了市场透明度和效率的假设。关于2008年金融危机的文章如山般涌现，提出了许多关于那次巨大崩溃的原因和模型。普遍认为，金融衍生品，如CDOs的定价错误

(抵押债务证券)发挥了核心作用，并且关于如何应对它的争论。一篇提出从计算复杂性角度看待这种情况的论文是协作[ABBG11]，“金融产品中的计算复杂性和信息不对称。” 在其中，作者提出一个简单的模型，展示了买方和卖方之间的信息不对称，以及因此他们分配给相同衍生品的成本可以极高，即使在完全透明的市场中，即使买方（或任何衍生品合同的检查员）在计算能力上极其强大（例如，大型银行、监管机构或政府）。任何阅读过第18章的人都可以轻松创建这样的*artificial*例子，实际上，正如其中解释的那样，这种信息不对称是所有互联网安全和电子商务系统的根本！论文[ABBG11]构建了简单的、*natural* CDOs，由混合“正常”和“垃圾”资产组成，并表明在随机混合和偏向“垃圾”混合之间区分是复杂度研究中的计算难题。这篇论文已经引起了一些关注，并被经济学文献引用（例如，参见非常不同的[CKL13, BCHP17]）。[23]我认为许多金融模型（例如，这里引用的论文中的市场和风险模型）在计算复杂性理论的背景下可能需要修订和扩展。与博弈论不同，到目前为止，经济学中的金融方面与ToC的合作非常有限，但我预计这些合作将会增加。

20.4.6 Social science

互联网的出现以及它所支持的各种社会活动和网络，为社会科学家带来了许多新问题，同时也首次使他们能够在大量参与者中进行社会实验。这创造了计算机科学家和社会科学家之间的大量新合作，丰富了社会科学的许多方面，并使其更加量化。除了下面提到的（早期）著名例子，该例子说明了ToC的影响性质，我将直接引用优秀的、广泛的书籍*Networks, crowds, and markets: Reasoning about a highly connected world* [EK10]。这本书本身就是一种合作，我再次指出，自其出版以来已有8年多的时间发生了许多事情。书中描述的ToC与社会科学之间互动的主题种类繁多（尤其是在互动开始以来的短时间内），我只是略作介绍。这些主题突出了ToC在分布式网络研究中的重大演变。这个广阔的领域的大部分研究集中在人为设计的网络及其算法上。与社会科学的互动极大地扩展了这一领域！在书籍[EK10]所描述的许多新方面中，包括不同（物理、信息、社会）网络的增长和变化，其中权力结构（例如，中心节点、权威机构、机构）的有机出现和演变，以及动态过程（八卦、影响、激励、连通性等）对网络的影响。许多网络模型在被视为市场时，其参与者被放置在节点上的策略性代理，包括博弈论和经济方面。这使这里的研究与第20.4.5节的研究相联系。总的来说，获取许多现实世界数据的便利性有助于测试新的模型、理论和算法。当然，书籍[EK10]没有涵盖许多其他社会互动的方面和模型，例如上面提到的Chazelle关于*influence systems* [Cha12, Cha15]的工作。

我们的例子是Jon Kleinberg的论文[Kle00]，题为“小世界现象：算法视角”。我们在这里仅作简要概述；鼓励读者阅读其清晰、详细的阐述和参考文献。这篇论文是对Stanley Milgram关于“小世界问题”的著名实验的解读，首次在[Mil67]中描述。在追随之前追求流行的陈词“六度分隔”的工作之后，Milgram设计了一个实验，通过让人们仅使用彼此相识的中间人将信件从一个人转发给另一个人，来测试美国人与人之间的连通性和距离。所需的链条通常确实非常短，并提出并分析了多个分析网络模型，这些模型可能同时解释社会关系并支持这样的短距离。但在Kleinberg指出存在另一个需要解释的基本问题之前，已经过去了30多年：*Even if short paths exist, why were such paths found by a purely local algorithm*? Kleinberg继续给出了一个彻底的答案。

[23] Be aware that the word “complexity” can mean many different things in economics.

首先，他证明了许多网络模型（包括过去工作中提出的多数模型）具有大量短路径，*no*局部算法将找到这些路径！接下来，他提出了扩展的自然模型，并确定了这一类中唯一一个局部算法能够成功实现的模型。这项单一工作对于上述合作以及众多后续工作至关重要。

20.5 Conceptual contributions; or, algorithms and philosophy

让我在讨论ToC影响本质时再提升一个抽象层次。我们已经在书中看到了一些以下重要贡献，但还有很多我们没有讨论！

艾伦·图灵不仅为计算机科学的数学基础和随后的计算机革命奠定了基础；他还展示了计算在揭示基本概念上的强大光芒。他的文章《计算机与智能》[Tur50]探讨了最困难且最具争议性的观念之一：*intelligence*。他以非凡的清晰度和简洁性，这在关于如此哲学、社会和科学性强烈的观念的论辩中是罕见的，[24]他提出了一种完全新颖且原创的方法来定义和理解它。

多年来，ToC 面临着理解、从而定义许多占据知识分子数百年的学科概念和观念的需求。这种需求有时源于科学或技术根源，有时确实源于该领域思想家的哲学倾向。在计算重点和语言核心的指导下，他们为这些概念产生了新颖的定义，为它们的含义和效用争论注入了新思想。这并不意味着计算视角比其他观点更好——它有时是其他观点的补充。相反，它指出了该领域的知识深度，以及计算（甚至更不用说计算复杂性）在哲学层面的相关性。我为自己属于一个认真对待定义（有时是重新定义，有时以几种方式）和理解这些基本观念的领域而感到自豪，这些观念包括：
collusion, coordination, conflict, entropy, equilibrium, evolution, fairness, game, induction, intelligence, interaction, knowledge, language, learning, ontology, prediction, privacy, process, proof, secret, simultaneity, strategy, synchrony, randomness, and verification.

它值得再次慢慢阅读这个列表。我发现将这个列表中的概念所代表的长远历史、文本数量和知识广度与ToC的相对较小规模和年轻性进行对比，相当引人注目，因为ToC为他们提供了很多理解。此外，对于许多这些概念，几乎没有或没有期望计算机科学需要处理它们，如果需要，将找到有意义的、启发性的方法来处理。只有回顾过去，这一切现在看起来才如此自然。本书讨论了这些概念中的一些，当我这样做时，我试图阐明计算复杂性如何阐明它们，通常直接借鉴那些最初提出这些定义的非常善于表达的人。Scott Aaronson [Aar13b] 指出，尽管我们中的一些人非常善于向自己的社区传达我们工作的哲学意义，但我们可能应该更多地将其传播到外界，特别是向哲学家们。他的论文正是试图做到这一点——解释计算复杂性在理解一些哲学和科学难题中可能发挥的重要作用。

这个整个方向，抽象了计算和复杂性在知识论中的作用，可能值得再写一本书，这本书比这本更少技术性，更易于理解。在这里，我将满足于以两个非常高级的哲学问题结束，可能是形而上学原则，这些问题我在书中讨论得不多。第一个是 *subjectivity*，第二个是 *interaction*。再次强调，这两个都不是新的，但借助计算复杂性的视角，它们为上述许多概念提供了信息和修正。

Subjectivity 第一个是计算复杂性在现实观中建立 *subjectivity* 的作用。几个世纪以来，属性（包括上述列出的许多概念）被定义并被视为对象、代理或交互的内在、*objective* 属性。现在它们被计算处理。

[24] But typical of all of Turing's arguments on many other issues, including his definition of a computer!

复杂性理论作为*subjective*，高度依赖于观察者！这让人联想到爱因斯坦的狭义相对论及其（在当时）激进的预测，即不同的观察者可能看到同一物体具有不同的长度，或者可能对两个事件是否同时发生持有不同意见。爱因斯坦通过添加一个公理得出这样的预测：*the speed of light is constant*。计算复杂性所采用的公理是*certain problems are intractable*。这同样具有激进的影响。一个随机事件可能对一个观察者来说是完全不可预测的，而对另一个观察者来说是可预测的。一条信息可能对一个接收者来说是清晰的，而对所有其他人来说则是完全不可理解的。这些简单的计算公理的后果是互联网安全和电子商务，以及许多其他应用的基石。更重要的是，不同观察者（他们实际上可能是某种活动的参与者）的计算能力差异使我们能够从他们的观点出发，量化“有多少”给定属性存在。例如，我们可以精确量化，给定一个观察者的计算复杂性，某些事件的可预测性，或者观察者可以从中获取多少关于秘密信息的知识，或者观察者可以多么准确地评估复杂的经济商品。简而言之，不同的计算能力意味着具有不同的推理、分析和推理能力。对这些限制的巧妙运用，以及我们计算公理的适当版本，使这种对现实的主体性观点成为计算系统最激动人心和重要的应用之一。

Interaction 交互增强了生活中的许多方面。通过比较允许学生提问的班级和不允许提问的班级，可以轻易地看出交互对*understanding*的重要性。这种性质的交互是图灵的“模仿游戏”的基础，这也支撑了他的*intelligence*方法。简单的例子也表明，自然计算任务中两个当事人之间的*cost* (，例如通信量)，在双向交互与单向交互相比可以小得多。计算复杂性中交互的研究揭示了极其惊人的能力，这在本书的一些章节中进行了讨论。以下是一些例子。一个主要影响是在*proof*的核心概念上。在经典上，证明是从证明者到验证者的单向通信。允许双向*interactive proofs*[25]导致了证明能力的全面修订。首先，可以交互地证明更多：$\mathcal{IP} = \mathcal{PSPACE}$定理[Sha92]基本上展示了证明者如何让验证者相信他在棋类游戏中拥有获胜策略，这在经典证明中是无法想象的。其次，证明可以具有悖论性质：$\mathcal{NP}$零知识定理[GMW91]基本上表明，每个有证明的陈述都有一个同样令人信服的交互证明，但不会向验证者透露任何新信息。其他基本发现将单向结果扩展到交互设置。香农著名的成果[Sha48]证明，单向通信可以通过仅使用常量冗余（通过他的纠错码）来保护免受常量噪声率的干扰。同样，通过Schulman的新颖码[Sch92, Sch93]，这个结果也适用于双向通信，尽管这种对话具有适应性。

让我以一个更高层次的贡献来结束这一节，在这个令人惊讶的信息隐私背景下，对交互的研究实际上确实影响了*scientific method*本身。这种方法的一个关键公理要求科学家决定关于她收集的数据*before*要提出的问题。这迫使与自然的单向沟通：首先接收或收集信息，然后才进行处理。当然，发现会导致新的问题，并需要新的实验和数据收集。由于效率原因，科学家可能会（实际上，有些人确实会）倾向于使用现有数据来回答新问题，而不是收集更多数据。这种数据自适应使用的成果是否有效？碰巧，*differential privacy*的交互式定义以及维护它的有效算法，令人惊讶地揭示了自适应数据分析[DFH+15]的脆弱有效性，表明在某些情况下，上述诱惑可能是合理的，以及如何！这种深入的联系，远远超出了开发出的模型和算法的预期效用，进一步展示了计算概念的深度。

[25] And incorporating randomness, error, and complexity.

20.6 Algorithms and technology

我们现在转换话题，讨论算法设计与计算系统产业之间激烈且持续的互动的几个方面。

考虑到其对计算复杂性的关注，本书对*the*在ToC中的主要努力投入很少，即算法设计。ToC的主要产品是为特定问题设计并分析有效算法，更根本的是，在各种模型中为广泛类别的计算任务设计算法技术和范式。在此，我讨论了与算法、它们的设计和影响以及与技术的相互作用相关的一些问题。一句忠告：在随后的许多讨论中，算法与技术之间的边界并不总是完全清晰（正如理论与实践、科学与工程、纯数学与应用数学之间的情形）。

20.6.1 Algorithmic heroes

算法和技术的结合力量已经提供了无数应用，仅仅在几十年内就彻底改变了人类社会。毫无疑问，这一趋势将继续并加速。尽管精巧的算法和显著的技术进步在这一革命中发挥了核心作用，但在我看来，公众并没有完全意识到算法在使他们体验到的应用中扮演的至关重要的角色，反而几乎将变化归因于技术进步。当然，在许多情况下，算法和技术进步是交织在一起的。但遗憾的是，即使在
a single algorithm has enabled a whole industry 的情况下，甚至在算法的本质可以相当容易地向任何有基本数学背景的动机听众解释的情况下，这种无知仍然存在。

在关于算法和复杂度的流行演讲中，我展示了两个名字列表，并询问观众哪个更熟悉。第一个列表有约翰内斯·古腾堡、约瑟夫·雅卡尔、托马斯·爱迪生和詹姆斯·瓦特等名字。在我成长过程中，他们被当作文化英雄，因为他们的发明在当时极大地推动了社会的发展：分别是印刷机、织布机、灯泡和蒸汽机。大多数普通观众都认识他们及其发明（尽管年轻观众认识得较少）。我展示的第二个列表有埃德加·迪杰斯特拉、唐纳德·克努特、约翰·图基和伊尔温·伯莱卡姆等名字。在这里，我几乎无人知晓。然后，我只是为了给人留下印象，展示了这些算法英雄及其合作者开发的相应算法——最短路径、字符串匹配、快速傅里叶变换和里德-所罗门解码——的几行伪代码。我指出，就像之前提到的物理设备的古老发明一样，这些微小的智力创造非常聪明和高效，但它们的影响可能更大。为了证明这一点，我最后讨论了（一些）这些算法的变体使它们得以实现的行业：分别是自动导航、计算生物学、医学成像以及所有存储和通信设备。

它似乎在我看来，这样的故事在中学校园里也是激动人心且易于理解的教材，理论界应该有充分的动力将其引入其中。当然，这些只是例子，还有很多其他的。这类算法的典型例子包括RSA，它是大多数电子商务的基础；PageRank，它是互联网搜索的基础；以及反向传播，没有它，机器学习中的“深度网络”革命（我们很快将讨论）将无法实现。在这些例子中，有些人可能认识这些算法（或围绕它们的炒作），但并不一定认识它们的发明者。最后，还有更多基本的算法原理和结构，用于高效的数据组织和访问，包括哈希、缓存、草图、堆、树和词袋，以及其他许多，[26]所有这些都是许多算法和应用的基础宝石。

20.6.2 Algorithms and Moore's law

另一个需要讨论的问题是技术[27]和算法对应用相对贡献，尤其是在未来。显然，只有当计算应用足够高效，对用户有用，对开发者和制造商有利可图时，它才变得可用。我们还能提高多少效率呢

[26]The reader can find details on all of these on Wikipedia and other Internet sources.
[27]Here, in the sense of physical infrastructure of computing systems.

硬件和软件？关注最基本资源，其效率至关重要——速度和大小——阐明了硬件设计科学技术是多么令人难以置信。在被称为 *Moore's law* 的东西中，Gordon Moore 在 20 世纪 60 年代预测，集成电路上的元件数量每 18 个月翻一番。关于计算速度增加的类似预测也被提出。奇迹般地（至少对我来说），这种指数增长以预测的速度持续了半个世纪！但不再是了。

无论是摩尔定律近期放缓是它即将死亡的信号，还是它将与我们相伴更长时间，技术都有*absolute*物理极限。组件永远不会比原子更小，它们之间的通信永远不会超过光速。在那个时刻，以及可能更早的时候，效率的*only*来源将是更好算法的发明：我们将拥有哪些计算应用和产品，以及我们将不会拥有哪些，将更多地取决于算法的进步，而不是技术的进步。在这里，对于许多问题，我们所知的最佳时间和空间界限似乎远非最优，似乎有大量的改进空间。

20.6.3 Algorithmic gems vs. deep nets

所以，算法统治地球几乎是肯定的，但哪些算法呢？技术已经为我们提供了非凡的计算能力（以及非常重要的算法进步），这使得从机器学习中出现了一种新的算法。简而言之，我们现在有了作为技术产品的算法。科技公司、医疗公司（以及政府）在这些新算法上投入了巨额资金和努力，我现在将描述这些算法（并在第20.7.2节中更广泛地讨论）。

与上述由人类设计的优雅、简洁的算法瑰宝形成鲜明对比的是，许多新算法简单地“自我创造”，人类干预相对较少，主要通过与大量数据交互。这种与经典算法设计的对比更加明显，因为这些新算法通常（也许必然）规模巨大，人类对其理解非常有限。我当然指的是“深度网络”（参见[GBC16]中关于其在学习和优化中应用的详细文本），这是模仿神经元网络的启发式算法的通用名称.[28] 我以对这些算法及其提出的挑战和问题的几点评论来结束本节。请注意，尽管第17章的主题是机器学习，但该章侧重于理论模型和简单任务，而不是这里提到的复杂任务和启发式算法。

这些自学算法试图解决许多往往难以正式定义的问题，例如在通常极为嘈杂的巨大数据集中寻找“显著信号”——无论是金融、天体物理、生物还是互联网。可能想要提取/揭示的结构可能是聚类、相关性、几何或数值模式，或者是完全出乎意料的。例如，这可以代表图片中的熟悉面孔或物体、社交媒体中的朋友群体和兴趣、股票买卖中的公司市场表现、生物数据中治疗或遗传缺陷的影响，以及物理观察中物质和能量的启发式相互作用。{v*}

这并不是一个适当描述深度网络及其训练过程的地方。只需说，如今，它们的大小可以达到数百万个门和门之间的线。每个连接都有一个强度参数，训练过程试图优化它。简而言之，训练是一个具有不明确目标和大范围启发式搜索自由度的巨大优化问题。这种“黑魔法”到目前为止只在相对较少的情况下非常有效。但在越来越多的案例中，它大大优于任何人类设计的算法。当这样一个自学成才的程序能够像人类一样几乎同样好地标记人类拍摄的任意图片的关键内容时，这非常有趣。另一个这样的程序在与自己玩了一亿局围棋或象棋（只知道游戏规则）之后，现在可以击败世界上最优秀的人类玩家，这非常令人印象深刻。这类程序在人类语言理解和翻译方面取得了巨大进步，也许它们最快的增长是在“数据科学”领域，在这些程序在科学发现中扮演着越来越重要的角色。事实上，人们不禁要问，它们将在何时能够更好地

[28] There are many other types of heuristics, past or future, to which this discussion is relevant as well. However, deep nets seem to have become dominant in the past decade, so I'll focus on them.

阐述新发现的科学定律并追寻其后果，如医疗药物和治疗或食品和能源供应，也无需人类帮助。毫无疑问，每个人、每个政府以及整个人类社会都应该对这些机器（这些机器才刚刚开始解开）的非凡能力所引发的社会问题、潜在危险和承诺进行深入思考、考虑和讨论。

主要观点是我在这方面的迷人和具有挑战性的，ToC可以也应该为此做出贡献，即对这些算法的理论理解。这种理解应包括使它们的训练更具原则性和效率的方法，以及开发评估它们性能和输出质量的模式和工具。一个主要挑战是理解为什么深度网络在哪些任务上成功，以及它们的局限性是什么。此外，鉴于它们在人类至关重要的系统中的日益普及，探索如何使它们*fair* (以及这意味着什么)，它们对对抗数据的敏感性以及如何保护免受此类数据的影响，具有至关重要的意义。当然，深度网络的大小和复杂性可能限制了对它们理论理解的深度（就像自然中的巨大创造，如大脑和其他生物系统一样）。事实上，深度网络的理论研究（与动物不同，实验不受赫尔辛基宣言指南的限制）有助于更好地理解生物系统，反之亦然。

考虑到我们对许多良好设定的重大问题（即上下限之间的差距）的当前无知，我毫不怀疑，还有大量算法宝石等待我们去发现，这些宝石我们能够轻松理解、欣赏和使用。为了强调这一点，让我在结论中提到，没有几个伟大的算法宝石嵌入在几乎所有深度网络中，包括极其高效的 *back propagation* 和 *gradient descent* 算法，深度网络将无法存在。

20.7 Some important challenges of ToC

即使在高层次上，TOC的重要挑战太多，无法在此一一列举，尤其是在其与计算实际世界的广泛联系及其与科学领域的广泛联系方面。此外，许多重要猜想出现在本书的不同章节中。在此，我想特别指出TOC复杂性理论核心中的四个元挑战，这些挑战有众多化身和表述（其中一些在书中讨论过）。这些（自然相关）挑战是证明不可解性、理解启发式算法、加强密码学基础和探索丘奇-图灵论题。

20.7.1 Certifying intractability

迄今为止，计算理论（以及数学的一个主要挑战）的最大挑战是确立*some*，实际上任何自然计算任务对于通用计算模型来说都是*nontrivially*困难的。这个问题有许多化身，但令人遗憾的是，尽管我们真正关心的几乎所有自然任务似乎都是指数级困难的，我们甚至无法建立超线性下界（即，给定任务比仅仅读取输入更难）！当然，$\mathcal{P} \neq \mathcal{NP}$猜想是这种挑战最流行的表现（例如，证明某些$\mathcal{NP}$-完全问题如布尔公式的可满足性或地图的三色问题的下界）。同样，即使是对于整数乘法或离散傅里叶变换等更基本（且实际迫切）的问题，我们也无法建立非平凡的下界（如果为真）。在80年里，我们一直有一个可以用来证明这种下界的正式数学模型（以及当然，做这样的事情的巨大动力），这证明了这个主要问题的难度！

本质上，在关于 *proof complexity* 的问题上存在相同的悲伤状态：我们不知道关于例如 *any* 命题重言式的弗雷格证明长度的非平凡下界。这一追求体现在 $\mathcal{NP} \neq \text{co}\mathcal{NP}$ 猜想中。同样，在 *arithmetic complexity* 中，我们不知道计算自然多项式（如永真式）所需的算术运算（求和和乘积）数量的非平凡下界；这一追求体现在 $\mathcal{VP} \neq \mathcal{VNP}$ 猜想中。对于许多其他模型和计算方式，情况也是如此。我们无法证明困难！

第五章中讨论过，我们有一种形式为 *barrier results* 的“借口”，它部分解释了为什么当前技术无法提供非平凡的布尔下界。但我们并不真正理解为什么我们还没有找到绕过这些障碍的技术。此外，对于证明复杂性和算术复杂性，我们甚至没有任何形式为障碍结果的理由。在所有模型中，下界可以相当整洁地以各种形式表达为组合和代数问题，这些问题与其他许多在这些或其他数学领域（尤其是对于受限模型的相同下界问题）中已解决的问题没有明显的区别。因此，理解 *why* 证明非平凡下界如此困难是该领域最大的挑战之一，而抽象地探究这个谜团可能有助于证明它们。当然，必须提到两种解释，尽管很少有人（如果有的话）相信它们。首先，这些猜想是错误的，不存在下界，一切计算和证明都极其简单。其次，这些猜想实际上与数学的逻辑基础（例如，集合论中的ZFC公理）无关。因此，从可证明的角度来看，它们可以是“我们愿意”的真或假，尽管在“现实世界”中它们只有一个特定的真值。排除逻辑独立性的第二种选择，即使没有解决这些猜想，也是一个极其有趣的方向。

最简单的解释（可能也是正确的解释）是，为通用模型证明下界确实是我们所知的 deepest 和 most difficult 数学问题之一。因此，可能需要更多的基础工作，在各种各样的有限计算模型上，才能真正理解通用计算模型及其限制。在所有领域的挑战中，这是我最关心的一个，希望它将继续吸引优秀的研究人员来发展证明这种下界所需的必要结构理论和技巧。我确信，这种理解不仅会揭示计算的局限性，还会揭示其全部力量。

20.7.2 Understanding heuristics

我们通常得到的比我们（理论上）应得的要多得多。许多算法在*some*实例中表现极差，但在“典型”实例上却表现出色。让我们讨论一些示例家族。

- 某些算法对于各种问题似乎在时间消耗或输出质量方面表现得远比其理论分析（如果有）所保证的要好。这类*heuristics*的典型例子通常包括SAT和大规模实例上的TSP求解器，以及线性规划的单纯形算法。当然，这些启发式方法非常巧妙，经过精心优化和微调。

- 在20.6.3节中更详细地讨论了，由于新能力训练深度网络来解决学习、优化和游戏问题，涌现出大量的启发式方法。在这些情况下，也存在微调的过程，但在那个神秘的自教“训练”过程中，它主要被自动化。在许多（但肯定不是所有）由各种应用动机引起的问题中，这些启发式方法在许多自然数据集上优于精心设计的算法或人类专家。

- 另一个性质完全不同的例子如下。概率算法旨在在假设它们使用的随机抛硬币是完美的（即独立和无偏）的情况下正确工作（以高概率）。但它们在接收到以某种任意确定方式生成的位序列时，或者从物理源（如互联网流量或用户按键）中获取时，通常也能表现得同样好。这种现象（可能是令人惊讶的）在物理学家使用的众多蒙特卡洛模拟中发生。

- 最后，自然界本身为我们提供了许多快速解决“难题”的过程，如蛋白质折叠、泡沫形成、市场均衡和群体协调行动。当然，我们通常并不真正了解自然界的算法，但可能对其有模型，这些模型在许多情况下无法预测或保证观察到的（高质量）性能。

[29] This can happen, for instance, if a polynomial-time algorithm for SAT exists whose correctness is independent of ZFC.

一般但模糊的解释是，这些算法和过程的所有 *inputs* 都来自某些集合或分布（自然或人类产生），对于这些集合或分布，给定的启发式方法比对于任意输入的效率要高得多。启发式方法对它处理的“现实世界”输入的适用性可能有多种原因。在某些情况下，算法设计者设法掌握了实际的（或相关的）输入分布，并设计了算法，使其在这样输入上非常高效（尽管在一般情况下可能较慢），就像一些TSP和 SAT 求解器所发生的那样。在某些情况下，算法被设计得自然而优雅，如单纯形算法，它在现实生活中的线性规划上的出色性能令人惊讶，确实很幸运；尽管后来有各种尝试来解释它，但我们对其理解仍然不足。在其他情况下，算法 *evolved* 以适应其输入分布，就像许多神经网络和其他（监督和非监督）学习框架所发生的那样。此外，在自然界中，这种学习和进化无处不在，在许多情况下，算法和输入 *co-evolved* 都可能产生优越的性能（例如，可能存活下来的蛋白质是那些在我们的细胞中通过算法容易折叠的蛋白质）。

理解在“实践中”出现的各种计算任务的实际输入分布，并设计能够很好地处理这些分布的算法（或解释为什么某些启发式方法有效），是该领域非常古老且基本的问题。几十年的算法研究已经提供了从不同角度解决这个重要问题的理论模型和框架。一般方法包括算法的概率分析[Kar76]、半随机模型[FK01]、平滑分析[ST04b]、稳定实例[BL12, BBG13]、具有某些矩界限的输入分布[HS17, KS17]等。[30]其中一些方法得到了开发、扩展和显著应用。除了这些一般方法，它们适用于广泛的算法任务家族之外，还有许多论文针对特定任务，针对并利用它们的“自然”输入分布。这种建模输入和算法设计之间的交互非常重要，并有望更好地理解这些算法在“相关”输入上的性能（以及其他行为属性）。

随着与数据互动并不断改进的机器学习算法的出现，越来越多的启发式算法（在许多应用中表现极其出色）的性能难以解释。这些以及其他现有的启发式神秘现象对开发评估此类算法实际性能和行为的标准提出了巨大挑战，我们已在第20.6.3节中讨论过。可能缺少的是基准测试理论，这将允许比较启发式算法，并在将学习系统参数应用于数据之前提出合理的指导方针。当然，缺少的是解释哪些类型的问题和哪些类型的数据，例如深度网络擅长解决，哪些不擅长，以及原因的理论。这种理论的需求不仅是理论上的！我们很快将不得不*trust*这样的算法（当它们取代人类时，例如在驾驶汽车和提供医疗诊断和治疗时），并且信任我们不理解算法与信任我们理解的算法是非常不同的事情。现有的大多数计算机系统属于后者，但平衡正在改变。我相信（除了性能之外），应该并且将会开发出模拟程序输出“社会方面”的理论，并且公共领域使用的启发式算法将不得不提供一些保证。这些问题已经进入公共政策和法律讨论（例如，关于通过用算法取代法官来提供更好正义的想法）。

显然，期待一个能够产生理想、*instance optimal*性能的完全预测性的算法设计理论是困难的。然而，人们越来越容易完全忽视任何理论分析，并满足于由通用启发式方法（如深度网络）给出的输出，这可能是危险的。我个人的感觉是，关于对现实世界数据建模、设计通用算法技术以高效准确地解决这些问题以及评估启发式方法的性能，还有很多理论理解可以获取。这个挑战对于该领域和社会都是核心的。我预计，与自然科学日益增长的互动将为这一研究带来新的问题和新的想法。

[30]Including average-case analysis [Lev86], which focuses on hardness rather than easiness.

20.7.3 Resting cryptography on stronger foundations

密码学在第18章中进行了详细讨论。到目前为止，密码学是对不可解假设的显著后果最广泛和反复令人惊讶的研究。此外，这些后果（以及因此所需的假设）构成了计算机隐私和安全以及电子商务的当前基础，这些被个人、公司、军队和政府所使用。人们可能会想象，社会会对这些基础进行测试，并为潜在的脆弱性做好准备，至少与为核反应堆事故、地震和其他潜在的巨大灾难源做好准备一样好。

然而，如第18.11节所述，尽管大多数密码学需要存在 *trap-door* 函数，但我们对于这些函数的候选者却寥寥无几，其中整数分解仍然是主要的一个（其他重要的候选者基于椭圆曲线、格和噪声线性方程）。此外，尽管尚未发现这些问题容易解决，但我们没有任何证据表明其中任何一个实际上很难。例如，对于这些少数问题中的任何一个的效率算法都不会意味着大量其他被认为难以解决的问题的效率算法，也不会意味着任何复杂类别的崩溃。

我发现这种状况非常令人担忧，并认为应该投入更多的努力来应对寻找更安全密码学基础的挑战。如果这些基础崩溃，世界将失去太多。应该强调的是，密码学研究正在蓬勃发展，主要扩展了比门函数甚至更强的（因此，更不可信的）不可解性假设的应用和后果的前沿。自然，应用通常是在强或特定假设下首先发现的，这些假设后来被削弱。尽管如此，仍需谨慎，尤其是在将实际系统的安全性建立在摇摇欲坠的假设之上时。

当然，这种状况（即拥有少量可靠、有用的假设）的部分原因并非缺乏尝试：这个挑战确实很严峻。鉴于证明任何事物超线性下界的第一个挑战，我们目前还不能期望对分解有超多项式下界。一个自然的方向是简单地开发更多陷阱门函数的候选者，以防其他函数被破解。在这里，困难似乎在于这些问题似乎“要求”的（通常是代数）结构。另一个方向是开发陷阱门函数之间的归约，以表明任何一种函数的效率算法都有非平凡的影响。在这里，严格的结构似乎也抵制归约。最后，将密码学建立在更可信的不可解性上，最好是 $\mathcal{P} \neq \mathcal{NP}$ (或者至少是其平均情况版本)，会遇到对黑盒使用此类假设的不可能性结果。Barak [Bar01] 是第一个超越这种黑盒使用的人；此后还有更多。但似乎进一步发展非黑盒密码学技术（其中确实有令人兴奋的活动）对于这个挑战是至关重要的。

20.7.4 Exploring physical reality vs. computational complexity

著名的丘奇-图灵论题总是让我觉得这是一个惊人的大胆挑战。毕竟，它基本上断言：

Everything computable is computable by a Turing machine.

“一切”是指任何具有离散输入和输出域的明确定义函数。这篇论文挑战任何人，尤其是科学家，提出这样一个明确定义的计算任务，这个任务可以在现实世界中以某种方式执行（包括准备输入和读取输出），使用任何物理手段，但不能由图灵机（可以想象的最简单的物理设备之一）计算。然而，在过去的80年里，没有人对这个挑战做出严肃的回应。

不久在20世纪70年代复杂性理论被开发之后，人们迅速将这一论点扩展到一个更为大胆的论点，即即使我们将自己限制在*efficient*计算中，它仍然成立。这有时被称为“强”或“可行”的丘奇-图灵论点，本质上表述为：

[31] The choice of proper cryptographic assumptions is itself a field of study in cryptography, discussed, for example, in [GK16] and its references.

Everything efficiently computable is efficiently computable by a Turing machine.

这里“高效”通常等同于“多项式”（尽管可以采用更严格的效率度量）。这意味着在计算过程中消耗的物理资源（如能量和质量），以及用于计算过程的时间和空间，都允许仅以输入大小多项式增长。

这篇论文，连同 $\mathcal{P} \neq \mathcal{NP}$ 猜想一起，提出了一个看似更具实质性的挑战：简单地找到一个自然过程，该过程可以用多项式资源解决一些 $\mathcal{NP}$-完全问题。确实，这里出现了许多建议，其中许多（但肯定不是全部）在 Aaronson 的美丽调查 [Aar05] 中得到了解释和讨论。正如我们在第 3.10 节中讨论的那样，许多这样的提议加强了将 $\mathcal{P} \neq \mathcal{NP}$（以及更一般地，计算复杂性考虑）作为模拟自然现象的指导原则的论据。[32]

然而，在这里我想转变由强大的丘奇-图灵论题带来的挑战，而不仅仅将其视为科学家们要么反驳它，要么在模拟自然过程时使用计算复杂性的挑战，而是将其视为对复杂性理论家的挑战，即计算性地抽象和模拟那些（潜在的）自然赠礼，这些赠礼最终可能被用来增强我们的笔记本电脑。应对这一挑战（或者更确切地说，挑战，因为它们来自自然界的各个方面）已经对关于高效计算和算法的基础性问题以及实践产生了极其重要的影响。本书中探索这类赠礼的两个主要例子占据了几个章节：随机性的赠礼和“量子性”的赠礼。我将从这些开始，然后提及一些其他不太发达的赠礼。

首先，自然界似乎为我们提供了大量的不可预测现象，这导致了概率算法、概率证明、伪随机性和随机提取等极其重要的（理论上和实践上）理论。前两种理论假设可以访问 *perfect randomness*，即一系列无偏、独立的硬币投掷，以增强计算能力。这些极其丰富的理论分别在第七章和第十章中进行了详细讨论。后两种理论探索放松（以不同的方式）这一强假设（即可以访问完美的随机性），这在现实世界中可能会被违反。一种（在第7.2节中描述）探索了无法访问随机性 *at all*（并用“生成不可预测性”的普遍信念的困难假设来代替。另一种（在第9章中描述）假设我们只能访问非常弱的随机源，例如高度偏斜和相关的比特序列（并解释了如何提高它们的质量）。

现在，计算机程序中“随机”序列的实际应用（例如，在天气预报和核相互作用中的蒙特卡洛模拟）早于所有这些理论，而且似乎在许多情况下，使用何种“随机”源来产生“有用”的结果并不重要（包括确定性源！）。但当然，在这些情况下没有保证，也没有理解何时这些源“有效”何时无效。这两个理论为研究概率算法的随机性要求提供了正式基础，证明了它们在超越原始完美随机性要求的设置中具有可证明的适用性。这些理论为选择“随机”源提供了依据和指导，确定了不应使用的源，并找到了提高弱源效率的有效方法。从这些理论中得出的去随机化结果（特别是$\mathcal{BPP} = \mathcal{P}$）直接影响了丘奇-图灵论题：它们证明了用概率性图灵机（以小概率出错）替换确定性图灵机是有效计算*is.*[33]的定义。除此之外，正如我在本书中讨论的，对随机性的抽象研究在许多意想不到的方式上对计算理论（和数学）的理论和实践非常有用，远远超出了原始意图。

接下来，自然界实际上似乎为我们提供了量子力学现象，其中吐出完美的随机抛硬币结果是其中最简单的力量之一。确实，这种力量似乎承诺了高效操纵（某些）指数长度的信息！这个承诺导致了使用此类量子现象进行计算的理论研究，量子力学图灵机的制定，以及对其力量和限制的研究（第11章致力于这个令人兴奋的研究领域）。在这里，理论远未完全发展。尽管我们有形式化的通用模型和它们之间的等价关系，

[32] Which in many of these proposals also bring out the need to understand the actual inputs fed to these “heuristics of nature” in the sense discussed in the “heuristics challenge” in Section 20.7.2.

[33] This does not belittle the importance of using randomness, when it gives polynomial speed-ups.

他们，对某些量子（退相干）错误纠正、量子密码协议以及与物理学家的大量互动，我们只有少数珍贵的问题，这样的量子计算机可以高效解决，而经典（概率）计算机则不能.[34] 幸运的是，这些少数问题包括整数分解和离散对数，因此这个模型破坏了大多数加密安全系统的当前基础。当然，在这种情况下（与概率计算不同），理论先于实践，算法潜力导致了众多极具挑战性（且昂贵）的实际项目，即尝试将这种量子增强添加到我们的笔记本电脑中，并再次修订我们对高效图灵机的理解。

陪审团尚未确定哪种双赢的结果将实现：要么我们将拥有这样的量子笔记本电脑（显著增强我们的计算能力），要么我们将发现我们自然模型中的基本差距，这阻止了它们的产生（并且，希望修正这些模型并制定更好的模型）。但除了激励和协助实际项目外，量子计算的理论、证明、建议、学习、游戏等在许多意想不到的方式上对*classical*计算理论具有极大的启发作用（导致下界、经典问题之间的量子缩减、用于计算委托的无信号PCPs、随机性认证等）。在这里，与概率计算不同，人们相信量子计算*does*为图灵机增加了力量（即，$\mathcal{BQP} \neq \mathcal{BPP}$）。我发现发现更多的问题在差距中，以及尽管如此，$\mathcal{NP} \not\subseteq \mathcal{BQP}$的原因也是一个极具挑战性的计算复杂性理论方向。

我们能为我们笔记本电脑、网络或其他算法添加什么，以增强对高效可计算内容的认识？自然界还有许多其他礼物，可能比上述两种更具体，它们已经被转化为计算模型和范式，其算法能力和限制正在被探索，而许多其他模型正在被追求。正如所讨论的，我发现除了对自然（以及许多其他）模型的自然科学兴趣，以理解自然之外，它们作为计算模型的纯粹理论追求将有利于对计算本身的更广泛理解，也许将导致新算法技术的发现，可能最终导致新计算设备和系统的创造。本章提到了许多具有这种潜力的自然现象的例子。因此，我期待着从研究这种计算设置中产生的力量、限制和关系的发现，这些关系可能被形式化为DNA机器、纳米机器、自私网络或自然算法（以及其他许多），丰富计算实践及其理论。其中一些可能导致像nano-$\mathcal{P}$、selfish-$\mathcal{P}$这样的类别，以及可能的概率、量子、非确定性、空间限制、非均匀等变体，以及与先前研究模型的新联系，这将加深我们对计算的理解。

20.8 K–12 education

这是一个巨大的主题，因此这里的讨论相当肤浅且不集中，缺乏适当的参考文献（可能确实存在）加剧了这种情况。或许可以将其视为一个简短的愿望清单，并呼吁TOC专业人士参与这一努力。

ToC 在使每个人的教育课程（所有级别）变得更好、更丰富、更有价值、更有趣方面提供了如此之多，以至于需要一本书来讨论这一点。当然，这里没有原创性，而且有许多思想和正在进行的项目（有文章描述它们）表明，以各种方式将“计算思维”整合到各个年龄段课堂上的努力。像往常一样，即使是非常好的想法最终也必须得到实施。与其他学科一样，除了决定教什么和怎么教孩子，最不平凡的任务是弄清楚如何培训教师以热情地向学生传授这些材料。为了在原则和实施方面做出更好的决定，我强烈希望我们社区中愿意这样做的人参与这些发展。这个挑战至少与探索我们领域的科学挑战一样重要。

以下我列出了一些我希望所有孩子都能在适当年龄以某种方式学习的真理的小样本（）。

[34] Beyond the basic problem of simulating quantum systems.

将激发他们，并通过适当的例子和演示让他们（无论他们最终追求什么生活兴趣）永远记住它们。我相信这个样本捕捉了所有受过教育的人都应该具备的关于计算的最小知识和理解，就像我们期望他们具备其他主题（如基础科学、历史和社会系统）的基本知识和理解一样。

- 图灵，像爱因斯坦一样，是20世纪科学巨匠之一。他对技术和社会的影响远远大于爱因斯坦。
- 计算机不能做任何事情。确实，有一些基本的、期望的计算任务它们永远无法完成。这是一个绝对真理（一个数学定理），就像毕达哥拉斯定理一样。
- 计算机能做什么取决于聪明、高效的算法的发现。这样的算法可以创造一个新行业，或者预测并预防一场流行病。算法的大发现者与伟大的发明家，如古腾堡、爱迪生和巴斯德，同列于同一荣誉殿堂。
- 自然理论应该是可预测的。预测是一种算法！预测天气或自然灾害*before*它们发生需要这些算法非常高效，*faster*比自然还要高效！
- 大多数数学问题并非只有一个正确答案（其他都是“错误”的）。就像生活中一样，许多数学任务有不同的解决方案质量。找到任何解决方案都是一个良好的开端，而尝试改进它则更好（这将在下面进行更多讨论）。
- 数值（例如，十进制或二进制）表示系统是我们用来表示数字的，它被发明并延续了数千年，仅仅因为它提供了极其高效的存储和执行算术运算的算法。将其与一进制系统或罗马数字相比……
- 我们小学学习乘法的方法首先是一个 *algorithm*，而且，它还是已知最快的算法 *not*。有一个更快的方法，但仍然比我们用于加法的算法慢。乘法是否本质上比加法更难，这是一个伟大的智力之谜，具有重大的实际意义。
- 世界經濟假設計算難度（在這個時候，這是“逆運算乘法”——尋找大數的質因數——是一個不可能的難題的假設）。此外，如果這個（或相關的）計算假設是錯誤的，世界經濟可能會崩潰。

一个更大的问题，我认为其中TOC教育有很多可以提供的，是小学和中学的数学教育。关于这个主题已经有很多著作，我将简要地讨论这一点。要了解更多详细内容，请参阅Fellows [Fel93] 的这篇文章，他也在为小学开发课堂材料投入了大量努力，并且还尝试了交付它。

数学教育的一个公认的主要问题是过分关注算术和代数，反复进行大量等效且枯燥的数字练习，这些练习的答案要么正确要么错误。相比之下，从幼儿园开始，离散对象的算法问题提供了大量的问题，例如游戏、谜题和迷宫，这些问题提供了各种质量和特性的解决方案。例如，在优化问题中，每个解决方案可能都是“正确”的，但它们的成本不同。或者，许多不同的算法、策略和方法可以解决一个问题，但它们消耗的时间或其他资源不同。在这样的问题中，学生的解决方案邀请挑战*can you do better?*，这比*wrong!*更具建设性。这类问题也与所有学生在解决生活冲突时的基本人类经验有更好的联系，从安排他们的活动到最大化某些事物（兴趣、快乐、他们父母的满意度，或玩得更好的象棋或足球）。这类问题也与上述提到的自然 *processes*（自身算法通常解决优化问题的观点有更好的联系），这应该在科学课程中强调，将数学与方程式联系起来，而不仅仅是通过方程式。存在不同质量解决方案的问题，以及寻找更好解决方案的探索，使学生能够更深入地理解，我相信，他们也会更有趣。此外，这类问题将数学展示得更好，

一门 *modeling* 科学，它允许精确地表述这样的生命和科学情境，以及诸如机遇、不确定性、部分信息和逆境等方面的内容。我认为将这些想法融入数学教育中，首先是教师，然后是学生，这一点非常重要（并且我没有低估这项任务的难度）。这些材料应该是信息时代每个年轻人的知识组成部分，同时，教授这些内容也可以帮助转变许多代人对于数学这一学科的悲伤、消极态度。这些想法并非原创，实际上在很多地方都曾尝试过。我只是在呼吁更多 ToC 专业人士的参与，对他们来说，这种思维方式是如此自然，以开发这一重要事业的材料和实施细节。

20.9 The ToC community

我相信该领域的显著成功，部分总结在本章中，并不仅仅归因于ToC的智力内容和其研究人员的原始才能，还归因于ToC社区的组织和自身建设方式。本节致力于这一过程及其特征，正如我所经历的那样，通过在我职业生涯各个阶段积极参与所有相关活动。当我知道的例外很少时，我允许自己进行概括，充分意识到我的观点是有偏见的，并且我倾向于理想化。

自20世纪60年代以来，理论计算机科学（ToC）的许多活动都集中在两个年度会议上，即FOCS（计算机科学基础）和STOC（计算理论研讨会）。[35] 分别在秋季和春季举行，这些论坛成为了整个社区的聚会场所（以及跳动的心脏），其中很大一部分人前来听取过去6个月该领域最激动人心的进展报告。这些报告由一个程序委员会决定，该委员会没有时间审稿，而是评估它们的创新性和对领域进展的影响。这些委员会有两个影响。一是委员会会议本身：大约20位来自不同研究领域的专家（以及不同的年龄！）定期聚会，讨论提交的论文，做出艰难的选择决定，帮助阐明贡献，并塑造方向。二是这些会议定期吸引了大量ToC社区的成员，包括许多研究生。他们对领域提供的挑战的持续智力兴奋，以及无疑的资深人士的个性，创造了一种我愿意列出（并理想化）的文化特征。它们无疑在迄今为止领域的成功中发挥了重要作用，随着领域的增长和多样化（这可能并非易事），保留和加强其中许多将是件好事。不言而喻，其中许多是其他学术学科的一部分，但我发现它们在ToC中的结合非常独特。

- *Level of discourse*. 需要说服一个程序委员会，其中许多成员不是您提交内容的专家，这促使提交的内容包括对所描述研究的高层次动机的详细介绍。此外，这也影响了技术发展的直观解释的包含。由于篇幅限制，提交的内容因此专注于思想和概念，这有助于它们传播给非专家。同样的效果也影响了讲座。由于许多听众是非专家，动机和直觉占据了简短讲座的重要部分。因此，论文和讲座有助于突出思想和概念，这些思想和概念容易被许多人理解。此外，这种讨论水平允许将一个环境中开发的思想和概念应用于其他环境，可能是针对完全不同的问题或应用。这种共同的高层次语言，以及频繁交流最新的重要发现，无疑对领域的快速发展负有责任。我相信这也影响了ToC教师教授课程和撰写书籍、综述的方式，在技术定义和证明之前和之后突出动机、思想、概念和直觉。

- *Openness and evolution*计算之丰饶之角，其来源和表现的多样性，以及其研究者的各种兴趣，共同产生了完全新颖的

[35] These conferences generally take place in North America, and I have attended most of them since 1980. Other conferences that encompass most areas in ToC, with some biases or different geographic locations were later added with a similar broad appeal, including ICALP, MFCS, SODA, and ITCS. What I say here is relevant to them as well.

模型、问题和研究方向。这些新的兴趣在社区中争夺对现有和更成熟的方向以及开放问题的关注。这落在了相同的机构——这些会议的程序委员会——身上，成为新旧平衡的仲裁者，以及选择哪些部分接受和展示。这些不同年龄和品味的研究者之间的频繁讨论和辩论，由于有了共同的语言，不仅让实际参与者了解该领域的广泛活动及其同事们的观点，而且还创造了一个必须反复在这些趋势中做出艰难选择的论坛，即哪些趋势将变得可见以及它们的比例。回顾趋势和方向的历史演变，可以观察到极大的敏捷性去追求和调查，以及极大的开放性去拥抱和宣传全新的想法、模型、联系和问题（源于技术、科学或纯粹抽象的考虑）。这创造了一个美好的动态，孕育了许多在这些论坛中孵化出来的新研究方向。其中许多（例如，本书中大多数章节标题都是完美的例子）已经足够庞大，可以建立自己的专业会议和社区，通常以相同的方式进行运作。当然，也有一些其他研究方向缩小了规模、消亡或与其他研究方向合并。

- *A common core*尽管这个焦点领域的话题不断进入和退出，但核心主题，主要在领域的核心算法和复杂性理论领域，保持了长期的存在，即使它们在增长。许多基本原理（在ToC中具有广泛的应用性）来自于对纯粹数学的、*seemingly*“非动机”模型和问题的研究，这些模型和问题似乎是对现有模型的“不切实际”或“不自然”的推广，或者只是从左边的领域出现的。简而言之，我们遵循了领域的内部逻辑和美学，这在数学中很常见，但在应用驱动领域则很少见。社区对这些想法的开放性得到了巨大的回报，正如本书许多章节所例证的那样，并不断重新确立核心的价值。另一个使核心成为可能并加强的核心因素是，尽管覆盖领域的多样性大幅扩张，但仍然保持了上述讨论水平和风格。自然地，这种多样性需要会议结构和程序委员会规模的变化，但它仍然允许在不同动机和结构的研究方向之间以及这些方向之间交换想法、概念和方法。此外，这种交流的速度、流动性和质量已经阐明了ToC是一个多么具有凝聚力的领域，以及拥有一个核心和基础设施来利用它是多么重要。在许多情况下，这种想法、方法和结果的自然流动非常自然，而在许多其他情况下，它需要非凡的独创性和当然是对其他领域发生情况的了解。在关于这种知识价值的讲座中，我称这种现象为*depth through breadth*，我认为它在我们的领域中是独特的。这些许多线索的共同点在方法第20.3节中得到了捕捉。

- *Community and attraction of talent*具有挑战性和重要性的长期目标；一个丰富多样的概念、思想和工具的织锦，它们在连接和扩展方面取得了令人兴奋的进展；构建理论以逐步获得这种理解；以及科学领域的许多其他智力方面，这些都是持续吸引该领域年轻人才并进一步这项工作所必需的。确实，计算理论拥有这一切以及更多，正如本章以及整本书所展示的。但我认为，该领域的社交和教育结构使其对年轻人才进入、留下、成长以及希望发现、教育和领导更加有吸引力。我已经解释了话语水平如何使它对新来者来说既容易又吸引人，尤其是那些在本科和研究生阶段接触过它的人，以及对于对这一领域感兴趣的不同领域的研究人员也是如此。另一个重要方面是该领域的“民主”和社会欢迎，它通常考虑的是原始想法，而不是提出这些想法的人的资历。我作为一年级研究生的一段形成性经历是在1980年的FOCS会议上被介绍给理查德·卡普，*the*该领域的最杰出的领导者。“迪克，这是阿维，阿维，这是迪克”，随后是五到十分钟的时间，卡普带着真正的兴趣探索了我喜欢哪些理论领域以及我可能追求哪些领域。我很快了解到甚至不需要介绍，

该领域的领导者在你带着问题或想法去找他们时同样容易接触。我相信这是新进入该领域的人至今仍能感受到的经验，并且它创造了一个充满同侪精神、合作和友谊的社区。尽管自然且受欢迎的竞争和动力，这种氛围依然存在。

Critique

在这些许多积极方面之后，我认为提及一些我认为我们的社区表现不佳的问题是有意义的，希望随着时间的推移这些问题会得到改善。它们都是相关的，可能源于一个单一的根本原因，即过于关注领域的智力问题，在一定程度上忽视了我们在其中生活和运作的外部领域的必要投资。最显著的忽视是在与外界沟通关于ToC成就方面，包括它们的智力内容和实际影响。当然，多年来，一些ToC成员已经为普通计算机科学社区、普通科学社区甚至公众投入了普及讲座、调查和书籍。但我担心他们所取得的成就远远小于他们通过更多（完全有理由）的努力所能取得的成就。有许多方式可以证明外界对ToC创造的令人惊叹的工作成果的理解、认可和欣赏普遍较低。这种情况很容易得到改善，因为我们的许多发现具有如此吸引人的概念内容，因此很容易与几乎任何受众联系起来。简而言之，我认为该领域应将其作为优先事项，并希望在这方面做更多的工作。

一种使传达主题更容易的方法，特别是对于公众（但同样对于领域外的人）来说，就是拥有一个富有表现力的词汇来描述核心概念和结果。在这方面，远远超过其他所有领域的领域是物理学。当然，它从比大多数其他领域都大的自然优势开始，因为它们的研究对象从很小的时候就激起了我们的好奇心和 fascination，比如星星、彩虹、海浪、飞翔的鸟儿，等等。但对于我们无法直接接触的研究对象，物理学家们挑选出引人入胜和迷人的名字，如“大爆炸”、“黑洞”、“暗物质”、“超新星”，并将基本粒子的某些属性称为“奇异”和“魅力”。相比之下，ToC不仅从没有人有计算或算法经验开始，我们还用无聊、难以解释的首字母缩略词（如P、NP、BPP、SC、NC）来称呼我们的主要概念（这些只是“复杂性动物园”中最著名的动物中的一小部分）。[36]还有许多其他的选择，我认为可以更好，也许最好的例子就是“复杂性”本身（甚至在“计算复杂性”这个短语中），它过于笼统，并且经常与其他这个词的使用和含义混淆。对于我们真正研究的内容，更好的短语可能是“计算效率”和“计算不可解”。当然，有些选择既有意义又引人入胜，比如“零知识证明”、“蒙特卡洛算法”、“随机性提取器”、“无序成本”、“哲学家就餐问题”、“拜占庭将军”、“完美匹配”和“最大团”等问题。我认为，糟糕的命名选择阻碍了我们本科和研究生ToC学生的内部教育，而不仅仅是我们在领域外传达我们的工作！尽管一些糟糕的选择已经根深蒂固，可能已经太晚改变，但我认为，意识到这个问题可能会产生更好的新概念名称，并激发社区在公开讲座（在那里不需要使用专业术语）中采用旧概念的创造性的名称。

20.10 Conclusions

科学学科以不同的方式诞生（并演变）；关于这个主题已经写下了大量著作。在ToC的情况下，确定其确切诞生时刻非常容易。它是在一个独特事件中发生的，遵循了一个古老的过程。计算，以它的多种形式，自从人类文明之初就一直是研究的主题，因为资源的有效利用对人类（以及所有生命形式，甚至受物理定律支配的无生命物质）来说是第二（或第一）本能。但这项研究是

[36] A website of most complexity classes is maintained at [AGKR17].

缺少一种使其精确的形式化方法。这种形式化方法于1936年由图灵的奠基性论文提供，该论文为我们提供了*Turing machine*模型。它终于找到了金子，最终使得理论的创造成为可能！计算揭示了自己是一个无尽的宝库，而图灵机研究已经揭示的内容是科学史上最伟大的成就之一。

在这一章中，我讨论了ToC的许多方面，特别是其对人类探索和社会的过去和潜在贡献。它纯粹学术、数学的追求探索了人类提出的一些最深奥的智力问题。而且，它对大多数其他科学、技术和人类生活具有根本的重要性。简而言之，这一章以及包含它的书籍，清楚地表明ToC是一个独立且重要的学术学科。我认为这种观点应该是关于该领域及其未来的任何（内部或外部）讨论的基础。

这个目录在知识领域的位置伴随着巨大的责任。为了继续在内部以及与其他学科的扩展和动态合作中发展，我认为目录社区需要成长，而这种成长带来了新的挑战。其中一些挑战将通过这个年轻领域的精神和社交文化得到充分满足，我希望这种文化能够得到保留。其他挑战与该领域的性质变化及其多样性的增加有关，我认为需要集中精力在教育和对齐上。让我详细说明一下。

ToC 处于我预期将是一场漫长努力的初期阶段。像数学、物理、生物学和哲学等拥有数百年历史的学科一样，ToC 在未来可能会多次改变形式和重点。我发现当前时期无疑是这样的一个变化时期，可能是一个相变，因为该领域与其他科学学科的连接爆炸以及来自计算机科学和工程更大爆炸的日益增长的理论需求，以难以置信的速度产生了新的计算系统和应用技术。这些带来了巨大的挑战，因为 ToC 的性质和规模都在发生变化。我发现几个（相关的）预测是不可避免的，并且每个都要求 ToC 社区采取行动。首先，随着科学和工业中计算建模和算法设计的需求不断扩大，将需要更多专业人士接受其原理和方法培训。这些原理将成为大多数本科教育的前提，我强烈认为 ToC 必须在设计和制定此类课程和课程大纲中发挥积极作用。其次，计算理论研究和研究人员将成为计算机科学之外学科的不可或缺的部分，同时，ToC 研究（最初是计算机系统——动机问题——的纯数学追求）将需要适应和采用研究现实世界的科学和社会理论的实验方面。

这次伟大的发展和多样性，不仅体现在研究主题上，也体现在方法和文化上，无疑是件好事，确实是ToC使命的一部分。但这也意味着，保持该领域的可行、核心将变得更加困难。尽管如此，我坚信，保留这样一个核心——它将允许跨学科流动和交流关于计算的基本思想、技术和洞察——至少与过去一样重要。相反，我担心领域的碎片化会失去这个重要优势！保持ToC凝聚力的挑战无疑是复杂的，需要深思熟虑的重组和适应。除了创建物理和在线论坛以维持这种凝聚力和思想交流外，我发现该领域可以从本科教育中的行政变革中受益。具体来说，创建专注于ToC的专业、证书和其他本科课程是自然而然的。关于该领域的基础和联系有大量资料，非常适合创建丰富而具有挑战性的课程。
The value, to both academia and industry, of a student graduating with this knowledge and understanding of theory, is extremely high。展望未来，我认为鉴于ToC在许多学术部门中的日益增长的智力中心地位（尽管计算机科学可能仍然是最强的联系），对于大学和该领域本身来说，创建*separate* ToC系可能是自然且有益的；这将反映和加强该领域的独立性及其对其他领域的中心地位。

构建和维护必要的结构，如此处建议的，以保持目录结构的连贯性，只有在社区对其独立性、价值和使命有信心的情况下才能存在。也就是说，只有在目录成员充分理解过去成就的意义、重要性、范围和影响，以及未来成就的更大潜力时，这才能实现。这些概念信息应该是其不可或缺的一部分

我们提供给学生的教育（以及彼此之间的教育），远远超出了技术知识。这样的基础有望促进对该领域组织进一步的建设性讨论，我认为这对它的未来非常重要。

References

[AA11] S. Aaronson 和 A. Arkhipov. 线性光学的计算复杂性。在 *Proceedings of the 43rd annual ACM symposium on Theory of Computing*，第 333–342 页。ACM，2011。引用于 116 [Aar03] S. Aaronson. “P 与 NP” 是否形式上独立？*Bulletin of the EATCS*，81:109–136，2003。引用于 55 [Aar05] S. Aaronson. 专栏：NP 完全问题与物理现实。*ACM SIGACT News*，36(1):30–52，2005。引用于 35，261 [Aar13a] S. Aaronson. *Quantum computing since Democritus*。剑桥大学出版社，2013。引用于 114 [Aar13b] S. Aaronson. 哲学家为什么应该关心计算复杂性。*Computability: Turing, Gödel, Church, and beyond*，第 261–328 页，2013。引用于 253 [Aar16a] S. Aaronson. 量子状态和变换的复杂性：从量子货币到黑洞。*arXiv preprint arXiv:1607.05256*，2016。引用于 248 [Aar16b] Scott Aaronson. P=?NP。在 *Open problems in mathematics*，第 1–122 页。Springer，2016。引用于 26 [AAVL11] D. Aharonov，I. Arad，U. Vazirani 和 Z. Landau。可检测引理及其在量子哈密顿量复杂性中的应用。*New Journal of Physics*，13(11):113043，2011。引用于 119，121 [AB87] N. Alon 和 Boppana R. B. 布尔函数的单调电路复杂性。*Combinatorica*，7(1):1–22，1987。引用于 53 [AB03] M. Agrawal 和 S. Biswas。通过中国剩余定理进行素性和身份测试。*Journal of the ACM*，50(4):429–443，2003。引用于 87，136 [AB09] S. Arora 和 B. Barak。*Computational complexity: a modern approach*。剑桥大学出版社，2009。引用于 5 [ABBG11] S. Arora，B. Barak，M. Brunnermeier 和 R. Ge。金融产品中的计算复杂性和信息不对称。*Communications of the ACM*，54(5):101–107，2011。引用于 252 [ABF+08] M. Alekhnovich，M. Braverman，V. Feldman，A.R. Klivans 和 T. Pitassi。正确学习简单概念类的复杂性。*Journal of computer and system sciences*，74(1):16– 34，2008。引用于 200 [ABL02] S. Arora，B. Bollobás，和 L. Lovász。在不了解线性规划的情况下证明积分间隙。在 *Proceedings of 43rd annual IEEE Symposium on Foundations of Computer Science*，第 313–322 页。IEEE，2002。引用于 235 [ABND+87] H. Attiya，A. Bar-Noy，D. Dolev，D. Koller，D. Peleg 和 R. Reischuk。异步环境中的可实现情况。在 *Proceedings of 28th annual IEEE Symposium on Foundations of Computer Science*，第 337–346 页，1987。引用于 220，225 [ABO97] D. Aharonov 和 M. Ben-Or。具有常数错误的容错量子计算。在 *Proceedings of the 29th annual ACM symposium on Theory of Computing*，第 176–188 页。ACM，1997。引用于 117

[ABOE10] D. Aharonov, M. Ben-Or, 和 E. Eban. 量子计算的交互式证明。在 *Proceedings of innovations of Computer Science (ICS 2010), China*，第 453–469 页，2010。引用于 121 [ABOEM17] D. Aharonov, M. Ben-Or, E. Eban, 和 U. Mahadev. 量子计算的交互式证明。*arXiv preprint arXiv:1704.04487*，2017。引用于 121 [AC11] H. Attiya 和 A. Castañeda。k-集一致性不可能性的非拓扑证明。在 *Symposium on self-stabilizing systems*，第 108–119 页。Springer，2011。引用于 226 [ACORT11] D. Achlioptas, A. Coja-Oghlan, 和 F. Ricci-Tersenghi。随机约束满足问题的解空间几何。*Random Structures & Algorithms*，38(3):251–268，2011。引用于 144 [AD97] M. Ajtai 和 C. Dwork。具有最坏情况/平均情况等价的公钥密码系统。在 *Proceedings of the 29th annual ACM symposium on Theory of Computing*，第 284–293 页。ACM，1997。引用于 47，148 [Adl78] L. Adleman。关于随机多项式时间的两个定理。在 *Proceedings of 19th annual IEEE Symposium on Foundations of Computer Science*，第 75–83 页。IEEE，1978。引用于 71 [Adl94] L.M. Adleman。组合问题的分子计算。*Nature*，369:40，1994。引用于 244 [AEH75] E.A. Akkoyunlu，K. Ekanadham，和 R.V. Huber。网络通信设计中的某些约束和权衡。在 *ACM SIGOPS operating systems review*，第 9 卷，第 67–74 页。ACM，1975。引用于 223 [AFG14] A. Ambainis，Y. Filmus，和 F. Gall。快速矩阵乘法：激光方法的局限性。*arXiv preprint arXiv:1411.5414*，2014。引用于 128 [AFR06] J. Aspnes，F.E. Fich，和 E. Ruppert。在可靠的匿名分布式系统中，广播和共享内存之间的关系。*Distributed Computing*，18(3):209–219，2006。引用于 220 [AFT11] B. Alexeev，M.A. Forbes，和 J. Tsimerman。张量秩：一些下界和上界。在 *2011 IEEE 26th annual conference on Computational Complexity (CCC)*，第 283–291 页。IEEE，2011。引用于 133 [AGHP92] N. Alon，O. Goldreich，J. Håstad，和 R. Peralta。几乎 k-独立随机变量的简单构造。*Random Structures & Algorithms*，3(3):289–304，1992。附录：随机结构和算法 4。引用于 85 [AGKR17] S. Aaronson，C. Granade，G. Kuperberg，和 V. Russo。复杂性动物园，2017。`https://complexityzoo.uwaterloo.ca/Complexity_Zoo`。引用于 266 [AH89] K.I. Appel 和 W. Haken。*Every planar map is four colorable*，第 98 卷。美国数学学会，1989。引用于 14 [AH92] L.M. Adleman 和 M.A. Huang。*Primality testing and Abelian varieties over finite fields*，第 1512 卷。Springer，1992。引用于 136 [AH17] E. Allender 和 S. Hirahara。关于电路最小化和相关问题的（非）困难性的新见解。在 *Mathematical Foundations of Computer Science*，*LIPIcs* 第 83 卷，第 54:1–54:14 页。Schloss Dagstuhl – Leibniz-Zentrum für Informatik，2017。引用于 37，40，56

[AHK12] S. Arora, E. Hazan, 和 S. Kale. 乘性权重更新方法：一种元算法及其应用。*Theory of Computing*，8(1):121–164，2012。在95，182，235处引用[AHT06] I. Agol，J. Hass，和 W. P. Thurston. 节结 genus 和跨度面积的计算复杂性。*Transactions of the American Mathematical Society*，358:3821–3850，2006。在32处引用[Ajt83] M. Ajtai. σ_1-公式在有限结构中的应用。*Annals of the Pure and Applied Logic*，24(1):1–48，1983。在52，55处引用[Ajt96] M. Ajtai. 生成格问题的困难实例。在*Proceedings of the 28th annual ACM symposium on Theory of Computing*，第99–108页。ACM，1996。在47，148处引用[AK09] N. Alon 和 B. Klartag. 通过Cheeger不等式实现经济的托里拆利脊。*Journal of Topology and Analysis*，1(02):101–111，2009。在147处引用[AKK17] P. Austrin，P. Kaski，和 K. Kubjas。多线性映射的张量网络复杂性。*arXiv preprint arXiv:1712.09630*，2017。在119处引用[AKN98] D. Aharonov，A. Kitaev，和 N. Nisan。混合状态的量子电路。在*Proceedings of the 30th annual ACM symposium on Theory of Computing*，第20–30页。ACM，1998。在114处引用[AKS83] M. Ajtai，J. Komlós，和 E. Szemerédi。O(n log n)排序网络。在*Proceedings of the 15th annual ACM symposium on Theory of Computing*，第1–9页。ACM，1983。在55处引用[AKS87] M. Ajtai，J. Komlós，和 E. Szemerédi。LOGSPACE中的确定性模拟。在*Proceedings of the 19th annual ACM symposium on Theory of Computing*，第132–140页。ACM，1987。在101处引用[AKS04] M. Agrawal，N. Kayal，和 N. Saxena。素数属于P。*Annals of Mathematics*，160(2):781–793，2004。在19，28，80，87，136处引用[AL88] D. Angluin 和 P. Laird。从噪声示例中进行学习。*Machine Learning*，2(4):343–370，1988。在192处引用[AL11] G. Asharov 和 Y. Lindell。BGW协议对于完美多方计算的全证明。*Journal of Cryptology*，30:1–94，2011。在214处引用[Ald83] D. J. Aldous。有限群上的随机游走和快速混合马尔可夫链。在*Séminaire de Probabilités XVII 1981/82*，第243–297页。Springer，1983。在142处引用[Ald90] D. J. Aldous。均匀生成树和均匀标记树的随机游走构造。*SIAM Journal on Discrete Mathematics*，3(4):450–465，1990。在143处引用[ALGV18] N. Anari，K. Liu，S. O. Gharan，和 C. Vinzant。对数凹多项式II：高维游走和计数基的FPRAS。*arXiv preprint arXiv:1811.01816*，2018。在144处引用[All17] E. Allender。复杂性的复杂性。在*Computability and Complexity*，第79–94页。Springer，2017。在40处引用[ALM+98] S. Arora，C. Lund，R. Motwani，M. Sudan，和 M. Szegedy。证明验证和近似问题的困难性。*Journal of the ACM*，45(3):501–555，1998。在110，235处引用[ALN08] S. Arora，J. Lee，和 A. Naor。欧几里得失真和最稀疏割。*Journal of the American Mathematical Society*，21(1):1–21，2008。在140处引用

[Alo86] N. Alon. 特征值和扩张器。*Combinatorica*，6(2)：83–96，1986。在90，91处引用[ALW01] N. Alon，A. Lubotzky和A. Wigderson。群中的半直接积和图中的之字形积：联系和应用。在*Proceedings of 42nd annual IEEE Symposium on Foundations of Computer Science*，第630–637页。IEEE，2001。在92处引用[AM75] L. Adleman和K. Manders。多项式决策问题的计算复杂性。在*Proceedings of 16th annual IEEE Symposium on Foundations of Computer Science*，第169–177页。IEEE，1975。在32处引用[AM85] N. Alon和V. D. Milman。λ_1，图的测度不等式和超集中器。*Journal of Combinatorial Theory*，38(1)：73–88，1985。在90处引用[AM88] N. Alon和W. Maass。蜿蜒和它们在下界论证中的应用。*Journal of computer and system sciences*，37(2)：118–129，1988。在165处引用[AMPS13] A. Almheiri，D. Marolf，J. Polchinski和J. Sully。黑洞：互补性或防火墙?*Journal of High Energy Physics*，2013(2)：62，2013。在248处引用[AMS96] N. Alon，Y. Matias和M. Szegedy。近似频率矩的空间复杂性。在*Proceedings of the 28th annual ACM symposium on Theory of Computing*，第20–29页。ACM，1996。在157处引用[And87] A.E. Andreev。关于一种获得超过二次有效复杂度下界的方法。*Vestnik Moskovskogo Universiteta Seriya 1 Matematika Mekhanika*，(1)：70–73，1987。在52处引用[Ank52] N.C. Ankeny。最小的二次非剩余。*Annals of Mathematics*，第65–72页，1952。在87处引用[ANS10] B. Adsul，S. Nayak和K.V. Subrahmanyam。Kronecker问题的几何方法ii：矩阵的左-右同时作用的不变量。*Manuscript, available in http://www.cmi.ac.in/kv/ANS10.pdf*，18，2010。在152处引用[APR83] L.M. Adleman，C. Pomerance和R.S. Rumely。区分素数和合数。*Annals of Mathematics*，第173–206页，1983。在136处引用[AR05] D. Aharonov和O. Regev。NP ∩ coNP中的格问题。*Journal of the ACM*，52(5)：749– 765，2005。在40处引用[AR08] M. Alekhnovich和A. Razborov。除非w[p]是可处理的，否则解析不是自动化的。*SIAM Journal on Computing*，38(4)：1347–1363，2008。在68处引用[Ar 09] E. Ar kan。信道极化：一种为对称二元输入无记忆信道构建容量达到码的方法。*IEEE Transactions on Information Theory*，55(7)：3051–3073，2009。在176处引用[Aro94] S. Arora。*Probabilistic checking of proofs and the hardness of approximation problems*。博士论文，UC Berkeley，1994。修订版在`http://eccc.hpi-web.de/eccc-local/ ECCC-Books/sanjeev_book_readme.html`。在110处引用[Art27] E. Artin。关于将确定函数分解为平方的。在*Abhandlungen aus dem mathematischen Seminar der Universität Hamburg*，第5卷，第100–115页。Springer，1927。在65处引用[AS83] D. Angluin和C.H. Smith。归纳推理：理论和方法。*ACM Computing Surveys (CSUR)*，15(3)：237–269，1983。在190处引用

[AS98] S. Arora 和 S. Safra. 证明的概率检查：NP 的新特征。*Journal of the ACM*，45(1):70–122，1998。引用于 235 [AS00] N. Alon 和 J. Spencer. *The probabilistic method*。Wiley-Interscience 离散数学与优化系列。John Wiley & Sons，第 2 版，2000。引用于 55，82，103 [AS03] S. Arora 和 M. Sudan. 改进的低度测试及其应用。*Combinatorica*，23(3):365–426，2003。引用于 236 [AS10] V. Arvind 和 S. Srinivasan. 非交换行列式的难度。在 *Pro- ceedings of the 42nd annual ACM symposium on Theory of Computing*，第 677–686 页。ACM，2010。引用于 134 [Asp99] A. Aspect. 贝尔不等式测试：比以往任何时候都更理想。*Nature*，398(6724):189–190，1999。引用于 121 [Asp03] J. Aspnes. 异步一致性随机协议。*Distributed Computing*，16(2– 3):165–175，2003。引用于 224 [AV08] M. Agrawal 和 V. Vinay. 算术电路：深度四的裂谷。在 *Proceedings of 49th annual IEEE Symposium on Foundations of Computer Science*，第 67–75 页。IEEE，2008。引用于 133 [AvDK+08] D. Aharonov，W. van Dam，J. Kempe，Z. Landau，S. Lloyd，和 O. Regev。绝热量子计算与标准量子计算等价。*SIAM Review*，50(4):755–787，2008。引用于 120 [AW85] M. Ajtai 和 A. Wigderson。确定性模拟概率常数深度电路。在
Proceedings of 26th annual IEEE Symposium on Foundations of Computer Science，第 11–19 页。IEEE，1985。引用于 79 [AW04] H. Attiya 和 J. Welch.
Distributed computing: Fundamentals, simulations, and advanced topics，第 19 卷。John Wiley & Sons，2004。引用于 218，219 [AW09] S. Aaronson 和 A. Wigderson。代数化：复杂性理论中的新障碍。*ACM Transactions on Computation Theory (TOCT)*，1(1):2，2009。引用于 48，56 [AW18] J. Alman 和 V.V. Williams。所有已知（以及一些未知）矩阵乘法方法的限制。在
Proceedings of 59th annual IEEE Symposium on Foundations of Computer Science，第 580–591 页。IEEE，2018。引用于 56，128 [AZGMS14] Z. Allen-Zhu，R. Gelashvili，S. Micali，和 N. Shavit。稀疏符号一致 Johnson-Lindenstrauss 矩阵：基于神经科学约束的压缩。*Proceedings of the National Academy of Sciences*，111(47):16872–16876，2014。引用于 247 [AZH16] Z. Allen-Zhu 和 E. Hazan。方差减少以实现更快的非凸优化。在 *Inter- national conference on Machine Learning*，第 699–707 页，2016。引用于 235 [AZO14] Z. Allen-Zhu 和 L. Orecchia。线性耦合：梯度下降和镜像下降的终极统一。*arXiv preprint arXiv:1407.1537*，2014。引用于 235 [Bab85] L. Babai。用群论交换随机性。在 *Proceedings of the 17th annual ACM symposium on Theory of Computing*，计算理论研讨会 85，第 421– 429 页，纽约，NY，USA，1985。ACM。引用于 104 [Bab90] L. Babai。电子邮件和意外的互动力量。*Proceedings of the 5th annual conference on structure in complexity theory*，第 30–44 页，1990。引用于 105

[Bab91] L. Babai. 顶点传递图的局部扩张和有限群中的随机生成。在 *Proceedings of the 23rd annual ACM symposium on Theory of Computing*，第 91 卷，第 164–174 页。Citeseer，1991。在第 142 处引用 [Bab15] L. Babai. 图同构在拟多项式时间内。*arXiv preprint arXiv:1512.03547*，2015。在第 20、40、103、141 处引用 [Bar86] D.A. Barrington. 有界宽度的多项式大小分支程序识别那些语言。在 *Proceedings of the 18th annual ACM symposium on Theory of Comput- ing*，第 1–5 页。ACM，1986。在第 159、241 处引用 [Bar01] B. Barak. 如何超越黑盒模拟障碍。在 *Proceedings of 42nd annual IEEE Symposium on Foundations of Computer Science*，第 106–115 页。IEEE，2001。在第 208、260 处引用 [Bar04] B. Barak. 密码学中的非黑盒技术。*PhD Dissertation, The Weizmann Institute of Science, Computer Science department*，2004。在第 112 处引用 [Bar16] A. Barvinok. *Combinatorics and complexity of partition functions*。Springer，2016。在第 237 处引用 [BB84] C.H. Bennett 和 G. Brassard。量子密码学：公钥分配和掷币。在 *Proceedings of IEEE international conference on computers, systems and signal processing*，第 175 卷。纽约，1984。在第 118 处引用 [BBC+93] C.H. Bennett，G. Brassard，C. Crépeau，R. Jozsa，A. Peres 和 W.K. Wootters。通过双经典和爱因斯坦-波多尔斯基-罗森通道传输未知量子态。*Physical Review Letters*，70(13):1895，1993。在第 118 处引用 [BBCR13] B. Barak，M. Braverman，X. Chen 和 A. Rao。如何压缩交互式通信。*SIAM Journal on computing*，42(3):1327–1363，2013。在第 172、173、175 处引用 [BBG13] M. Balcan，A. Blum 和 A. Gupta。近似稳定性下的聚类。*Journal of the ACM*，60(2):8，2013。在第 43、259 处引用 [BC06] M. Braverman 和 S. Cook。在实数上的计算：科学计算的基础。*Notices of the AMS*，53(3):318–329，2006。在第 16 处引用 [BCC+17] J. Blasiak，T. Church，H. Cohn，J.A. Grochow 和 C. Umans。哪些群适用于证明矩阵乘法的指数为二？*arXiv preprint arXiv:1712.02302*，2017。在第 128 处引用 [BCE+95] P. Beame，S. Cook，J. Edmonds，R. Impagliazzo 和 T. Pitassi。NP 搜索问题的相对复杂性。在 *Proceedings of the 27th annual ACM symposium on Theory of Computing*，第 303–314 页。ACM，1995。在第 38 处引用 [BCHP17] S. Benoit，J. Colliard，C. Hurlin 和 C. Pérignon。风险在哪里：系统性风险综述。*Review of Finance*，21(1):109–152，2017。在第 252 处引用 [BCL+18] S. Bubeck，M. B. Cohen，Y. T. Lee，J. R. Lee 和 A. Madry。*k*-服务器通过多尺度熵正则化。在 *Proceedings of the 50th Annual ACM Symposium on Theory of Computing*，第 3–16 页。ACM，2018。在第 181 处引用 [BCS10] P. Bürgisser，M. Clausen 和 M.A. Shokrollahi。*Algebraic Complexity Theory*。Springer Publishing Company, Incorporated，2010。在第 124 处引用 [BCSS98] L. Blum，F. Cucker，M. Shub 和 S. Smale。*Complexity and Real Computation*。Springer-Verlag，1998。在第 16 处引用

[BD02] A. Blum 和 J. Dunagan. 线性规划感知器算法的平滑分析。在
Proceedings of the 13th annual ACM-SIAM symposium on discrete algorithms，第 905–914 页。SIAM，200
2。引用于 187 [BDBK+94] S. Ben-David，A. Borodin，R. Karp，G. Tardos 和 A. Wigderson。在线算法中随机
化的力量。*Algorithmica*，11(1):2–14，1994。引用于 179，181 [BDSMP91] M. Blum，A. De Santis，S. Mical
i 和 G. Persiano。非交互式零知识。*SIAM Journal on Computing*，20(6):1084–1118，1991。引用于 109 [BD
WY13] B. Barak，Z. Dvir，A. Wigderson 和 A. Yehudayoff。分数Sylvester-Gallai定理。
Proceedings of the National Academy of Sciences，110(48):19213–19219，2013。引用于 138 [Bea94] P. Bea
me。切换引理入门。技术报告，技术报告UW-CSE-95-07-01，华盛顿大学计算机科学与工程学院，1994。引
用于 52 [Bea97] R. Beals。对称群上的傅里叶变换的量子计算。在 *Proceedings*
of the 29th annual ACM symposium on Theory of Computing，第 48–53 页。ACM，1997。引用于 116 [Bec
91] J. Beck。Lovász局部引理的算法方法。i。*Random Structures & Algo- rithms*，2(4):343–365，1991。引
用于 144，238 [BEHW90] A. Blumer，A. Ehrenfeucht，D. Haussler 和 M.K. Warmuth。Occam的剃刀。
Readings in machine learning，第 201–204 页，1990。引用于 196，197 [Bel64] J.S. Bell。关于爱因斯坦-波
多尔斯基-罗森悖论。*Physics*，1(3):195–200，1964。引用于 121 [Ben80] P. Benioff。计算机作为物理系统:
用图灵机表示的计算机的微观量子力学哈密顿模型。*Journal of Statistical Physics*，22(5):563–591，1980。
引用于 114，242 [Ber67] E.R. Berlekamp。有限域上的多项式分解。*Bell System Technical Journal*，46(8):1
853–1859，1967。引用于 71，137 [Ber84] S.J. Berkowitz。使用少量处理器在小型并行时间内计算行列式。
Information processing letters，18(3):147–150，1984。引用于 129 [Bes19] A. Besicovitch。关于函数的积分
性问题。*J. Soc. Phys. Math.*，2:105– 123，1919。引用于 137 [BEY05] A. Borodin 和 R. El-Yaniv。
Online computation and competitive analysis。剑桥大学出版社，2005。引用于 179 [BF28] M. Born 和 V. Fo
ck。adiabatensatz的证明。*Zeitschrift für Physik*，51(3-4):165–180，1928。引用于 120 [BF90] D. Beaver 和 J.
Feigenbaum。在多或然查询中隐藏实例。*Annual Symposium on Theoretical Aspects of Computer Science*，
第 37–48 页，1990。引用于 106 [BFK09] A. Broadbent，J. Fitzsimons 和 E. Kashefi。通用盲量子计算。在
Proceedings of 50th annual IEEE Symposium on Foundations of Computer Science，第 517–526 页。IE
EE，2009。引用于 121 [BFK10] A. Broadbent，J. Fitzsimons 和 E. Kashefi。QMIP= MIP*。
arXiv preprint arXiv:1004.1130，2010。引用于 121

[BFL91] L. Babai, L. Fortnow, 和 C. Lund. 非确定性指数时间有两个证明者交互协议。
Computational complexity, 1(1)：3–40，1991。在106，236处引用[BFS86] L. Babai，P. Frankl，和 J. Simon。在通信复杂性理论中的复杂度类。在
*Proceedings of 27th annual IEEE Symposium on Foundations of Computer Science*中，第337–347页。IEEE，1986。在161，163处引用[BG93] E. Borowsky 和 E. Gafni。t-容错异步计算的广义FLP不可行性结果。在
*Proceedings of the 25th annual ACM symposium on Theory of Computing*中，第91–100页。ACM，1993。在225，226处引用[BG08] J. Bourgain 和 A. Gamburd。Cayley图$\mathrm{SL}_2(\mathbb{F}_p)$的均匀扩张界限。*Annals of Mathematics*，167(2)：625–642，2008。在91，92处引用[BG10] J. Bourgain 和 A. Gamburd。$SU(d)$中的谱间隙。*Comptes Rendus Mathematique*，348(11)：609–611，2010。在93处引用[BGBD+04] Y. Benenson，B. Gil，U. Ben-Dor，R. Adar，和 E. Shapiro。一个用于逻辑控制基因表达的自主分子计算机。*Nature*，429(6990)：423–429，2004。在244处引用[BGGT13] E. Breuillard，B. Green，R. Guralnick，和 T. Tao。有限简单李型群的扩张。*arXiv preprint arXiv:1309.1975*，2013。在93处引用[BGI+01] B. Barak，O. Goldreich，R. Impagliazzo，S. Rudich，A. Sahai，S. Vadhan，和 K. Yang。关于程序混淆的（不可）可能性。在
*Annual International Cryptology Conference*中，第1–18页。Springer，2001。在216处引用[BGS75] T. Baker，J. Gill，和 R. Solovay。P =? NP问题的相对化。*SIAM Journal on Computing*，4：431–442，1975。在48处引用[BGS10] J. Bourgain，A. Gamburd，和 P. Sarnak。仿射线性筛、扩张器和乘积和。
Inventiones mathematicae，179(3)：559–644，2010。在93处引用[BGT10] E. Breuillard，B. Green，和 T. Tao。Suzuki群作为扩张器。*arXiv preprint arXiv:1005.0782*，2010。在92处引用[BGW99] L. Babai，A. Gál，和 A. Wigderson。单调跨度程序的超级多项式下界。*Combinatorica*，19(3)：301–319，1999。在85处引用[BH14] S. Bravyi 和 M. Hastings。量子Ising模型的复杂性。*arXiv preprint arXiv:1410.0703*，2014。在119处引用[BIP16] P. Bürgisser，C. Ikenmeyer，和 G. Panova。几何复杂性理论中没有发生障碍。
arXiv preprint arXiv:1604.06431，2016。在131，151处引用[BIW06] B. Barak，R. Impagliazzo，和 A. Wigderson。使用少量独立源提取随机性。*SIAM Journal on computing*，36(4)：1095–1118，2006。在102，138处引用[BJ11] L. Brouwer 和 Egbertus J. Über abbildung von mannigfaltigkeiten。*Mathematische Annalen*，71(1)：97–115，1911。在227处引用[BJ01] A.A. Bulatov 和 P. Jeavons。组合问题中的代数结构。*International Journal of Algebra and Computing*，2001。在41处引用[BJKS04] Z. Bar-Yossef，T.S. Jayram，R. Kumar，和 D Sivakumar。数据流和通信复杂性的信息统计方法。*Journal of computer and system sciences*，68(4)：702–732，2004。在163，172处引用

[BJSW16] A. Broadbent, Z. Ji, F. Song, and J. Watrous. QMA的零知识证明系统。在 *Proceedings of 57th annual IEEE Symposium on Foundations of Computer Science*，第 31–40 页。IEEE，2016。引用于 121 [BK89] M. Blum 和 S. Kannan。设计检查自身工作的程序。在 *Proceedings of the 21st annual ACM symposium on Theory of Computing*，第 86–97 页。ACM，1989。引用于 106 [BKI+96] P. Beame，J. Krajicek，R. Impagliazzo，T. Pitassi 和 P. Pudlak。希尔伯特零点定理和命题证明的下界。*Proceedings of the London Math Society*，73(3):1–26，1996。引用于 62 [BKN14] Z. Brakerski，Y.T. Kalai 和 M. Naor。对抗噪声的快速交互式编码。*Journal of the ACM*，61(6):35，2014。引用于 177 [BKPS02] P. Beame，R. Karp，T. Pitassi 和 M. Saks。归结和 Davis–Putnam 程序的效率。*SIAM Journal on computing*，31(4):1048–1075，2002。引用于 68 [BKS99] I. Benjamini，G. Kalai 和 O. Schramm。布尔函数的噪声敏感性和在渗透中的应用。*Publications Mathématiques de l'Institut des Hautes Études Scientifiques*，90(1):5–43，1999。引用于 145 [BKS+05] B. Barak，G. Kindler，R. Shaltiel，B. Sudakov 和 A. Wigderson。模拟独立性：新的冷凝器、拉姆齐图、分散器和提取器的构造。在 *Proceedings of the 37th annual ACM symposium on Theory of Computing*，第 1–10 页。ACM，2005。引用于 84 [BKT04] J. Bourgain，N. Katz 和 T. Tao。有限域中的和积估计及其应用。*Geometric & Functional Analysis GAFA*，14(1):27–57，2004。引用于 93，138 [BKW17] L. Barto，A. Krokhin 和 R. Willard。多态性和如何使用它们。在 *Dagstuhl Follow-Ups*，第 7 卷。Schloss Dagstuhl – Leibniz-Zentrum für Informatik，2017。引用于 41 [BL97] A. L. Blum 和 P. Langley。在机器学习中选择相关特征和示例。*Artificial intelligence*，97(1-2):245–271，1997。引用于 200 [BL06] Y. Bilu 和 N. Linial。提升、差异和近似谱间隙。*Combinatorica*，26(5):495–519，2006。引用于 93，139 [BL08] P. A. Brooksbank 和 E. M. Luks。测试模块的同构。*Journal of Algebra*，320(11):4020–4029，2008。引用于 152 [BL12] Y. Bilu 和 N. Linial。稳定的实例是否容易？*Combinatorics, Probability and Computing*，21(05):643–660，2012。引用于 43，259 [BLG12] H. Bäärnhielm 和 C.R. Leedham-Green。产品替换勘探者。*Journal of Symbolic Computation*，47(1):64–75，2012。引用于 142 [BLMW11] P. Bürgisser，J.M. Landsberg，L. Manivel 和 J. Weyman。几何复杂性理论方法中出现的数学问题的概述。*SIAM Journal on computing*，40(4):1179–1209，2011。引用于 131 [BLNPL14] M. Babaioff，B. Lucier，N. Nisan 和 R. Paes Leme。关于瓦尔拉斯机制的效率。在 *Proceedings of the 15th ACM conference on Economics and computation*，第 783–800 页。ACM，2014。引用于 251 [BLR93] M. Blum，M. Luby 和 R. Rubinfeld。自测试/自纠正及其在数值问题中的应用。*Journal of computer and system sciences*，47(3):549–595，1993。引用于 106，236

[Blu86] M. Blum. 从相关偏置源获取独立无偏硬币抛掷——有限状态马尔可夫链。*Combinatorica*，6(2):97–108，1986。在98 [BM84] M. Blum 和 S. Micali。如何生成密码学安全的伪随机比特序列。*SIAM Journal on Computing*，13:850–864，1984。在77，207 [BO83] M. Ben-Or。自由选择的另一个优点：完全异步的协议。在*Proceedings of the second annual ACM symposium on principles of distributed computing*，第27–30页。ACM，1983。在224 [BO85] M. Ben-Or。私人通信，1985。在126 [BOC92] M. Ben-Or 和 R. Cleve。使用固定数量的寄存器计算代数公式。*SIAM Journal on computing*，21(1):54–58，1992。在160 [BOGKW89] M. Ben-Or，S. Goldwasser，J. Kilian 和 A. Wigderson。使用两个证明者交互证明的有效身份识别方案。*Advances in Cryptography (CRYPTO 89), Lecture Notes in Computing*，435:498–506，1989。在105，106，109，122 [BOGW88] M. Ben-Or，S. Goldwasser 和 A. Wigderson。非密码学容错分布式计算的完备性定理。在*Proceedings of the 20th annual ACM symposium on Theory of Computing*，第1–10页。ACM，1988。在214，251 [BOL85] M. Ben-Or 和 N. Linial。集体抛硬币，鲁棒的投票方案和 Banzhaf 值的最小值。在*Proceedings of 26th annual IEEE Symposium on Foundations of Computer Science*，第408–416页。IEEE，1985。在145 [Bor85] C. Borell。Ornstein-Uhlenbeck 速度过程的几何界限。*Zeitschrift für Wahrscheinlichkeitstheorie und Verwandte Gebiete*，70(1):1–13，1985。在146 [Bou85] J. Bourgain。有限度量空间在 Hilbert 空间中的 Lipschitz 嵌入。*Israel Journal of Mathematics*，52(1-2):46–52，1985。在139 [BP98] P. Beame 和 T. Pitassi。命题证明复杂性的过去、现在和未来。*Bulletin EATCS*，65:66–89，1998。在57 [BPR97] M. Bonet，T. Pitassi 和 R. Raz。具有小系数的切割平面证明的下界。*The Journal of Symbolic Logic*，62(03):708–728，1997。在68 [BPR15] N. Bitansky，O. Paneth 和 A. Rosen。关于找到纳什均衡的密码学难度。在*Proceedings of 56th annual IEEE Symposium on Foundations of Computer Science*，第1480–1498页。IEEE，2015。在250 [BR14] M. Braverman 和 A. Rao。信息等于摊销通信。*IEEE Transactions on Information Theory*，60(10):6058–6069，2014。在173，174，175 [Bra84] G. Bracha。异步$[(n-1)/3]$-容错共识协议。在*Proceedings of the third annual ACM symposium on principles of distributed computing*，第154–162页。ACM，1984。在224 [Bra15] M. Braverman。交互信息复杂性。*SIAM Journal on computing*，44(6):1698– 1739，2015。在175 [Bra18] Z. Brakerski。全同态加密的基本原理——综述。*Electronic Colloquium on Computational Complexity (ECCC)*，25:125，2018。在149 [Bro89] A. Broder。生成随机生成树。在*Proceedings of 30th annual IEEE Symposium on Foundations of Computer Science*，第442–447页。IEEE，1989。在143

[Bro15] A. Broadbent. 委托私人量子计算。*Canadian Journal of Physics*，93(999)：1–6，2015。在121 [BRS11] M. Biely，P. Robinson，和U. Schmid。在消息传递系统中k集一致性的简单不可行性证明。在*International conference On Principles Of Distributed Systems*，第299–312页。Springer，2011。在225 [BRSW12] B. Barak，A. Rao，R. Shaltiel，和A. Wigderson。$n^{o(1)}$熵的2源分散器和击败Frankl-Wilson构造的Ramsey图。*Annals of Mathematics*，176(3)：1483–1543，2012。在84 [BRW05] R.D. Barish，P.W.K. Rothemund，和E. Winfree。算法自组装的两个计算原语：复制和计数。*Nano letters*，5(12)：2586–2592，2005。在244 [BS83] W. Baur和V. Strassen。偏导数的复杂性。*Theoretical Computer Science*，22(3)：317–330，1983。在126，127 [BS84] L. Babai和E. Szemerédi。关于矩阵群问题的复杂性I。在*Proceedings of 25th annual IEEE Symposium on Foundations of Computer Science*，第229–240页。IEEE，1984。在22，141 [BS97] E. Bach和J. Shallit。*Algorithmic Number Theory: Efficient Algorithms*，第1卷。MIT Press，剑桥，1997。在137 [BS14] B. Barak和D. Steurer。和平方证明以及寻找最优算法的探索。在*Proceedings of the International Congress of Mathematicians (ICM)*，2014。在65，235 [BSBC+17] E. Ben-Sasson，I. Bentov，A. Chiesa，A. Gabizon，D. Genkin，M. Hamilis，E. Pergament，M. Ri- abzev，M. Silberstein，E. Tromer，等。使用准线性pcps的公共随机字符串进行计算完整性。在*Annual International conference on the Theory and Applications of Cryptographic Techniques*，第551–579页。Springer，2017。在113 [BSCTV17] E. Ben-Sasson，A. Chiesa，E. Tromer，和M. Virza。通过椭圆曲线循环的可扩展零知识证明。*Algorithmica*，79(4)：1102–1160，2017。在112 [BSS14] J. Batson，D. A. Spielman，和N. Srivastava。两次拉马努金稀疏器。*SIAM Review*，56(2)：315–334，2014。在138 [BSSV03] P. Beame，M. Saks，X. Sun，和E. Vee。随机计算决策问题的时空权衡下界。*Journal of the ACM*，50(2)：154–195，2003。在156 [BSW99] E. Ben-Sasson和A. Wigderson。简短证明是狭窄的——简化了解决方案。在*Proceedings of the 31st annual ACM symposium on Theory of Computing*，第517–526页。ACM，1999。在67 [BSZ13] J. Bourgain，P. Sarnak，和T. Ziegler。莫比乌斯与测地线流的分离。在*From Fourier analysis and number theory to Radon transforms and geometry*，第67–83页。Springer，2013。在85 [BT91] J. Bourgain和L. Tzafriri。关于Kadison和Singer的问题。*Journal Fur Die Reine Und Angewandte Mathematik*，420：1–43，1991。在138 [Bul06] A.A. Bulatov。在3元素集上的约束满足问题的二分定理。*Journal of the ACM*，53(1)：66–120，2006。在41 [Bul17] A.A. Bulatov。非均匀CSP的二分定理。*arXiv preprint arXiv:1703.03021*，2017。在41

[Bus87] S. Buss. 命题抽屉原理的多项式大小证明。*Journal of Symbolic Logic*，52:916–927，1987。在66 [BV97] E. Bernstein和U. Vazirani。量子复杂度理论。*SIAM Journal on computing*，26(5):1411–1473，1997。在114 [BV14] Z. Brakerski和V. Vaikuntanathan。从（标准）LWE的高效全同态加密。
SIAM Journal on computing，43(2):831–871，2014。在215 [BY13] J. Bourgain和A. Yehudayoff。在$SL_2(\mathbb{R})$中的扩展和单调扩展器。*Geometric and Functional Analysis*，23(1):1–41，2013。在93 [Can01] R. Canetti。普遍可组合安全性：密码协议的新范式。在
Proceedings of 42nd annual IEEE Symposium on Foundations of Computer Science，第136–145页。IEEE，2001。在213 [Cay45] A. Cayley。*On the theory of linear transformations*。E. Johnson，1845。在149 [CC12] J. Cai和X. Chen。具有复杂权重的计数CSP的复杂性。在*Proceedings of the
44th annual ACM symposium on Theory of Computing*，第909–920页。ACM，2012。在42 [CC17] J. Cai和X. Chen。*Complexity Dichotomies for Counting Problems: Volume 1, Boolean Domain*。剑桥大学出版社，2017。在42，237 [CCD88] D. Chaum，C. Crépeau，和I. Damgård。无条件安全的多方协议。在
Proceedings of the 20th annual ACM symposium on Theory of Computing，第11–19页。ACM，1988。在214 [CCD+03] A.M. Childs，R. Cleve，E. Deotto，E. Farhi，S. Gutmann，和D.A. Spielman。量子行走指数算法加速。在*Proceedings of the 35th annual ACM symposium on Theory of Computing*，第59–68页。ACM，2003。在116 [CCL10] J. Cai，X. Chen，和D. Li。在任何特征下永真式与行列式的二次下界。
Computational Complexity，19(1):37–56，2010。在131 [CDD+99] R. Cramer，I. Damgård，S. Dziembowski，M. Hirt，和T. Rabin。有效多方计算安全对抗自适应对手。在*International conference on the Theory and Applications of Cryptographic Techniques*，第311–326页。Springer，1999。在213 [CDNO97] R. Canetti，C. Dwork，M. Naor，和R. Ostrovsky。可否认加密。在*Annual Interna- tional Cryptology Conference*，第90–104页。Springer，1997。在207 [CDT09] X. Chen，X. Deng，和S. Teng。解决计算两人纳什均衡的复杂性。
Journal of the ACM，56(3):14，2009。在38，250 [CEI96] M. Clegg，J. Edmonds，和R. Impagliazzo。使用Groebner基算法寻找不可满足性的证明。在
Proceedings of the 28th annual ACM symposium on Theory of Computing，第174–183页。ACM，1996。在62 [CG88] B. Chor和O. Goldreich。从弱随机源得到无偏比特和概率通信复杂性。
SIAM Journal on computing，17(2):230–261，1988。在98，102 [CGH+15] J. Coron，C. Gentry，S. Halevi，T. Lepoint，H.K. Maji，E. Miles，M. Raykova，A. Sahai，和M. Tibouchi。无需低级零点的零化：新的MMAP攻击及其局限性。在*Annual Cryptology Conference*，第247–266页。Springer，2015。在216

[CGMA85] B. Chor, S. Goldwasser, S. Micali, 和 B. Awerbuch. 可验证的秘密共享和在故障存在的情况下实现同时性。在 *Proceedings of 26th annual IEEE Symposium on Foundations of Computer Science*，第 383–395 页。IEEE，1985。引用于 213 [CGW89] F.R.K. Chung, R.L. Graham, 和 R.M. Wilson. 准随机图。*Combinatorica*，9(4):345–362，1989。引用于 89 [Cha93] S. Chaudhuri. 更多选择允许更多故障：在完全异步系统中的集合共识问题。*Information and Computation*，105(1):132–158，1993。引用于 225，227 [Cha12] B. Chazelle. 自然算法和影响系统。*Communications of the ACM*，55(12):101–110，2012。引用于 246，252 [Cha15] B. Chazelle. 集体行为的算法方法。*Journal of Statistical Physics*，158(3):514–548，2015。引用于 246，252 [Che70] J. Cheeger. 拉普拉斯算子最小特征值的下界。*Problems in Analysis*，625:195–199，1970。引用于 90 [CHHL18] E. Chattopadhyay, P. Hatami, K. Hosseini, 和 S. Lovett. 从极化随机游走生成伪随机生成器。在 *LIPIcs-Leibniz International Proceedings in Informatics*，第 102 卷。Schloss Dagstuhl – Leibniz-Zentrum für Informatik，2018。引用于 79 [CHPW98] A. Condon, L. Hellerstein, S. Pottle, 和 A. Wigderson. 关于具有非确定性状态和概率状态的有限自动机的力量。*SIAM Journal on computing*，27(3):739–762，1998。引用于 160 [CHSH69] J.F. Clauser, M.A. Horne, A. Shimony, 和 R.A. Holt. 提出测试局域隐变量理论的实验。*Physical review letters*，23(15):880，1969。引用于 121，122 [Chv73] V. Chvátal. Edmonds 多面体和组合问题层次。*Discrete Mathe- maics*，4:305–337，1973。引用于 63 [CIKK16] M.L. Carmosino, R. Impagliazzo, V. Kabanets, 和 A. Kolokolova. 从自然证明学习算法。在 *LIPIcs-Leibniz International Proceedings in Informatics*，第 50 卷。Schloss Dagstuhl – Leibniz-Zentrum für Informatik，2016。引用于 80 [CJW06] J.A. Carlson, A. Jaffe, 和 A. Wiles. *The Millennium Prize Problems*。美国数学学会，2006。引用于 24 [CKGS98] B. Chor, E. Kushilevitz, O. Goldreich, 和 M. Sudan. 私有信息检索。*Journal of the ACM*，45(6):965–981，1998。引用于 112，215 [CKL13] B.I. Carlin, S. Kogan, 和 R. Lowery. 交易复杂资产。*The Journal of Finance*，68(5):1937–1960，2013。引用于 252 [CKM$^+$11] P. Christiano, J.A. Kelner, A. Madry, D.A. Spielman, 和 S. Teng. 电气流，拉普拉斯系统，以及无向图中最大流的更快近似。在 *Proceedings of the 43rd annual ACM symposium on Theory of Computing*，第 273–282 页。ACM，2011。引用于 235 [CKSU05] H. Cohn, R. Kleinberg, B. Szegedy, 和 C. Umans. 群论算法用于矩阵乘法。在 *Proceedings of 46th annual IEEE Symposium on Foundations of Computer Science*，第 379–388 页。IEEE，2005。引用于 128 [CKW11] X. Chen, N. Kayal, 和 A. Wigderson. *Partial derivatives in arithmetic complexity and beyond*。Now Publishers Inc.，2011。引用于 124，134

[CL16] E. Chattopadhyay 和 X. Li. Sumset 源的提取器。在 *Proceedings of the 48th annual ACM symposium on Theory of Computing*，第 299–311 页。ACM，2016。引用于 102 [CLO92] D. Cox，J. Little 和 D. O’Shea。*Ideals, Varieties, and Algorithms: an introduction to computational algebraic geometry and commutative algebra*。数学本科教材。Springer-Verlag，纽约，1992。引用于 149 [CLPV13] E. Chastain，A. Livnat，C. Papadimitriou 和 U. Vazirani。协调博弈中的乘性更新和进化理论。在 *Proceedings of the 4th conference on innovations in theoretical Computer Science*，第 57–58 页。ACM，2013。引用于 182 [CLPV14] E. Chastain，A. Livnat，C. Papadimitriou 和 U. Vazirani。算法、博弈和进化。*Proceedings of the National Academy of Sciences*，111(29)：10620–10623，2014。引用于 36，246 [CLR01] T.H. Cormen，C. Leiserson 和 R. Rivest。*Introduction to Algorithms*。MIT 压力，剑桥，MA，麦格劳-希尔图书公司，纽约，2001。引用于 18 [CLRS16] S.O Chan，J.R. Lee，P. Raghavendra 和 D. Steurer。近似约束满足需要大的 LP 松弛。*Journal of the ACM*，63(4)：34，2016。引用于 235 [CM84] K.M. Chandy 和 J. Misra。饮酒哲学家问题。*ACM Transactions on Pro- gramming Languages and Systems*，6(4)：632–646，1984。引用于 221 [CM13] T. Cubitt 和 A. Montanaro。局部哈密顿问题的复杂度分类。*arXiv preprint arXiv:1311.3161*，2013。引用于 119 [Cob65] A. Cobham。函数的内禀计算难度。*Logic, Methodology, and Philosophy of Science: Proceedings of the 1964 International Congress*，第 24–30 页，1965。引用于 17 [Coh16] M.B. Cohen。多项式时间内拉马努金图。*arXiv preprint arXiv:1604.03544*，2016。引用于 93 [Coh17] G. Cohen。向最优双源提取器和拉姆齐图迈进。在 *Proceedings of the 49th annual ACM symposium on Theory of Computing*，第 1157–1170 页。ACM，2017。引用于 84 [Col06] Roger A. Colbeck。*Quantum and relativistic protocols for secure multi-party computation*。博士论文，剑桥大学三一学院，2006。引用于 123 [Con92] A. Condon。随机博弈的复杂性。*Information and Computation*，96(2)：203–224，1992。引用于 28 [Con93] A. Condon。关于简单随机博弈的算法。*Advances in computational complexity theory*，13：51–73，1993。引用于 40 [Coo71] S.A. Cook。定理证明过程的复杂性。在 *Proceedings of the 3rd annual ACM symposium on Theory of Computing*，第 151–158 页。ACM，1971。引用于 22，30，31 [Coo02] G. Cooperman。向有限群中随机生成的一个实用且理论上有根据的算法迈进。*arXiv preprint arXiv:0205203*，2002。引用于 142 [Cor11] Graham Cormode。近似查询处理中的草图技术。*Foundations and Trends in Databases*，2011。引用于 156 [Cov69] R.R. Coveyou。随机数生成太重要了，不能留给机会。*Studies in applied mathematics*，3：70–111，1969。引用于 77

[Cov91] T.M. Cover. 通用投资组合。*Mathematical finance*，1(1)：1–29，1991。引用于184 [CR79] S.A. Cook和R.A. Reckhow。命题证明系统的相对效率。*Journal of Symbolic Logic*，44：36–50，1979。引用于60，61，66 [CRVW02] M. Capalbo，O. Reingold，S. Vadhan和A. Wigderson。随机性导体和常度无损展开器。在*Proceedings of the 34th annual ACM symposium on Theory of Computing*，第155卷，第659–668页。ACM，2002。引用于92，236 [CS88] V. Chvátal和E. Szemerédi。对Resolution的许多困难示例。*Journal of the ACM*，35(4)：759–768，1988。引用于67 [CS07] A. Czumaj和C. Sohler。通过随机抽样的聚类子线性时间近似算法。*Random Structures & Algorithms*，30(1-2)：226–256，2007。引用于112 [CSWY01] A. Chakrabarti，Y. Shi，A. Wirth和A. Yao。信息复杂性和直接和问题对于同时消息复杂度。在*Proceedings of 42nd annual IEEE Symposium on Foundations of Computer Science*，第270–278页。IEEE，2001。引用于172，173 [CU03] H. Cohn和C. Umans。快速矩阵乘法的群论方法。在*Proceedings of 44th annual IEEE Symposium on Foundations of Computer Science*，第438–449页。IEEE，2003。引用于128 [CU13] H. Cohn和C. Umans。使用一致配置的快速矩阵乘法。在*Proceedings of the 24th annual ACM-SIAM symposium on discrete algorithms*，第1074–1086页。SIAM，2013。引用于128 [CV86] R. Cole和U. Vishkin。确定性掷硬币及其在最优并行列表排序中的应用。*Information and Control*，70(1)：32–53，1986。引用于231 [CvD10] A.M. Childs和W. van Dam。代数问题的量子算法。*Reviews of Modern Physics*，82(1)：1，2010。引用于116 [CVZ18] M. Christandl，P. Vrana和J. Zuiddam。从不可逆性快速矩阵乘法的障碍。*arXiv preprint arXiv:1812.06952*，2018。引用于128 [CW89] A. Cohen和A. Wigderson。分散器，确定性放大和弱随机源。在*Proceedings of 30th annual IEEE Symposium on Foundations of Computer Science*，第14–19页。IEEE，1989。引用于101 [CY14] M. Coudron和H. Yuen。使用固定数量设备的无限随机扩展。在*Proceedings of the 46th annual ACM symposium on Theory of Computing*，第427–436页。ACM，2014。引用于123 [CZ12] J.I. Cirac和P. Zoller。量子模拟的目标和机遇。*Nature Physics*，8(4)：264–266，2012。引用于119 [CZ16] E. Chattopadhyay和D. Zuckerman。显式双源提取器和鲁棒函数。在*Proceedings of the 48th annual ACM symposium on Theory of Computing*，第670–683页。ACM，2016。引用于84，102 [DA01] P. Dayan和L. F. Abbott。*Theoretical neuroscience*，第806卷。MIT Press，2001。引用于247 [Dav71] R.O. Davies。关于Kakeya问题的几点评论。在*Mathematical proceedings of the Cambridge Philosophical Society*，第69卷，第417–421页。剑桥大学出版社，1971。引用于137

[DDMO04] E.D. Demaine, S.L. Devadoss, J.S.B. Mitchell, 和 J. O'Rourke. 多边形纸张的连续可折叠性。在 *CCCG*，第 64–67 页，2004 年。引用于 245 [DDN03] D. Dolev，C. Dwork，和 M. Naor。不可篡改密码学。*SIAM review*，45(4):727–784，2003 年。引用于 207 [DdW09] A. Drucker 和 R. de Wolf。经典定理的量子证明。*arXiv preprint arXiv:0910.3376*，2009 年。引用于 122 [Del74] P. Deligne。韦伊猜想。I。*Publications Mathématiques de l'Institut des HautesÉtudes Scientifiques*，43(1):273–307，1974 年。引用于 85 [Del80] P. Deligne。韦伊猜想。II。*Publications mathématiques de l'IHÉS*，52(1):137–252，1980 年。引用于 85 [Der17] Z. Derakhshandeh。*Algorithmic Foundations of Self-Organizing Programmable Matter*。博士论文，亚利桑那州立大学，2017 年。引用于 244 [Deu85] D. Deutsch。量子理论，图灵-丘奇原理和通用量子计算机。在 *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*，第 400 卷，第 97–117 页。英国皇家学会，1985 年。引用于 114，242 [DFH+15] C. Dwork，V. Feldman，M. Hardt，T. Pitassi，O. Reingold，和 A.L. Roth。在自适应数据分析中保持统计有效性。在 *Proceedings of the 47th annual ACM symposium on Theory of Computing*，第 117–126 页。ACM，2015 年。引用于 254 [DFK91] M. Dyer，A. Frieze，和 R. Kannan。逼近凸体体积的随机多项式时间算法。*Journal of the ACM*，38(1):1–17，1991 年。引用于 72，143，144 [DFP04] M. Düflot，L. Fribourg，和 C. Picaronny。无公平性假设的随机哲学家就餐问题。*Distributed Computing*，17(1):65–76，2004 年。引用于 222 [DG16] Z. Dvir 和 S. Gopi。具有亚多项式通信的 2 服务器 PIR。*Journal of the ACM*，63(4):39，2016 年。引用于 112，236 [DGP09] C. Daskalakis，P.W. Goldberg，和 C.H. Papadimitriou。计算纳什均衡的复杂性。*SIAM Journal on computing*，39(1):195–259，2009 年。引用于 38，250 [DGW09] Z. Dvir，A. Gabizon，和 A. Wigderson。多项式源提取器和秩提取器。*Computational Complexity*，18(1):1–58，2009 年。引用于 102 [DH76] W. Diffie 和 M. Hellman。密码学的新方向。*IEEE Transactions Information Theory*，22:644–654，1976 年。引用于 45，46，203，206 [DHK+19] I. Dinur，P. Harsha，T. Kaufman，I. Livni，和 A. Ta-Shma。双样本列表解码。在 *Proceedings of the Thirtieth annual ACM-SIAM Symposium on Discrete Algorithms, 2019*，第 2134–2153 页，2019 年。引用于 93 [Dia88] P. Diaconis。概率论和统计学中的群表示。*Lecture Notes-Monograph Series*，11:i–192，1988 年。引用于 142 [Dij65] E.W. Dijkstra。并发程序控制中的问题解决方案。*Communications of the ACM*，8(9):569，1965 年。引用于 220 [Dij71] E.W. Dijkstra。顺序进程的分层排序。*Acta informatica*，1(2):115–138，1971 年。引用于 220

[Din07] Dinur I. 通过间隙放大的PCP定理。*Journal of the ACM*, 54(12), 2007. 引用于92, 110 [Dix08] J.D. Dixon. 在有限群中生成随机元素。*The electronic journal of combina- torics*, 13(R94):1, 2008. 引用于142 [DK15] H. Derksen 和 G. Kemper. *Computational invariant theory*. Springer, 2015. 引用于149 [DK16] V. Danos 和 H. Koeppl. 计算机科学和生物学中的自组装和自组织（Dagstuhl研讨会15402）。在 *Dagstuhl Reports*，第5卷，第125–138页。Schloss Dagstuhl – Leibniz-Zentrum für Informatik，2016。引用于244 [DK17] I. Dinur 和 T. Kaufman. 高维扩张器意味着协议扩张器。*Electronic Colloquium on Computational Complexity (ECCC)*, 24:89, 2017. 引用于93, 236 [DKSS13] Z. Dvir, S. Kopparty, S. Saraf 和 M. Sudan. 多重性方法的扩展及其在Kakeya集和合并中的应用。*SIAM Journal on computing*, 42(6):2305–2328, 2013. 引用于137 [DL78] R.A. DeMillo 和 R.J. Lipton. 关于代数程序测试的概率性评论。*Informa- tion Processing Letters*, 7(4):193–195, 1978. 引用于70 [DLL62] M. Davis, G. Logemann 和 D. Loveland. 用于定理证明的机器程序。*Journal of the ACM*, 5(7):394–397, 1962. 引用于67 [DM15] H. Derksen 和 V. Makam. 矩阵半不变量的多项式度数界限。*arXiv preprint arXiv:1512.03393*, 2015. 引用于151, 152 [DM16] I. Dinur 和 O. Meir. 向KRW组合猜想迈进：通过通信复杂性得到三次公式下界。在 *Proceedings of the 31st conference on computational com- plexity*，第3页。Schloss Dagstuhl – Leibniz-Zentrum für Informatik，2016。引用于53 [DO08] E.D. Demaine 和 J. O’Rourke. *Geometric folding algorithms: linkages, origami, polyhedra*. 剑桥大学出版社，2008。引用于245 [Don92] S. Donkin. 几个矩阵的不变量。*Inventiones mathematicae*, 110(1):389–401, 1992。引用于151 [DR14] C. Dwork 和 A. Roth. 差分隐私的算法基础。*Foundations and Trends in Theoretical Computer Science*, 9(3-4), 2014。引用于202 [DR16] D. Dadush 和 O. Regev. 向强反向Minkowski型不等式迈进。在 *Proceedings of 57th annual IEEE Symposium on Foundations of Computer Science*，第447–456页。IEEE，2016。引用于147 [DS90] C. Dwork 和 L. Stockmeyer. 双向概率有限状态自动机的时复杂度间隙。*SIAM Journal on computing*, 19(6):1011–1023, 1990。引用于160 [DS92] C. Dwork 和 L. Stockmeyer. 有限状态验证器I：交互作用的力量。*Journal of the ACM*, 39(4):800–828, 1992。引用于160 [DS07] Z. Dvir 和 A. Shpilka. 具有两个查询的局部可解码码和深度3电路的多项式身份测试。*SIAM Journal on computing*, 36(5):1404–1434, 2007。引用于138 [DS13] A. De 和 R. Servedio. 低度多项式阈值函数的有效确定性近似计数。*arXiv preprint arXiv:1311.7178*, 2013。引用于95

[DSS14] A. Daniely 和 S. Shalev-Shwartz. 在学习 DNF 上的复杂性理论限制。*CoRR, abs/1404.3378*，1(2.1)：2–1，2014。被引用于 200 [DSTW14] I. Diakonikolas，R.A. Servedio，L. Tan 和 A. Wan。低度多项式阈值函数的正则性引理和低权重逼近器。*Theory of Computing*，10(2)：27–53，2014。被引用于 96 [DSVW04] M. Dyer，A. Sinclair，E. Vigoda 和 D. Weitz。晶格自旋系统的时空混合：组合观点。
Random Structures & Algorithms，24(4)：461–479，2004。被引用于 144，238 [DSW14a] Z. Dvir，S. Saraf 和 A. Wigderson。在实数上的 3-LCC 的二次屏障被打破。在
Proceedings of the 46th annual ACM symposium on Theory of Computing 中，第 784–793 页。ACM，2014。被引用于 112 [DSW14b] Z. Dvir，S. Saraf 和 A. Wigderson。设计矩阵的改进秩界限和 Kelly 定理的新证明。在 *Forum of Mathematics, Sigma* 中，第 2 卷。剑桥大学出版社，2014。被引用于 112，236 [Dvi09] Z. Dvir。有限域中 Kakeya 集的大小。*Journal of the American Mathematical Society*，22(4)：1093–1097，2009。被引用于 101，137 [Dvi10] Z. Dvir。专栏：从随机提取到旋转针。*ACM SIGACT News*，40(4)：46–61，2010。被引用于 137 [DW00] H. Derksen 和 J. Weyman。箭头的半不变量和 Littlewood-Richardson 系数的饱和度。*Journal of the American Mathematical Society*，13(3)：467–479，2000。被引用于 152 [DW06] H. Derksen 和 J. Weyman。箭头表示的组合。*arXiv preprint math/0608288*，2006。被引用于 151 [DW11] Z. Dvir 和 A. Wigderson。Kakeya 集集，新的合并和旧的提取器。*SIAM Journal on computing*，40(3)：778–792，2011。被引用于 101 [DZ01] M. Domokos 和 A.N. Zubkov。箭头的半不变量作为行列式。*Transformation groups*，6(1)：9–24，2001。被引用于 152 [Edm65a] J. Edmonds。最大匹配和一个具有 0，l-顶点的多面体。
Jornal of Research of the National Bureau of Standards B，69(1965)：125–130，1965。被引用于 169 [Edm65b] J. Edmonds。路径、树和花。*Canadian Journal of Mathematics*，17(3)：449–467，1965。被引用于 17，19，169 [Edm66] J. Edmonds。将拟阵划分为独立集的最小分割。*Journal of Research of the National Bureau of Standards*，B(69)：67–72，1966。被引用于 17，23，26，27 [Edm67a] J. Edmonds。最佳分支。*Journal of Research of the National Bureau of Standards B*，71：233，1967。被引用于 17，23 [Edm67b] J. Edmonds。不同代表系统和线性代数。*J. Res. Nat. Bur. Standards Sect. B*，71：241–245，1967。被引用于 132 [Efr12] K. Efremenko。子指数长度的 3 查询局部可解码码。*SIAM Journal on computing*，41(6)：1694–1703，2012。被引用于 236 [EGL85] S. Even，O. Goldreich 和 A. Lempel。签订合同的随机协议。*Commu- nications of the ACM*，28(6)：637–647，1985。被引用于 213

[EGO84] A. El Gamal和A. Orlitsky. 交互式数据压缩。在*Proceedings of 25th annual IEEE Symposium on Foundations of Computer Science*，第100–108页。IEEE，1984。引用于174 [EGOW17] K. Efremenko，A. Garg，R. Oliveira和A. Wigderson。算术复杂度中秩方法的障碍。*arXiv preprint arXiv:1710.09502*，2017。引用于132，133 [EGSZ16] F. Ellen，R. Gelashvili，N. Shavit和L. Zhu。多处理器同步的基于复杂度的层次结构。*arXiv preprint arXiv:1607.06139*，2016。引用于219 [EHKS14] K. Eisenträger，S. Hallgren，A. Kitaev和F. Song。计算任意度数数域单位群的量子算法。在*Proceedings of the 46th annual ACM symposium on Theory of Computing*，第293–302页。ACM，2014。引用于116 [EK10] S. Easley和J. Kleinberg。*Networks, crowds, and markets: Reasoning about a highly connected world*。剑桥大学出版社，2010。引用于36，252 [EK16] S. Evra和T. Kaufman。每个维度的有界度数协同扩张器。在*Proceedings of the 48th annual ACM Symposium on Theory of Computing, 2016*，第36–48页，2016。引用于93 [Eke91] A.K. Ekert。基于贝尔定理的量子密码学。*Physical review letters*，67(6):661，1991。引用于118 [Elg85] T. Elgamal。基于离散对数的公钥密码系统和签名方案。*IEEE Transactions on Information Theory*，31(4):469–472，1985。引用于46 [Eli57] P. Elias。噪声信道中的列表解码。*Research Laboratory of Electronics, Massachusetts Institute of Technology*，1957。引用于235 [EPR35] A. Einstein，B. Podolsky，和N. Rosen。量子力学对物理现实的描述能被认为是完整的吗？*Physical review*，47(10):777，1935。引用于121 [ER59] P. Erdős和A. Rényi。关于随机图，I。*Publicationes Mathematicae*，6:290–297，1959。引用于89 [ER60] P. Erdős和A. Rényi。关于随机图的演变。*Publications of the Mathematical Institute of the Hungarian Academy of Sciences*，5:17–61，1960。引用于89 [Erd47] P. Erdős。关于图论的一些评论。*Bulletin of the American Mathematical Society*，53(4):292–294，1947。引用于82 [Fár48] I. Fáry。关于平面图的直线表示。*Acta Scientarum Mathematicarum (Szeged)*，11:229–233，1948。引用于16 [Fel71] W. Feller。概率论及其应用导论。*Wiley series in probability and mathematical statistics*，1971。引用于145 [Fel93] M. R. Fellows。小学的计算机科学和数学。在Naomi D. Fisher，Harvey B. Keynes和Philip D. Wagreich编辑的*Proceedings of the Mathe- maticians and Education Reform Workshop, Seattle, 1991*，*Issues in Mathe- matics Education*卷的第3卷，第143–163页。数学科学会议委员会，1993。https://larc.unt.edu/ian/research/cseducation/fellows1991.pdf。引用于263 [Fey82] R.P. Feynman。用计算机模拟物理。*International journal of theoretical physics*，21(6):467–488，1982。引用于114，242，243

[Fey86] R.P. Feynman. 量子力学计算机。*Foundations of physics*，16(6):507–531，1986。在243处引用[FFN14] B. Fisch，D. Freund，和M. Naor。物理零知识证明的物理属性。在*International Cryptology Conference*中，第313–336页。Springer，2014。在113处引用[FG98] E. Farhi和S. Gutmann。量子计算与决策树。*Physical Review A*，58(2):915，1998。在116处引用[FGGS00] E. Farhi，J. Goldstone，S. Gutmann，和M. Sipser。通过绝热演化的量子计算。*arXiv preprint quant-ph/0001106*，2000。在120处引用[FGL+96] U. Feige，S. Goldwasser，L. Lovász，S. Safra，和M. Szegedy。交互式证明与近似团问题的困难性。*Journal of the ACM*，第268–292页，1996。在110，235处引用[FHL80] M. Furst，J. Hopcroft，和E. Luks。置换群的多项式时间算法。在*Proceedings of 21st annual IEEE Symposium on Foundations of Computer Science*中，第36–41页。IEEE，1980。在20处引用[FK96] A. Frieze和R. Kannan。正则性引理和稠密问题的近似方案。在*Proceedings of 37th annual IEEE Symposium on Foundations of Computer Science*中，第12–20页。IEEE，1996。在95处引用[FK01] U. Feige和J. Kilian。半随机图问题的启发式算法。*Journal of computer and system sciences*，63(4):639–671，2001。在43，259处引用[FKL+91] A. Fiat，R. M. Karp，M. Luby，L. A Mc Geoch，D. D. Sleator，和N. E. Young。竞争分页算法。*Journal of Algorithms*，12(4):685–699，1991。在181处引用[FKLW03] M. Freedman，A. Kitaev，M. Larsen，和Z. Wang。拓扑量子计算。*Bulletin of the American Mathematical Society*，40(1):31–38，2003。在120处引用[FKO07] U. Feige，G. Kindler，和R. O'Donnell。理解并行重复需要理解泡沫。在*Proceedings of the 22nd annual IEEE conference on computational complexity*中，第179–192页。IEEE，2007。在146处引用[FLP85] M.J. Fischer，N.A. Lynch，和M.S. Paterson。一个故障进程的分布式共识的不可能性。*Journal of the ACM*，32(2):374–382，1985。在223，225处引用[FLvMV05] L. Fortnow，R. Lipton，D. van Melkebeek，和A. Viglas。可满足性的时间-空间下界。*Journal of the ACM*，52(6):835–865，2005。在156处引用[FMP+15] S. Fiorini，S. Massar，S. Pokutta，H.R. Tiwary，和R. Wolf。组合优化中多面体的指数下界。*Journal of the ACM*，62(2):17，2015。在170，235处引用[For66] G.D. Forney。*Concatenated codes*，第11卷。Citeseer，1966。在84，236处引用[For84] E. Formanek。不变量和通用矩阵环。*Journal of Algebra*，89(1):178–223，1984。在151处引用[For94] L. Fortnow。复杂性理论中相对化的作用。*Bulletin EATCS*，52:229–244，1994。在48处引用[For00] L. Fortnow。可满足性的时间-空间权衡。*Journal of computer and system sciences*，60(2):337–353，2000。在48，156处引用

[For13] L. Fortnow. *The golden ticket: P, NP, and the search for the impossible*. Princeton University Press, 2013. 引用于25 [FR03] F. Fich和E. Ruppert. 分布式计算的数百个不可行性结果。*Dis- tributed computing*, 16(2-3):121–163, 2003。引用于218 [Fre81] R. Freivalds. 概率双向机。在 *International Symposium on Mathematical Foundations of Computer Science*, 第33–45页。Springer, 1981。引用于159, 160 [Fre90] Y. Freund. 通过多数投票增强弱学习算法。在*Proceedings of the 3rd Annual Workshop on Computational Learning Theory*, 第90卷, 第202–216页, 1990。引用于197 [FRS88] L. Fortnow, J. Rompel和M. Sipser. 关于多幂交互协议的能力。在 *Proceedings of Third annual Structure in Complexity Theory Conference, 1988.*, 第156–161页。IEEE, 1988。引用于109 [FS95] Y. Freund和R.E. Schapire. 在线学习的决策理论推广及其在增强中的应用。在 *European conference on computational learning theory*, 第23–37页。Springer, 1995。引用于198 [FS99] Y. Freund和R.E. Schapire. 使用乘性权重进行自适应游戏。*Games and Economic Behavior*, 29(1):79–103, 1999。引用于183, 184 [FS13] M.A. Forbes和A. Shpilka. 通过多项式恒等式测试同时共轭的显式Noether规范化。在*Approximation, Randomization, and Combinatorial Optimiza- tion. Algorithms and Techniques*, 第527–542页。Springer, 2013。引用于151, 152 [FSS84] M. Furst, J. Saxe和M. Sipser. 奇偶校验、电路和多项式时间层次。*Math. Systems Theory*, 17:13–27, 1984。引用于52, 55 [FSV17] M.A. Forbes, A. Shpilka和B.L. Volk. 简洁击中集和证明代数电路下界的障碍。*arXiv preprint arXiv:1701.05328*, 2017。引用于132 [Für09] M. Fürer. 更快的整数乘法。*SIAM Journal on computing*, 39(3):979–1005, 2009。引用于51 [FV98] T. Feder和M.Y. Vardi. 单调单态SNP和约束满足的计算结构：通过Datalog和群论的研究。 *SIAM Journal on computing*, 28(1):57–104, 1998。引用于41 [FW81] P. Frankl和R.M. Wilson. 具有几何结果的交点定理。*Combinator- ica*, 1(4):357–368, 1981。引用于84 [Gab72] P. Gabriel. Unzerlegbare darstellungen I。*Manuscripta Mathematica*, 6(1):71–103, 1972。引用于151 [Gal62] R. Gallager. 低密度奇偶校验码。 *IRE Transactions on information theory*, 8(1):21– 28, 1962。引用于236 [GBC16] I. Goodfellow, Y. Bengio和A. Courville。*Deep learning*。MIT Press, 2016。引用于185, 256 [GBG14] A. Glaser, B. Barak和R.J. Goldston. 核弹头验证的无知协议。*Nature*, 510(7506):497, 2014。引用于113 [GBGL10] T. Gowers, J. Barrow-Green和I. Leader。*The Princeton companion to mathematics*。Princeton University Press, 2010。引用于xiii

[Gel15] R. Gelles. 交互式通信的编码：一项调查。*Survey*，2015。`http://www.cs.princeton.edu/~rgelles/papers/survey.pdf`。在176处引用。[Gen09a] C. Gentry。*A fully homomorphic encryption scheme*。博士论文，斯坦福大学，2009。在214处引用。[Gen09b] C. Gentry。使用理想格的完全同态加密。在*Proceedings of the 41st annual ACM symposium on Theory of Computing*，第9卷，第169–178页，2009。在149，214处引用。[GG16] O. Goldreich和T. Gur。通用局部可检验码。在*Electronic Colloquium on Computational Complexity (ECCC)*，第23卷，第1页，2016。在112处引用。[GGH+13] S. Garg，C. Gentry，S. Halevi，M. Raykova，A. Sahai和B. Waters。对所有电路的候选不可区分混淆和功能加密。在*Proceedings of 54th annual IEEE Symposium on Foundations of Computer Science*，第40–49页。IEEE，2013。在216处引用。[GGHR14] S. Garg，C. Gentry，S. Halevi和M. Raykova。从不可区分混淆的二次安全MPC。在*Theory of Cryptography Conference*，第74–94页。Springer，2014。在216处引用。[GGM86] O. Goldreich，S. Goldwasser和S. Micali。如何构造随机函数。*Journal of the ACM*，33(4):792–807，1986。在78，86，200处引用。[GGOW15] A. Garg，L. Gurvits，R. Oliveira和A. Wigderson。非交换有理身份测试的确定性多项式时间算法。*arXiv preprint arXiv:1511.03730*，2015。在132，153，235处引用。[GGOW16] A. Garg，L. Gurvits，R. Oliveira和A. Wigderson。Brascamp-Lieb不等式的算法方面。*arXiv preprint arXiv:1607.06711*，2016。在153处引用。[GGR98] O. Goldreich，S. Goldwasser和D. Ron。属性测试及其与学习和逼近的联系。*Journal of the ACM (JACM)*，45(4):653–750，1998。在112处引用。[GH14] M. Ghaffari和B. Haeupler。交互式编码的最佳错误率II：效率和列表解码。在*Proceedings of 55th annual IEEE Symposium on Foundations of Computer Science*，第394–403页。IEEE，2014。在177处引用。[GHK+16] R. Gelles，B. Haeupler，G. Kol，N. Ron-Zewi和A. Wigderson。向最优确定性编码用于交互式通信迈进。在*Proceedings of the 27th annual ACM-SIAM symposium on discrete algorithms*，第1922–1936页。SIAM，2016。在177处引用。[GHL14] S. Gharibian，Y. Huang和Z. Landau。量子哈密顿量复杂性。*arXiv preprint arXiv:1401.3916*，2014。在118处引用。[GHSY12] P. Gopalan，C. Huang，H. Simitci和S. Yekhanin。关于码字符号的局部性。*IEEE Transactions on Information Theory*，58(11):6925–6934，2012。在236处引用。[Gil52] E.N. Gilbert。信号字母的比较。*Bell System Technical Journal*，31(3):504– 522，1952。在176处引用。[Gil77] J. Gill。概率图灵机的计算复杂性。*SIAM Journal on Com- puting*，6:675–695，1977。在71处引用。[Gil98] D. Gillman。在扩张图上的随机游走的Chernoff界。*SIAM Journal on computing*，27(4):1203–1220，1998。在101处引用。

[GJ79] M.R. Garey 和 D.S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W.H. Freeman and Company, New York, 1979. 引用于 31, 33 [GJL16] H. Guo, M. Jerrum, 和 J. Liu. 通过 Lovász 局部引理进行均匀抽样。*arXiv preprint arXiv:1611.01647*, 2016. 引用于 144, 238 [GK86] S. Goldwasser 和 J. Kilian. 几乎所有素数都可以快速认证。在 *Proceedings of the 18th annual ACM symposium on Theory of Computing*，第 316–329 页。ACM，1986。引用于 136 [GK10] L. Guth 和 N.H. Katz. 关于平面上的 Erdős 不同距离问题。*arXiv preprint arXiv:1011.4105*, 2010。引用于 137 [GK16] S. Goldwasser 和 Y.T. Kalai. 密码学假设：立场论文。在 *Theory of Cryptography Conference*，第 505–522 页。Springer，2016。引用于 204, 260 [GKKS13] A. Gupta, P. Kamath, N. Kayal, 和 R. Saptharishi. 接近深度四的鸿沟。在 *Proceedings of the 28th IEEE conference on Computational complexity*，第 65–73 页。IEEE，2013。引用于 133, 134 [GKM+00] Y. Gertner, S. Kannan, T. Malkin, O. Reingold, 和 M. Viswanathan. 公钥加密与无记忆传输之间的关系。在 *Proceedings of 41st annual IEEE Symposium on Foundations of Computer Science*，第 325–335 页。IEEE，2000。引用于 211 [GKR08] S. Goldwasser, Y.T. Kalai, 和 G.N. Rothblum. 委派计算：为普通人设计的交互式证明。在 *Proceedings of the 40th annual ACM symposium on Theory of Computing*，第 113–122 页。ACM，2008。引用于 215 [GKR14] A. Ganor, G. Kol, 和 R. Raz. 信息与通信的指数分离。在 *Proceedings of 55th annual IEEE Symposium on Foundations of Computer Science*，第 176–185 页。IEEE，2014。引用于 175 [GKR15] A. Ganor, G. Kol, 和 R. Raz. 信息与通信的指数分离。在 *Symposium on the Theory of Computing*，第 557–?566 页。ACM，2015。引用于 175 [GKSS17] J. A. Grochow, M. Kumar, M. Saks, 和 S. Saraf. 通过多项式身份测试向代数自然证明障碍迈进。*arXiv preprint arXiv:1701.01717*，2017。引用于 132 [GL89] O. Goldreich 和 L.A. Levin. 所有单向函数的核。在 *Proceedings of the 21st annual ACM symposium on Theory of Computing*，第 25–32 页。ACM，1989。引用于 236 [GLR10] V. Guruswami, J.R. Lee, 和 A. Razborov. 通过扩张码的几乎欧几里得子空间。*Combinatorica*，30(1):47–68，2010。引用于 138 [GM82] S. Goldwasser 和 S. Micali. 概率加密 & 如何玩心理扑克同时保持所有部分信息秘密。在 *Proceedings of the 14th annual ACM symposium on Theory of Computing*，第 365–377 页。ACM，1982。引用于 210 [GM84] S. Goldwasser 和 S. Micali. 概率加密。*Journal of computer and system sci- ences*，28:270–299，1984。引用于 46, 73, 75, 76, 206, 207, 208 [GMR89] S. Goldwasser, S. Micali, 和 C. Rackoff. 交互式证明系统的知识复杂性。*SIAM Journal on Computing*，18(1):186–208，1989。引用于 104, 107, 141 [GMR+12] P. Gopalan, R. Meka, O. Reingold, L. Trevisan, 和 S. Vadhan. 从较弱的伪随机限制中获得更好的伪随机生成器。在 *Proceedings of 53rd annual IEEE Symposium on Foundations of Computer Science*，第 120–129 页。IEEE，2012。引用于 79

[GMW87] O. Goldreich, S. Micali, 和 A. Wigderson. 如何玩任何心理游戏。在 *Proceedings of the 19th annual ACM symposium on Theory of Computing*，第 218–229 页。ACM，1987。引用于 46，108，160，210，211，213，251 [GMW91] O. Goldreich, S. Micali, 和 A. Wigderson. 只产生其有效性的证明，或所有 NP 语言都具有零知识证明系统。*Journal of the ACM*，38(1):691–729，1991。引用于 103，107，108，209，212，213，254 [GMWW14] D. Gavinsky，O. Meir，O. Weinstein，和 A. Wigderson。向更好的公式下界迈进：KRW 组合猜想的信息复杂度方法。在 *Proceedings of the 46th annual ACM symposium on Theory of Computing*，第 213–222 页。ACM，2014。引用于 53 [GNW11] O. Goldreich，N. Nisan，和 A. Wigderson。关于 Yao 的 XOR-引理。在 *Studies in complexity and cryptography. Miscellanea on the interplay between randomness and computation*，第 273–301 页。Springer，2011。引用于 80 [Goe97] M.X. Goemans。组合优化中的半定规划。*Mathematical Pro- gramming*，79(1-3):143–161，1997。引用于 139 [Gol67] E.M. Gold。极限语言识别。*Information and control*，10(5):447–474，1967。引用于 190 [Gol97] O. Goldreich。关于 Levin 的平均案例复杂度理论的笔记。*ECCC*，TR97-058，1997。引用于 44 [Gol99] O. Goldreich。现代密码学，概率证明和伪随机性。*Algorithms and Combinatorics*，17，1999。引用于 70，73，77，103 [Gol04] O. Goldreich。*Foundations of Cryptography*。剑桥大学出版社，剑桥，2001；2004。I. 基本工具；II. 基本应用。引用于 44，75，77，202，205，212 [Gol08] O. Goldreich。*Computational complexity: a conceptual perspective*。剑桥大学出版社，剑桥，2008。引用于 5，73 [Gol10] O. Goldreich。*Property testing: current research and surveys*，第 6390 卷。Springer，2010。引用于 90 [Gol11] O. Goldreich。短局部可检验码和证明。在 *Studies in Complexity and Cryptography. Miscellanea on the Interplay between Randomness and Computation*，第 333–372 页。Springer，2011。引用于 112 [Gol17] O. Goldreich。*Introduction to property testing*。剑桥大学出版社，2017。引用于 112 [Gow01] W.T. Gowers。Szemerédi 定理的新证明。*Geometric and Functional Analysis*，11(3):465–588，2001。引用于 94 [GPW15] M. Göös，T. Pitassi，和 T. Watson。通信复杂度类的景观。*Computational Complexity*，第 1–60 页，2015。引用于 164 [GPW17] M. Göös，T. Pitassi，和 T. Watson。对于 bpp 的查询到通信提升。在 *Proceedings of 58th annual IEEE Symposium on Foundations of Computer Science*，第 132–143 页。IEEE，2017。引用于 164 [GR07] S. Goldwasser 和 G.N. Rothblum。关于最佳可能的模糊化。在 *Theory of Cryptography Conference*，第 194–213 页。Springer，2007。引用于 216

[GR08a] A. Gabizon和R. Raz. 大域上仿射源的决定性提取器。*Combi- natorica*，28(4):415–440，2008。被引用于102 [GR08b] V. Guruswami和A. Rudra。实现列表解码容量的显式码：最优冗余的错误纠正。*IEEE Transactions on Information Theory*，54(1):135–150，2008。被引用于101，235 [GR18] O. Goldreich和G.N. Rothblum。用于局部可描述集的简单双效交互式证明系统。在*LIPIcs-Leibniz international proceedings in informatics*，第94卷。Schloss Dagstuhl – Leibniz-Zentrum für Informatik，2018。被引用于215 [Gra05] A. Granville。确定一个给定的整数是否为素数很容易。*Bulletin American Math Society*，42:3–38，2005。被引用于19，135 [Gri01a] D. Grigoriev。背包问题的Positivstellensatz证明的复杂性。*Computational com- plexity*，10(2):139–154，2001。被引用于65，235 [Gri01b] D. Grigoriev。对于奇偶性的Positivstellensatz计算证明的线性下界。*Theoretical Computer Science*，259(1):613–622，2001。被引用于66，235 [Gro87] M. Gromov。群论论文集。*Mathematical Sciences Research Institute Publications*，8:75–263，1987。被引用于20 [Gro96] L.K. Grover。数据库搜索的快速量子力学算法。在*Proceedings of the 28th annual ACM symposium on Theory of Computing*，第212–219页。ACM，1996。被引用于116 [Gro03] M. Gromov。随机群中的随机游走。*Geometric and Functional Analysis*，13(1):73– 146，2003。被引用于58，140 [Gro10] M. Gromov。奇点、扩张子和映射拓扑。第2部分：通过代数等周性从组合学到拓扑。*Geometric and Functional Analysis*，20(2):416–526，2010。被引用于93 [Gro15] J. A. Grochow。通过几何复杂性理论统一已知的下界。*Computational Complexity*，24(2):393–475，2015。被引用于131 [GRS90] R.L. Graham，B.L. Rothschild，和J.H. Spencer。*Ramsey Theory*，第20卷。John Wiley and Sons，纽约，1990。被引用于82 [GRS16] V. Guruswami，A. Rudra，和M. Sudan。基本编码理论，2016。`http://www.cse.buffalo.edu/faculty/atri/courses/coding-theory/book/`。被引用于84 [GS71] R.L. Graham和J.H. Spencer。锦标赛问题的构造性解决方案。*Canadian Math Bulletin*，14(1):45–48，1971。被引用于84，86 [GS89] S. Goldwasser和M. Sipser。*Private coins versus public coins in interactive proof systems*，*Advances in Computing Research*的第5卷。JAI Press，Inc.，格林威治，康涅狄格州，1989。Silvio Micali，主编。被引用于103，104，105 [GS98] V. Guruswami和M. Sudan。改进Reed-Solomon和代数几何码的解码。在*Proceedings of 39th annual IEEE Symposium on Foundations of Computer Science*，第28–37页。IEEE，1998。被引用于235 [GS00] O. Goldreich和S. Safra。一个组合一致性引理及其在证明PCP定理中的应用。*SIAM Journal on computing*，29(4):1132–1154，2000。被引用于236

[GS06] O. Goldreich 和 M. Sudan. 本地可检验码和近似线性长度的PCPs。*Journal of the ACM*，53(4):558–655，2006。引用于236 [GS14] L. Gurvits 和 A. Samorodnitsky。永真值和某些应用的范围。在 *Proceedings of 55th annual IEEE Symposium on Foundations of Computer Science*，第 90–99 页。IEEE，2014。引用于237 [GST14] D. Genkin，A. Shamir 和 E. Tromer。通过低带宽声学密码分析提取RSA密钥。在 *International Cryptology Conference*，第 444–461 页。Springer，2014。引用于216，217 [GT09] B. Green 和 T. Tao。有限域上多项式的分布，及其在Gowers范数中的应用。*Contributions to Discrete Mathematics*，4(2):1–36，2009。引用于95 [GTZ12] B. Green，T. Tao 和 T. Ziegler。Gowers $U^{s+1}[N]$-范数的逆定理。*Annals of Mathematics*，176(2):1231–1372，2012。引用于96 [Gur08] L. Gurvits。Van der Waerden/Schrijver-Valiant 类似猜想和稳定（又称双曲）齐次多项式：一个适用于所有的情况的定理。*The electronic journal of combinatorics*，15(1):R66，2008。引用于139 [GUV09] V. Guruswami，C. Umans 和 S. Vadhan。非平衡扩张器和从Parvaresh–Vardy码中提取随机数。*Journal of the ACM*，56(4):20，2009。引用于101 [GV01] D. Grigoriev 和 N. Vorobjov。Null-和Positivstellensatz证明的复杂性。*Annals of Pure and Applied Logic*，113(1):153–160，2001。引用于65 [GV08] A. Goel 和 V. Vogel。利用生物马达设计和纳米尺度运输和组装系统。*Nature nanotechnology*，3(8):465–475，2008。引用于244 [GW96] O. Goldreich 和 A. Wigderson。计算理论：科学视角，1996。`http://www.wisdom.weizmann.ac.il/~oded/toc-sp2.html`。引用于232 [GW11] C. Gentry 和 D. Wichs。将简洁非交互式论证与所有可证伪假设分离。在 *Proceedings of the 43rd annual ACM symposium on Theory of Computing*，第 99–108 页。ACM，2011。引用于204 [GZ11] O. Goldreich 和 D. Zuckerman。另一个证明BPP ⊆ PH（以及更多）。在 *Studies in Complexity and Cryptography. Miscellanea on the Interplay between Randomness and Compu- tation*，第 40–53 页。Springer，2011。引用于97 [Had10] J. Hadamard。*Note sur quelques applications de l'indice de Kronecker*。Bussiere，1910。引用于227 [Hae14] B. Haeupler。重新审视交互信道容量。在 *Proceedings of 55th annual IEEE Symposium on Foundations of Computer Science*，第 226–235 页。IEEE，2014。引用于177 [Hak61] W. Haken。正常流形的理论：圆结的同调判据。*Acta mathematica*，105:245–375，1961。引用于13 [Hak85] A. Haken。解析的不可能性。*Theoretical Computer Science*，39:297–308，1985。引用于67 [HAK07] E. Hazan，A. Agarwal 和 S. Kale。在线凸优化的对数遗憾算法。*Machine Learning*，69(2-3):169–192，2007。引用于184 [Hal05] T.C. Hales。开普勒猜想的证明。*Annals of mathematics*，第 1065–1185 页，2005。引用于14

[Ham50] R.W. Hamming. 检错和纠错码。*Bell System technical journal*，29(2):147–160，1950。引用于176 [Har16] D. Harlow. 黑洞和量子信息耶路撒冷讲座。*Reviews of Modern Physics*，88(1):015002，2016。引用于248 [Has86] John Hastad. 小深度电路的几乎最优下界。在 *Proceedings of the 18th annual ACM symposium on Theory of computing*，第6–20页。Citeseer，1986。引用于78 [Hås90] J. Håstad. 张量秩是NP完全的。*Journal of Algorithms*，11(4):644–654，1990。引用于133 [Hås98] J. Håstad. 德摩根公式的收缩指数是2。*SIAM Journal on computing*，27(1):48–64，1998。引用于52 [Hås99] J. Håstad. 在 $n^{1-\epsilon}$ 内 clique 是难以近似的。*Acta mathematica*，182:105–142，1999。引用于111 [Hås01] J. Håstad. 一些最优不可近似结果。*Journal of the ACM*，48:798–859，2001。引用于111 [Has07] M.B. Hastings. 一维量子系统的面积定律。*Journal of Statistical Mechanics: Theory and Experiment*，2007(08):P08024，2007。引用于119 [Hat10] H. Hatami. 具有小型总影响的布尔函数的结构定理。*arXiv preprint arXiv:1008.1021*，2010。引用于95 [Haz16] E. Hazan. 在线凸优化的介绍。*Foundations and trends in optimization*，2(3-4):157–325，2016。引用于179 [HBD17] H.A. Helfgott，J. Bajpai，和 D. Dona。在准多项式时间内进行图同构。*arXiv preprint arXiv:1710.04574*，2017。引用于20，40，103，141 [HC99] A. Haken 和 S.A. Cook。单调实电路大小的指数下界。*Journal of computer and system sciences*，58(2):326–335，1999。引用于65 [Hea08] A.D. Healy。在 NC^1 内的随机采样。*Computational complexity*，17(1):3– 37，2008。引用于101 [Hel08] H.A. Helfgott。在 $SL_2(\mathbb{Z}_p)$ 中的增长和生成。*Annals of Mathematics*，第601–623页，2008。引用于92 [HEO05] D.F. Holt，B. Eick，和 E.A. O'Brien。*Handbook of computational group theory*。CRC Press，2005。引用于141 [Her91] M. Herlihy。无等待同步。*ACM Transactions on Programming Languages and Systems*，13(1):124–149，1991。引用于223，224 [Het16] S. Hetterich。分析随机公式引导的抽样传播。*arXiv preprint arXiv:1602.08519*，2016。引用于144 [HH13] D. Harlow 和 P. Hayden。量子计算与防火墙。*Journal of High Energy Physics*，6:085，2013。引用于36，248 [Hil93] D. Hilbert。关于完全不变量系统的。*Mathematische Annalen*，42(3):313–373，1893。引用于150，151 [HILL99] J. Håstad，R. Impagliazzo，L.A. Levin，和 M. Luby。从任何单向函数生成伪随机生成器。*SIAM Journal on computing*，28(4):1364–1396，1999。引用于78

[HJ16] Y. Hu 和 H. Jia. GGH 映射的密码分析。在 *Annual International conference on the Theory and Applications of Cryptographic Techniques*，第 537–565 页。Springer，2016。引用于 216 [HKR 13] M. Herlihy，D. Kozlov 和 S. Rajsbaum。*Distributed computing through combinatorial topol- ogy*。Newnes，2013。引用于 225，230 [HL72] O.J. Heilmann 和 E.H. Lieb。单体-二聚体系统的理论。*Communications in Math- ematical Physics*，25(3):190–232，1972。引用于 142 [HLP99] J. Hass，J.C. Lagari as 和 N. Pippenger。结和链接问题的计算复杂性。*Journal of the ACM*，46:185–211，1999。引用于 28，32，40 [HLW06] S. Hoory，N. Linial 和 A. Wigderson。扩张图及其应用。*Bulletin of the American Mathematical Society*，43(4):439–561，2006。引用于 90，91，93，138 [HMV06] D. Hankerso n，A.J. Menezes 和 S. Vanstone。*Guide to elliptic curve cryptography*。Springer Science & Business Media，2 006。引用于 217 [HO13] B. Hemenway 和 R. Ostrovsky。从损失加密构建损失陷阱门函数。在 *International conference on the Theory and Application of Cryptology and Information Security*，第 241 –260 页。Springer，2013。引用于 211 [Hoa78] C.A.R. Hoare。通信顺序进程。在 *The origin of concurrent programming*，第 413–443 页。Springer，1978。引用于 220 [Hru12] P. Hrubeš。关 于距离矩阵的非负秩。*Information Processing Letters*，112(11):457–461，2012。引用于 170，171 [HS65] J. Hartmanis 和 R.E. Stearns。关于算法的计算复杂性。*Transactions of the American Mathematical Society*，1 17:285–306，1965。引用于 48 [HS99] M. Herlihy 和 N. Shavit。异步可计算性的拓扑结构。*Journal of the ACM*，46(6):858–923，1999。引用于 225，226，227，229 [HS11] M. Herlihy 和 N. Shavit。关于进步 的本质。在 *International Conference On Principles Of Distributed Systems*，第 313–328 页。Springer，2011 。引用于 220 [HS17] S.B. Hopkins 和 D. Steurer。从少量样本进行贝叶斯估计：社区检测和相关问题。*arXiv preprint arXiv:1710.00264*，2017。引用于 43，259 [HSSW98] D.P. Helmbold，R.E. Schapire，Y. Singe r 和 M.K. Warmuth。使用乘法更新的在线投资组合选择。*Mathematical Finance*，8(4):325–347，1998。引 用于 184 [HT74] J. Hopcroft 和 R. Tarjan。有效的平面性测试。*Journal of the ACM*，21(4):549–568，1974。 引用于 19 [Huf52] D. Huffman。构建最小冗余码的方法。*Proceedings of the IRE*，9(40):1098–1101，1952。 引用于 174 [HW87] D. Haussler 和 E. Welzl。ε-nets 和单纯形范围查询。*Discrete & Computational Ge- ometry*，2(2):127–151，1987。引用于 194 [HW03] J. Håstad 和 A. Wigderson。简单分析图测试的线性性和 P CP。*Random Structures & Algorithms*，22(2):139–160，2003。引用于 80 [HW14] P. Hrubeš 和 A. Wigderson 。具有除法的非交换算术电路。在 *Proceed- ings of the 5th conference on innovations in theoretical Computer Science*，第 49–66 页。ACM，2014。引 用于 134

[HWY10] P. Hrubeš, A. Wigderson, 和 A. Yehudayoff. 非交换电路和平方和问题。在
Proceedings of the 42nd annual ACM symposium on Theory of Computing，第 667–676 页。ACM，2010。引用于 134 [HY11] P. Hrubeš 和 A. Yehudayoff。代数扩张中的算术复杂性。*Theory of Computing*，7(8):119–129，2011。引用于 125 [Hya79] L. Hyafil。关于多元多项式的并行评估。*SIAM Journal on computing*，8(2):120–123，1979。引用于 126，129 [IKW02] R. Impagliazzo，V. Kabanets，和 A. Wigderson。寻找简单证据：指数时间与概率多项式时间。*Journal of computer Systems and Sciences*，65(4):672– 694，2002。引用于 56，81 [IKW12] R. Impagliazzo，V. Kabanets，和 A. Wigderson。新的直接积测试器和 2 查询 pcps。
SIAM Journal on computing，41(6):1722–1768，2012。引用于 236 [IL90] R. Impagliazzo 和 L.A. Levin。没有比随机选择更好的方法来生成困难的 NP 实例。在
Proceedings of 31st annual IEEE Symposium on Foundations of Computer Science，第 812–821 页。IEEE，1990。引用于 43 [Imm88] N. Immerman。非确定性空间在补集下是封闭的。*SIAM Journal on computing*，17(5):935–938，1988。引用于 155 [Imp95a] R. Impagliazzo。对于某些困难问题的核心分布。在
Proceedings of 36th annual IEEE Symposium on Foundations of Computer Science，第 538–545 页。IEEE，1995。引用于 95 [Imp95b] R. Impagliazzo。关于平均情况复杂性的个人观点。
Proceedings of the 10th annual IEEE conference on structure in complexity theory，第 134–147 页，1995。引用于 44 [Ind00] P. Indyk。稳定分布、伪随机生成器、嵌入和数据流计算。在
Proceedings of 41st annual IEEE Symposium on Foundations of Computer Science，第 189–197 页。IEEE，2000。引用于 157 [INW94] R. Impagliazzo，N. Nisan，和 A. Wigderson。网络算法的伪随机性。在
Proceedings of the 26th annual ACM symposium on Theory of Computing，第 356–364 页。ACM，1994。引用于 89，171，172 [IP01] R. Impagliazzo 和 R. Paturi。关于 k-SAT 的复杂性。
Journal of computer and system sciences，62:367–375，2001。引用于 51，235 [IPU94] R. Impagliazzo，T. Pitassi，和 A. Urquhart。树形切割平面证明的上界和下界。在
Symposium on Logic in computer science, 1994.，第 220–228 页。IEEE，1994。引用于 168，169 [IPZ01] R. Impagliazzo，R. Paturi，和 F. Zane。哪些问题具有强指数复杂性？*Journal of computer and system sciences*，63(4):512–530，2001。引用于 51，235 [IQS15] G. Ivanyos，Y. Qiao，和 K.V. Subrahmanyam。非交换 Edmonds 问题和矩阵半不变量。*arXiv preprint arXiv:1508.00690*，2015。引用于 153 [IV12] T. Ito 和 T. Vidick。针对纠缠证明者的 NEXP 声明多证明者交互证明。在
Proceedings of 53rd annual IEEE Symposium on Foundations of Computer Science，第 243–252 页。IEEE，2012。引用于 121

[IW97] R. Impagliazzo 和 A. Wigderson. P {v*} BPP 除非 E 有亚指数级电路：XOR引理的随机化。在 {v*} {v*}，第 220–229 页。ACM，1997。在 72、80、100、136 处引用 [IW98] R. Impagliazzo 和 A. Wigderson. 随机性 vs. 时间：在均匀假设下的去随机化。在 {v*} {v*}，第 734–743 页，1998。在 73、81 处引用 [IZ89] R. Impagliazzo 和 D. Zuckerman. 如何回收随机比特。在 {v*} {v*}，第 248–253 页。IEEE，1989。在 101 处引用 [Jac94] J. Jackson. 关于均匀分布的 DNF 学习的有效成员查询算法。在 {v*} {v*}，第 42–53 页。IEEE，1994。在 200 处引用 [Jer92] M. Jerrum. 大团子逃避 Metropolis 过程。{v*}，3(4):347–359，1992。在 238 处引用 [Jer96] M. Jerrum. 马尔可夫链蒙特卡洛方法：近似计数和积分的方法。{v*}，第 482–520 页，1996。在 143、237 处引用 [JJUW10] R. Jain、Z. Ji、S. Upadhyay 和 J. Watrous。QIP{v*}PSPACE。{v*}，53(12):102–109，2010。在 121 处引用 [JP04] N. C Jones 和 P. Pevzner。{v*}。MIT Press，2004。在 244 处引用 [JS89] M. Jerrum 和 A. Sinclair. 近似永久量。{v*}，18(6):1149–1178，1989。在 237 处引用 [JS93] M. Jerrum 和 G.B. Sorkin. 图二分法的模拟退火。在 {v*} {v*}，第 94–103 页。IEEE，1993。在 144、238 处引用 [JSV04] M. Jerrum、A. Sinclair 和 E. Vigoda. 矩阵非负项的永久多项式时间近似算法。{v*}，51(4):671–697，2004。在 72、143、237 处引用 [Juk12] S. Jukna。{v*}，第 27 卷。Springer，2012。在 49 处引用 [Jus72] J. Justesen. 构造性渐近良好代数码的类。{v*} {v*}，18(5):652–656，1972。在 84 处引用 [JVV86] M.R. Jerrum、L.G. Valiant 和 V.V. Vazirani。从均匀分布随机生成组合结构。{v*}，43:169–188，1986。在 143、237 处引用 [Kah74] D. Kahn。{v*}。Weidenfeld and Nicolson，1974。在 207 处引用 [Kah95] N. Kahale. 正则图的特征值和扩张。{v*}，42(5):1091–1106，1995。在 91 处引用 [Kah96] D. Kahn。{v*} {v*}。Simon and Schuster，1996。在 202 处引用

[Kal83] E. Kaltofen. 多项式分解。*Computer Algebra: Symbolic and Algebraic Computa- tion, 2nd ed.*，583:95–113，1983。在72处引用。[Kal85] K.A. Kalorkoti。有理函数公式大小的下界。*SIAM Journal on computing*，14(3):678–687，1985。在129处引用。[Kan12] D.M. Kane。关于低抗集中高斯混沌的结构定理及其在多项式阈值函数研究中的应用。在*Proceedings of 53rd annual IEEE Symposium on Foundations of Computer Science*中，第91–100页。IEEE，2012。在95处引用。[Kar72] R. Karp。组合问题的可归约性。*Complexity of computer Computations*，第85–103页，1972。在31、32、33处引用。[Kar76] R.M. Karp。某些组合搜索算法的概率分析。*Algorithms and complexity: New directions and recent results*，1:19，1976。在43、259处引用。[Kar84] N. Karmarkar。线性规划的新多项式时间算法。*Combinatorica*，4:373–394，1984。在19、28、235处引用。[Kar11] R.M. Karp。通过计算视角理解科学。*Journal of Computer Science and Technology*，26(4):569–577，2011。在36处引用。[Kha79] L. Khachian。线性规划的多项式时间算法。*Soviet Math. Doklady*，10:191–194，1979。在19、28处引用。[Kha93] M. Kharitonov。特定分布学习的密码学难度。在*Proceedings of the 25th annual ACM symposium on Theory of Computing*中，第372–381页。ACM，1993。在200处引用。[Kho02] S. Khot。关于唯一2证明1轮游戏的威力。在*Proceedings of the 34th annual ACM symposium on Theory of Computing*中，第767–775页。ACM，2002。在42、43、93、140、235处引用。[Kho10] S. Khot。NP完全问题的不可近似性、离散傅里叶分析和几何。在*Proceedings of the International congress of mathematicians, vol. 5*中，第2676–2697页，新德里，2010。Hindustan Book Agency。在43、140处引用。[Khr71] V. M. Khrapchenko。确定π-方案复杂度下界的算法。*Mathematical Notes*，10(1):474–479，1971。在52处引用。[KI04] V. Kabanets和R. Impagliazzo。身份测试意味着证明电路下界。*Com- putional Complexity*，13(1–2):1–46，2004。在80、132处引用。[Kil88] J. Kilian。在无记忆传输上建立密码学。在*Proceedings of the 20th annual ACM symposium on Theory of Computing*中，第20–31页。ACM，1988。在160处引用。[Kit03] A.Y. Kitaev。任何子代数量子计算的容错性。*Annals of Physics*，303(1):2–30，2003。在117、118处引用。[KKL88] J. Kahn、G. Kalai和N. Linial。变量对布尔函数的影响。在*Pro- ceedings of 29th annual IEEE Symposium on Foundations of Computer Science*中，第68–80页。IEEE，1988。在145处引用。[KKMO07] S. Khot、G. Kindler、E. Mossel和R. O'Donnell。MAX-CUT和其他2变量CSPs的最优不可近似性结果？*SIAM Journal on computing*，37(1):319–357，2007。在145处引用。

[KL82] R. Karp 和 R.J. Lipton. 接受建议的图灵机。*L'Enseignement Mathematique*，2(28)：191–209，1982。引用于50 [KL08] T. Kaufman 和 S. Lovett. 多项式的最坏情况到平均情况缩减。在 *Proceedings of 49th annual IEEE Symposium on Foundations of Computer Science*，第166–175页。IEEE，2008。引用于95 [Kle51] S.C. Kleene. 神经网和有限自动机中事件的表示。技术报告，RAND 项目空军圣莫尼卡 CA，1951。引用于158 [Kle00] J. Kleinberg. 小世界现象：算法视角。在 *Proceedings of the 32nd annual ACM symposium on Theory of Computing*，第163–170页。ACM，2000。引用于252 [KLN06] M. Kassabov，A. Lubotzky，和 N. Nikolov。有限单群作为扩张器。*Proceedings of the National Academy of Sciences*，103(16)：6116–6119，2006。引用于92 [KLOS14] J.A. Kelner，Y.T. Lee，L. Orecchia，和 A Sidford。无向图中近似最大流的几乎线性时间算法及其多商品推广。在 *Proceedings of the twenty-fifth annual ACM-SIAM symposium on discrete algorithms*，第217–226页。SIAM，2014。引用于235 [KLS01] M. Kearns，M.L. Littman，和 S. Singh。博弈论中的图模型。在 *Proceedings of the 17th conference on uncertainty in artificial intelligence*，第253–260页。Morgan Kaufmann 出版社，2001。引用于250 [KMRZS17] S. Kopparty，O. Meir，N. Ron-Zewi，和 S. Saraf。具有亚多项式查询复杂度的高速率局部可纠正和局部可检测码。*Journal of the ACM*，64(2)：11，2017。引用于236 [KMS18] S. Khot，D. Minzer，和 M. Safra。Grassmann 图中的伪随机集具有近乎完美的扩张。在 *Electronic Colloquium on Computational Complexity (ECCC)*，第25卷，2018。引用于43，93 [KMSY14] G. Kol，S. Moran，A. Shpilka，和 A. Yehudayoff。近似非负秩等价于平滑矩形界限。在 *International Colloquium on Automata, Languages, and Programming*，第701–712页。Springer，2014。引用于162 [KN97] E. Kushilevitz 和 N. Nisan。*Communication Complexity*。剑桥大学出版社，1997。引用于161 [KO18] T. Kaufman 和 I. Oppenheim。构造新的局部光谱高维扩张器。在 *Proceedings of the 50th annual ACM Symposium on theory of Computing, 2018*，第773– 786页，2018。引用于93 [Koi12] P. Koiran。算术电路：深度四的鸿沟变得更宽。*Theoretical Computer Science*，448：56–65，2012。引用于133 [KORW08] G. Kindler，R. O'Donnell，A. Rao，和 A. Wigderson。高维度的球体立方体和舍入。在 *Proceedings of 49th annual IEEE Symposium on Foundations of Computer Science*，第189–198页。IEEE，2008。引用于146 [KP89] J. Kraj ček 和 P. Pudlák。命题证明系统，一阶理论的相容性和计算的复杂性。*The Journal of Symbolic Logic*，54(03)：1063–1079，1989。引用于68

[KP95] E. Koutsoupias 和 C.H. Papadimitriou. 关于 k-服务器猜想。*Journal of the ACM*, 42(5):971–983, 1995. 引用于 181 [KP99] E. Koutsoupias 和 C. Papadimitriou. 最坏情况均衡。在 *Proceedings of the 16th annual conference on theoretical aspects of computer science, 1999*，第 404–413 页。Springer，1999。引用于 250 [KPS85] R. Karp，N. Pippenger 和 M. Sipser. 时间-随机性权衡。在 *AMS conference on probabilistic computational complexity*，1985。引用于 92 [KPW92] J. Komlós，J. Pach 和 G. Woeginger。ε-网几乎紧界。*Discrete & Computa- tional Geometry*，7(2):163–173，1992。引用于 194 [KR13] G. Kol 和 R. Raz。交互信道容量。在 *Proceedings of the 45th annual ACM symposium on Theory of Computing*，第 715–724 页。ACM，2013。引用于 177 [Kra94] J. Kraj ́cek。常深命题证明的大小下界。*The Journal of Symbolic Logic*，59(01):73–86，1994。引用于 68 [Kra95] J Kraj ́cek。*Bounded arithmetic, propositional logic and complexity theory*。剑桥大学出版社，1995。引用于 57，58 [Kra19] J. Kraj ́cek。*Proof complexity*。剑桥大学出版社，2019。引用于 57 [Kri64] J. Krivine。预序环。*Journal d'analyse mathématique*，12(1):307–326，1964。引用于 65 [KRR14] Y.T. Kalai，R. Raz 和 R.D. Rothblum。如何委托计算：无信号证明的力量。在 *Proceedings of the 46th annual ACM symposium on Theory of Computing*，第 485–494 页。ACM，2014。引用于 122，215 [KRT17] G. Kol，R. Raz 和 A. Tal。学习稀疏奇偶性的时间-空间难题。在 *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing*，第 1067–1080 页。ACM，2017。引用于 201 [KRVZ11] J. Kamp，A. Rao，S. Vadhan 和 D. Zuckerman。小空间源的决定性提取器。*Journal of Computer and System Sciences*，77(1):191–220，2011。引用于 102 [KRW95] M. Karchmer，R. Raz 和 A. Wigderson。通过通信复杂度中的直接和的超对数深度下界。*Computational complexity*，5(3-4):191–204，1995。引用于 53，54，175 [KS59] R.V. Kadison 和 I.M. Singer。纯态的扩展。*American journal of mathematics*，81(2):383–400，1959。引用于 138 [KS92] B. Kalyanasundaram 和 G. Schintger。集合交集的概率通信复杂度。*SIAM Journal on Discrete Mathematics*，5(4):545–557，1992。引用于 163 [KS01] A.R. Klivans 和 R. Servedio。在时间 $2^{\tilde{o}(n^{1/3})}$ 内学习 DNF。在 *Proceedings of the 33rd annual ACM symposium on Theory of Computing*，第 258–265 页。ACM，2001。引用于 200 [KS09] N. Kayal 和 S. Saraf。深度 3 电路的黑盒多项式身份测试。在 *Proceedings of 50th annual IEEE Symposium on Foundations of Computer Science*，第 198–207 页。IEEE，2009。引用于 112，138，236 [KS13] V. King 和 J. Saia。多项式期望时间内的拜占庭协议。在 *Proceedings of the 45th annual ACM symposium on Theory of Computing*，第 401–410 页。ACM，2013。引用于 224

[KS17] P.K. Kothari 和 D. Steurer. 通过平方和进行异常值稳健矩估计。*arXiv preprint arXiv:1711.11581*，2017。在43，259处引用。[KSV02] A.Y. Kitaev，A. Shen 和 M.N. Vyalyi。*Classical and quantum computation*，第47卷。美国数学学会，2002。在114，118处引用。[KSZ+10] S. Ko，M. Su，C. Zhang，A.E. Ribbe，W. Jiang 和 C. Mao。RNA 和 DNA 分子的协同自组装。*Nature chemistry*，2(12):1050–1055，2010。在244处引用。[KT06] J. Kleinberg 和 E. Tardos。*Algorithm design*。Pearson Education India，2006。在18处引用。[Kup14] G. Kuperberg。结的性质在 NP 中，模 GRH。*Advances in Mathematics*，256:493–506，2014。在40处引用。[KV94a] M. Kearns 和 L.G. Valiant。对学习布尔公式和有限自动机的密码学限制。*Journal of the ACM*，41(1):67–95，1994。在200处引用。[KV94b] M.J. Kearns 和 U. Vazirani。*An introduction to computational learning theory*。MIT Press，1994。在192处引用。[KV03] A. Kalai 和 S. Vempala。通用投资组合的有效算法。*The Journal of Machine Learning Research*，3:423–440，2003。在184处引用。[KV05] S.A. Khot 和 N.K. Vishnoi。唯一游戏猜想，割问题的一致性间隙以及负类型度量的嵌入。在 *Proceedings of 46th annual IEEE Symposium on Foundations of Computer Science*，第53–62页。IEEE，2005。在140处引用。[KW90] M. Karchmer 和 A. Wigderson。单调电路对于连通性需要超对数深度。*SIAM Journal on Discrete Mathematics*，3(2):255–265，1990。在52，54，55，165，166，167处引用。[KY06] G. Kasparov 和 G. Yu。粗几何 Novikov 猜想和一致凸性。*Advances in Mathematics*，206(1):1–56，2006。在140处引用。[LAA87] M.C. Loui 和 H.H. Abu-Amara。不可靠异步进程之间的一致性所需的内存。*Advances in Computer Research*，4:163–183，1987。在224处引用。[Lac15] M. Lackenby。Reidemeister 移动的多项式上界。*Annals of Mathematics*，182:491–564，2015。在28，40，58处引用。[Lac16] M. Lackenby。结的性质和 Thurston 范数的有效认证。*arXiv preprint arXiv:1604.00290*，2016。在40处引用。[Lad75] R. Ladner。关于多项式时间可归约性的结构。*Journal of the ACM*，22(1):155– 171，1975。在38，39处引用。[Laf08] V. Lafforgue。对 (T) 属性的加强。*Duke Mathematical Journal*，143(3):559– 602，2008。在140处引用。[Lag84] J.C. Lagarias。背包公钥密码系统和丢番图逼近。在 *Ad- vances in cryptology*，第3–23页。Springer，1984。在148处引用。[Lan12] J.M. Landsberg。张量：几何和应用。*Representation theory*，381:402，2012。在133处引用。[Lan17] J. Landsberg。*Geometry and complexity theory*。剑桥大学出版社，2017。在150处引用。

[Las01] J.B. Lasserre. 多项式全局优化与矩问题。*SIAM Journal on Optimization*，11(3):796–817，2001。引用于65 [Las09] J. Lasserre. *Moments, positive polynomials and their applications*，第1卷。世界科学出版社，2009。引用于65 [Len83] H.W. Lenstra. 具有固定变量数量的整数规划。*Mathematics of opera- tions research*，8(4):538–548，1983。引用于148 [Lev73] L. A. Levin. 通用搜索问题。*Problemy Peredachi Informatsii*，9(3):115–116，1973。英文翻译 *Problems of Information Transmission* 9(3):265–266，1973。引用于22，30，31 [Lev86] L.A. Levin. 平均情况完全问题。*SIAM Journal on Computing*，15(1):285–286，1986。引用于43，259 [Lev87] L.A. Levin. 单向函数和伪随机生成器。*Combinatorica*，7(4):357–363，1987。引用于47 [LFKN90] C. Lund，L. Fortnow，H. Karloff，和N. Nisan。交互式证明系统的代数方法。在 *Proceedings of 31st annual IEEE Symposium on Foundations of Computer Science*，第2–10页，1990。引用于105 [Li16] X. Li. 改进的两个源提取器，以及多项式对数熵的仿射提取器。在 *Proceedings of 57th annual IEEE Symposium on Foundations of Computer Science*，第168–177页。IEEE，2016。引用于102 [Li17] X. Li. 改进的非可变提取器，非可变码和独立源提取器。在 *Proceedings of the 49th annual ACM symposium on Theory of Computing*，第1144–1156页。ACM，2017。引用于97 [Lin92] N. Linial. 分布式图算法中的局部性。*SIAM Journal on computing*，21(1):193– 201，1992。引用于230，231 [Lin02] N. Linial. 有限度量空间——组合数学、几何和算法。在 *Proceedings of the International Congress of Mathematicians III*，第573–586页。Citeseer，2002。引用于139 [Lin17] Y. Lindell. 如何模拟——模拟证明技术教程。在 *Tutorials on the Foundations of Cryptography*，第277–346页。Springer，2017。引用于207 [Lip91] R. Lipton. 测试的新方向。 *DIMACS Distributed Computing and Cryptography, American Math Society*，2:191–202，1991。引用于106 [Lip94] Richard J Lipton. 直线复杂度和整数分解。在 *International Algorithmic Number Theory Symposium*，第71–79页。1994。引用于124 [Lip95] R.J. Lipton. DNA解决难题。*Science*，268(5210):542，1995。引用于244 [LLL82] A.K. Lenstra，H.W. Lenstra，和L. Lovász. 有理系数多项式分解。*Mathematische Annalen*，261(4):515–534，1982。引用于19，147，148 [LLR95] N. Linial，E. London，和Y. Rabinovich. 图的几何及其算法应用。*Combinatorica*，15(2):215–245，1995。引用于139 [LLS84] R.E. Ladner，R.J. Lipton，和L.J. Stockmeyer. 交替推栈自动机。*SIAM Journal on computing*，13(1):135–155，1984。引用于159

[LM06] N. Linial 和 R. Meshulam. 随机 2-复形的同调连通性。*Combinatorica*，26(4):475–487，2006。被引用于 93 [LMM03] R.J. Lipton，E. Markakis 和 A. Mehta。使用简单策略玩大型游戏。在 *Proceedings of the 4th ACM conference on electronic commerce*，第 36–41 页。ACM，2003。被引用于 250 [LMS11] D. Lokshtanov，D. Marx 和 S. Saurabh。基于指数时间假设的下界。*Bulletin of the EATCS*，3(105):41–72，2011。被引用于 51 [LMSS01] M.G. Luby，M. Mitzenmacher，M.A. Shokrollahi 和 D.A. Spielman。使用不规则图改进低密度奇偶校验码。*IEEE Transactions on Information Theory*，47(2):585–598，2001。被引用于 236 [LN15] M. Lauria 和 J. Nordström。和平方证明的紧大小-度数界限。在 *Proceedings of the 30th conference on computational complexity*，第 448–466 页。Schloss Dagstuhl – Leibniz-Zentrum für Informatik，2015。被引用于 66 [Lov12] L. Lovász。*Large networks and graph limits*，第 60 卷。美国数学会，2012。被引用于 90，240 [Lov14] S. Lovett。通信复杂性中 log-rank 假设的最近进展。*arXiv preprint arXiv:1403.8106*，2014。被引用于 161，162 [LP06] A. Livnat 和 N. Pippenger。一个最优大脑可以由冲突的代理组成。*Pro- ceedings of the National Academy of Sciences*，103(9):3198–3202，2006。被引用于 246 [LP09] L. Lovász 和 M.D. Plummer。*Matching theory*，第 367 卷。美国数学会，2009。被引用于 19 [LP16] A. Livnat 和 C. Papadimitriou。性别作为算法：从计算的角度看进化。*Communications of the ACM*，59(11):84–93，2016。被引用于 246 [LPDF08] A. Livnat，C. Papadimitriou，J. Dushoff 和 M.W. Feldman。进化中性别作用的混合理论。*Proceedings of the National Academy of Sciences*，105(50):19803–19808，2008。被引用于 246 [LPPF10] A. Livnat，C. Papadimitriou，N. Pippenger 和 M.W. Feldman。性别、混合性和模块化。*Proceedings of the National Academy of Sciences*，107(4):1452–1457，2010。被引用于 246 [LPS88] A. Lubotzky，R. Phillips 和 P. Sarnak。拉马努金图。*Combinatorica*，8(3):261–277，1988。被引用于 91，92，139 [LPS14] J.W. Lichtman，H. Pfister 和 N. Shavit。连接组学的大数据挑战。*Nature neuroscience*，17(11):1448–1454，2014。被引用于 247 [LR81] D. Lehmann 和 M.O. Rabin。自由选择的优势：解决就餐哲学家问题的对称和完全分布式解决方案。在 *Proceedings of the 8th ACM SIGPLAN-SIGACT symposium on principles of programming languages*，第 133–138 页。ACM，1981。被引用于 221，222，224 [LRS14] J.R. Lee，P. Raghavendra 和 D. Steurer。半定规划松弛的大小下界。*arXiv preprint arXiv:1411.6317*，2014。被引用于 66，235 [LRVW03] C. Lu，O. Reingold，S. Vadhan 和 A. Wigderson。提取器：最优至常数因子。在 *Proceedings of the 35th annual ACM symposium on Theory of Computing*，第 602–611 页。ACM，2003。被引用于 137

[LS91] L. Lovász和A. Schrijver. 矩阵和集合函数的锥以及0-1优化。*SIAM Journal on Optimization*，1(2)：166–190，1991。被引用于65 [LS09] T. Lee和A. Shraibman. 通信复杂度下的下界。*Foundations and Trends in Theoretical Computer Science*，3(4)：263–399，2009。被引用于162 [LS14] Y.T. Lee和A. Sidford. 线性规划的路径寻找方法：在o(vrank)次迭代中解决线性规划以及最大流的更快算法。在*Proceedings of 55th annual IEEE Symposium on Foundations of Computer Science*，第424–433页。IEEE，2014。被引用于235 [LSP82] L. Lamport，R. Shostak，和M. Pease. 希腊将军问题。*ACM Transactions on Programming Languages and Systems*，4(3)：382–401，1982。被引用于222，223 [LSW00] N. Linial，A. Samorodnitsky，和A. Wigderson. 矩阵缩放和近似永久的确定性强多项式算法。*Combinatorica*，20(4)：545–568，2000。被引用于237 [LT89] N.A. Lynch和M.R. Tuttle. 输入/输出自动机简介。*CWI Quarterly*，2(3)：219–246，1989。被引用于222 [Lub02] M. Luby. LT码。在*Proceedings of 43rd annual IEEE Symposium on Foundations of Computer Science*，第271–280页。IEEE，2002。被引用于236 [Lub14] A. Lubotzky. 拉马努金复形和高维扩张子。*Japanese Journal of Mathematics*，9(2)：137–169，2014。被引用于93 [LVV13] Z. Landau，U. Vazirani，和T. Vidick. 1d间隙局部哈密顿量基态的多项式时间算法。*arXiv preprint arXiv:1307.5143*，2013。被引用于120 [LW86] N. Littlestone和M. Warmuth. 将数据压缩与学习性联系起来。技术报告，加州大学圣克鲁兹分校，1986。被引用于197 [LW92] A. Lubotzky和B. Weiss. 群和扩张子。在Joel Friedman，编辑的*Expanding graphs, Proceedings of a DIMACS Workshop*，第95–110页。DIMACS/AMS，1992。被引用于92 [LW94] N. Littlestone和M.K. Warmuth. 加权多数算法。*Information and com- putation*，108(2)：212–261，1994。被引用于182，183 [Lyn89] N. Lynch. 分布式计算的百个不可能性证明。在*Proceedings of the 8th annual ACM symposium on principles of distributed computing*，第1–28页。ACM，1989。被引用于218 [Lyn96] N.A. Lynch. *Distributed algorithms*。Morgan Kaufmann，1996。被引用于218 [LZ77] R.J. Lipton和Y. Zalcstein. 可在Logspace中求解的词问题。*Journal of the ACM*，24(3)：522–526，1977。被引用于154 [Mad13] A. Madry. 使用电流量导航中心路径：从流量到匹配，再回到流量。在*Proceedings of 54th annual IEEE Symposium on Foundations of Computer Science*，第253–262页。IEEE，2013。被引用于235 [Mah18] U. Mahadev. 量子计算的经典验证。在*2018 IEEE 59th Annual Symposium on Foundations of Computer Science*，第259–267页。IEEE，2018。被引用于121，215 [Mal99] J. Maldacena. 超共形场理论和超引力的N大极限。在*AIP conference proceedings CONF-981170*，第484卷，第51–63页。AIP，1999。被引用于249

[Man80] Y.I. Manin. *Vychislimoe i nevychislimoe (Computable and Noncomputable)*. 苏联广播，1980. 俄语。在第114处引用 [Mar73] G.A. Margulis. 显式构造集中器。*Problems Information Transmission*，9(4):71–80，1973。在第91、92处引用 [Mar88] G.A. Margulis. 显式组合方案群论构造及其在构造扩展器和集中器中的应用。*Problems Information Trans- mission*，24:39–46，1988。在第91、92、139处引用 [Mar06] H. Markram. 蓝脑项目。*Nature Neuroscience*，7(2):153，2006。在第247处引用 [Mat02] J. Matoušek. *Lectures on discrete geometry*，第108卷。Springer，2002。在第194处引用 [Maz18] A. Mazumdar. 局部可恢复码的容量。在*2018 IEEE Information Theory Workshop (ITW)*，第1–5页。IEEE，2018。在第112处引用 [McG14] A. McGregor. 图流算法综述。*ACM SIGMOD Record*，43(1):9–20，2014。在第156处引用 [Mer97] F. Mertens. 关于一个数论函数。*Akademie Wissenschaftlicher Wien Mathematisch-Naturwissenschaftliche Klasse, Abteilung 2a*，106:761–830，1897。在第85处引用 [MFK82] D. Mumford，J. Fogarty，和 F. Kirwan. *Geometric invariant theory*，第34卷。Springer-Verlag，第2版，1982。在第150处引用 [Mil67] S. Milgram. 小世界问题。*Phychology Today*，1:61–67，1967。在第252处引用 [Mil76] G.L. Miller. 黎曼猜想和素性检验。*Journal of computer and system sciences*，13(3):300–317，1976。在第28、87、136处引用 [Min10] H. Minkowski. *Geometrie der Zahlen*。Teubner，1910。在第147处引用 [Mit97] T. M. Mitchell. 机器学习真的有效吗？*AI Magazine*，18(3):11–20，1997。在第185处引用 [MM11] C. Moore 和S. Mertens. *The nature of computation*。牛津大学出版社，2011。在第5处引用 [MMS90] M.S. Manasse，L.A. McGeoch，和D.D. Sleator. 服务器问题的竞争算法。*Journal of Algorithms*，11(2):208–230，1990。在第181处引用 [MN14] M. Mendel和A. Naor. 非线性谱计算和超扩展器。*Publications mathématiques de l'IHÉS*，119(1):1–95，2014。在第140处引用 [Moi16] A. Moitra. 近似计数、Lovász局部引理和图形模型中的推理。*arXiv preprint arXiv:1610.04317*，2016。在第144、238处引用 [MOO10] E. Mossel，R. O'Donnell，和K. Oleszkiewicz. 低影响函数的噪声稳定性：不变性和最优性。*Annals of Mathematics*，171(1):295–341，2010。在第145、146处引用 [Mor85] J. Morgenstern. 如何快速计算一个函数及其所有导数：Baur-Strassen定理的一个变体。*ACM SIGACT News*，16(4):60–62，1985。在第127处引用 [Mos09] R.A. Moser. Lovász局部引理的一个构造性证明。在*Proceedings of the 41st annual ACM symposium on Theory of Computing*，第343–350页。ACM，2009。在第144、238处引用 [MP43] W.S. McCulloch和W. Pitts. 神经活动中内在思想的逻辑演算。*The bulletin of mathematical biophysics*，5(4):115–133，1943。在第158、247处引用

[MP69] M. Minsky 和 S. Papert. 感知器。*MIT Press*，18:19，1969。在200处引用。[MP15] R. Meir 和 D. Parkes。关于性别、进化和乘性权重更新算法。在
Proceedings of the 2015 international conference on autonomous agents and multiagent systems，第929-937页。国际自主代理和多智能体系统基金会，2015。在182，246处引用。[MPZ02] M. Mézard，G. Parisi 和 R. Zecchina。随机可满足性问题解析和算法解。*Science*，297(5582):812–815，2002。在144处引用。[MR95] R. Motwani 和 P. Raghavan。*Randomized Algorithms*。剑桥大学出版社，剑桥，1995。在70，72，80处引用。[MR04] T. Mignon 和 N. Ressayre。行列式和永真值问题的二次界限。
International Mathematics Research Notices，2004(79):4241–4253，2004。在131处引用。[MR11] E.W. Mayr 和 S. Ritscher。无度数界限的空间高效Groöbner基计算。在
Proceedings of the 36th international symposium on symbolic and algebraic computation，第257-264页。ACM，2011。在151处引用。[MR13] M. Mohri 和 A. Rostamizadeh。感知器错误界限。
arXiv preprint arXiv:1305.0208，2013。在187处引用。[MS82] K. Mehlhorn 和 E.M. Schmidt。在VLSI和分布式计算中，拉斯维加斯比确定性更好。在
Proceedings of the 14th annual ACM symposium on Theory of Computing，第330-337页。ACM，1982。在162处引用。[MS14a] C.A. Miller 和 Y. Shi。使用不受信任的量子设备安全扩展随机性和分发密钥的鲁棒协议。在*Proceedings of the 46th annual ACM symposium on Theory of Computing*，第417-426页。ACM，2014。在123处引用。[MS14b] C. Moore 和 L.J. Schulman。树码和指数和的一个猜想。在*Proceedings of the 5th conference on innovations in theoretical Computer Science*，第145-154页。ACM，2014。在178处引用。[MS16] C.A. Miller 和 Y. Shi。使用不受信任的量子设备安全扩展随机性和分发密钥的鲁棒协议。*Journal of the ACM (JACM)*，63(4):33，2016。在97，123处引用。[MSS13a] A. Marcus，D.A. Spielman 和 N. Srivastava。交错家族I：所有度数的二部拉马努金图。在
Proceedings of 54th annual IEEE Symposium on Foundations of Computer Science，第529-537页。IEEE，2013。在93，138，139处引用。[MSS13b] A. Marcus，D.A. Spielman 和 N. Srivastava。交错家族II：混合特征多项式和Kadison-Singer问题。*arXiv preprint arXiv:1306.3969*，2013。在138，139处引用。[MT10] R.A. Moser 和 G. Tardos。一般洛瓦茨局部引理的构造性证明。*Journal of the ACM*，57(2):11，2010。在144，238处引用。[Mul11] K.D. Mulmuley。关于P与NP以及几何复杂性理论。*Journal of the ACM*，58(2):5，2011。在131，150处引用。[Mul12a] K.D. Mulmuley。GCT项目向P与NP问题迈进。*Communications of the ACM*，55(6):98–107，2012。在131，150处引用。

[Mul12b] K.D. Mulmuley. 几何复杂性理论 V：黑盒多项式恒等式测试的去随机化与诺特定理去随机化之间的等价性。在 *Proceedings of 53rd annual IEEE Symposium on Foundations of Computer Science*，第 629–638 页。IEEE，2012。在 151、152 处引用 [Mum95] D. Mumford. *Algebraic Geometry: Complex projective varieties*，第 1 卷。Springer Science & Business Media，1995。在 149 处引用 [Mur12] K.P. Murphy. *Machine learning: a probabilistic perspective*。MIT Press，2012。在 185 处引用 [Mut05] S. Muthukrishnan. *Data streams: Algorithms and applications*。Now Publishers Inc，2005。在 156 处引用 [MW04] R. Meshulam 和 A. Wigderson. 群代数中的扩张器。*Combinatorica*，24(4):659–680，2004。在 92 处引用 [MWW16] W. Mou，Z. Wang 和 L. Wang。海马体中的稳定内存分配：基本限制和神经实现。*arXiv preprint arXiv:1612.04659*，2016。在 247 处引用 [MY16] S. Moran 和 A. Yehudayoff。VC 类的样本压缩方案。*Journal of the ACM*，63(3):21，2016。在 197 处引用 [Nao03] M. Naor。关于密码学假设和挑战。在 *Annual International Cryptology Conference*，第 96–109 页。Springer，2003。在 204 处引用 [NC10] M.A. Nielsen 和 I.L. Chuang。*Quantum computation and quantum information*。剑桥大学出版社，2010。在 114 处引用 [Nes00] Y. Nesterov。平方函数系统和优化问题。在 *High performance optimization*，第 405–440 页。Springer，2000。在 65 处引用 [New91] I. Newman。通信复杂性中的私有随机比特与公共随机比特。*Information processing letters*，39(2):67–71，1991。在 162 处引用 [Nil91] A. Nilli。关于图的第二个特征值。*Discrete Mathematics*，91(2):207–210，1991。在 91 处引用 [Nis91a] N. Nisan。非交换计算的界限。在 *Proceedings of the 23rd annual ACM symposium on Theory of Computing*，第 410–418 页。ACM，1991。在 134 处引用 [Nis91b] N. Nisan。常深度电路的伪随机比特。*Combinatorica*，11(1):63–70，1991。在 79 处引用 [Nis92] N. Nisan。空间受限计算的伪随机生成器。*Combinatorica*，12(4):449–461，1992。在 88、156、171、172 处引用 [Nis94] N. Nisan。RL $\subseteq$ SC。*Computational complexity*，4(1):1–11，1994。在 156 处引用 [Nis96] N. Nisan。提取随机性：如何和为什么。综述。在 *Proceedings of the 11th annual IEEE conference on Computational complexity*，第 44–58 页。IEEE，1996。在 97 处引用 [Nov62] A. Novikoff。关于感知器收敛证明。在 *Proceedings of the symposium on the mathematical theory of automata*，第 12 卷，第 615–620 页。Polytechnic Press，1962。在 187、191 处引用 [NRTV07] N. Nisan，T. Roughgden，E. Tardos 和 V.V. Vazirani。*Algorithmic game theory*，第 1 卷。剑桥大学出版社，2007。在 36、250 处引用

[NW94] N. Nisan 和 A. Wigderson. 难度与随机性。*Journal of Computer and System Sciences*，49(2):149–167，1994。在79，100处引用[NW96] N. Nisan 和 A. Wigderson. 通过偏导数对算术电路的下界。*Computational complexity*，6(3):217–234，1996。在134处引用[NY90] M. Naor 和 M. Yung. 对抗选择密文攻击的公钥密码系统。在*Proceedings of the 22nd annual ACM symposium on Theory of Computing*中，第427–437页。ACM，1990。在207处引用[NY17] A. Naor 和 R. Young. 垂直周长与水平周长。*arXiv preprint arXiv:1701.00620*，2017。在140处引用[NZ96] N. Nisan 和 D. Zuckerman. 随机性是空间的线性。*Journal of computer and system sciences*，52(1):43–52，1996。在99，144处引用[O'D05] R. O'Donnell. PCP 定理的历史。*Course notes on the PCP Theorem and Hardness of Approximation*，2005。在105处引用[O'D14] R. O'Donnell. *Analysis of Boolean functions*。剑桥大学出版社，2014。在43，145处引用[O'D17] R. O'Donnell. SOS 并非显然可自动化的，即使是近似的。在*8th Innovations in Theoretical Computer Science conference (ITCS 2017)*中。Schloss Dagstuhl – Leibniz-Zentrum für Informatik，2017。在65处引用[Orl92] A. Orlitsky. 平均情况下的交互通信。*IEEE Transactions on Information Theory*，38(5):1534–1547，1992。在174处引用[Orú14] R. Orús. 张量网络的实用介绍：矩阵乘积态和投影纠缠态。*Annals of Physics*，349:117–158，2014。在119处引用[OtR85] A. M. Odlyzko 和 H. J. J. te Riele. Mertens 猜想的反证。*Journal für die reine und angewandte Mathematik*，357:138–160，1985。在148处引用[OW93] R. Ostrovsky 和 A. Wigderson. 单向函数对于非平凡零知识是必要的。在*[1993] The 2nd Israel Symposium on Theory and Computing Systems*中，第3–17页。IEEE，1993。在107处引用[Pal33] R.E. Paley. 关于正交矩阵。*Journal of Mathematics and Physics*，12(1–4):311–320，1933。在84处引用[PAM+10] S. Pironio，A. Ac´ın，S. Massar，A.. de la Giroday，D.N. Matsukevich，P. Maunz，S. Olmschenk，D. Hayes，L. Luo，T.A. Manning，和 C. Monroe. 由贝尔定理认证的随机数。*Nature*，464(7291):1021–1024，2010。在123处引用[Pap94] C.H. Papadimitriou. 关于奇偶性论证的复杂性和其他低效的存在证明。*Journal of Computer and System Sciences*，48(3):498–532，1994。在38，59，250处引用[Pap97] C. H. Papadimitriou. NP-完全性：回顾。在*Proceedings of the 24th international colloquium on automata, languages and programming*中，第2–6页。Springer-Verlag，1997。在33处引用[Pap03] C.H. Papadimitriou. *Computational complexity*。John Wiley and Sons Ltd.，2003。在5处引用[Pap14] C. Papadimitriou. 算法、复杂性和科学。*Proceedings of the National Academy of Sciences*，111(45):15881–15887，2014。在36处引用

[Par00] P.A. Parrilo. *Structured semidefinite programs and semialgebraic geometry methods in robustness and optimization*. 博士学位论文，Citeseer，2000。被引用于65 [PB94] P. Pudlák和S.R. Buss。如何在命题演算中撒谎而不被（轻易）定罪以及证明的长度。在 *International Workshop on Computer Science Logic*，第933卷，第151–162页。Springer，1994。被引用于168 [Pei09] C. Peikert。从最坏情况最短向量问题中得到的公钥密码系统。在 *Pro- ceedings of the 41st annual ACM symposium on Theory of Computing*，第333–342页。ACM，2009。被引用于215 [Pei16] C. Peikert。十年来的格密码学。
Foundations and Trends in Theoretical Computer Science，10(4):283–424，2016。被引用于149 [Pev00] P. Pevzner。*Computational molecular biology: an algorithmic approach*。麻省理工学院出版社，2000。被引用于244 [Pig13] G. Pighizzini。双向有限自动机：旧的和最近的结果。*Fundamenta Informaticae*，126(2-3):225–246，2013。被引用于159 [Pin73] M.S. Pinsker。关于集中器的复杂性。在
7th International Telegraphic Confer- ence，第4卷，第1–318页。Citeseer，1973。被引用于90 [PPST83] W.J. Paul，N. Pippenger，E. Szemeredi和W.T. Trotter。关于确定性与非确定性及其相关问题。在
Proceedings of 24th annual IEEE Symposium on Foun- dations of Computer Science，第429–438页。IEEE，1983。被引用于48 [PR17] T. Pitassi和R. Robere。单调计算的强指数下界。在
Proceedings of the 49th annual ACM symposium on Theory of Computing，第1246–1255页。ACM，2017。被引用于54，167 [Pro76] C. Procesi。$n \times n$矩阵的不变量理论。*Advances in Mathematics*，19(3):306–381，1976。被引用于151 [Pud97] P. Pudlak。关于归结和切割平面证明以及单调计算的下界。
Journal of Symbolic Logic，62(3):981–998，1997。被引用于64 [Put93] M. Putinar。在紧半代数集上的正多项式。*Indiana University Mathe- matics Journal*，42(3):969–984，1993。被引用于65 [PV88] L. Pitt和L.G. Valiant。从示例中学习的计算限制。*Journal of the ACM*，35(4):965–984，1988。被引用于200 [PV05] F. Parvaresh和A. Vardy。在多项式时间内纠正Guruswami-Sudan半径之外的错误。在
Proceedings of 46th annual IEEE Symposium on Foundations of Computer Science，第285–294页。IEEE，2005。被引用于101，235 [PV15] C.H. Papadimitriou和S. Vempala。通过预测进行皮层学习。在
Conference on Learning Theory，第1402–1422页，2015。被引用于247 [PW85] J. Paris和A. Wilkie。有界算术中的计数问题。在 *Methods in mathematical logic*，第317–340页。Springer，1985。被引用于58 [PW11] C. Peikert和B. Waters。有损门控函数及其应用。*SIAM Journal on computing*，40(6):1803–1844，2011。被引用于47 [PZ89] M. Pohst和H. Zassenhaus。*Algorithmic algebraic number theory*。剑桥大学出版社，1989。被引用于147

[Rab63] M.O. Rabin. 概率自动机。*Information and control*, 6(3)：230–245，1963。引用于159 [Rab67] M. O. Rabin. 自动机的数学理论。在Jacob T. Schwartz编辑的*Mathematical aspects of computer science, Proceedings of symposia in applied mathematics*中，第153–175页。美国数学会，1967。引用于17 [Rab79] M.O. Rabin. 数字签名和公钥函数如同因子分解一样难以处理。技术报告，DTIC文档，1979。引用于206 [Rab80] M.O. Rabin. 测试素性的概率算法。*Journal of number theory*, 12(1)：128–138，1980。引用于71，87，136 [Rab81] M.O. Rabin. 如何通过无意识传递交换秘密。*Technical Memo TR-81*，1981。引用于213 [RAD78] R.L. Rivest，L. Adleman和M.L. Dertouzos。关于数据银行和隐私同态。*Foundations of secure computation*, 4(11)：169–180，1978。引用于214 [Rag08] P. Raghavendra。每个CSP的最优算法和不近似性结果？在*Proceedings of the 40th annual ACM symposium on Theory of Computing*中，第245–254页。ACM，2008。引用于42，235 [Ram30] F.P. Ramsey。关于形式逻辑的一个问题。*Proceedings of the London Mathematical Society*, 30：264–286，1930。引用于82 [Raz74] J.P. Razmyslov。零特征域上全矩阵代数的迹恒等式。*Mathematics of the USSR-Izvestiya*, 8(4)：727，1974。引用于151 [Raz85a] A.A. Razborov。某些布尔函数单调复杂度的下界。*Dokl. Akad. Nauk SSSR*, 281(4)：798–801，1985。英文翻译*Soviet Math. Doklady* **31** (1985)，354–357。引用于53 [Raz85b] A.A. Razborov。逻辑永真式的单调复杂度的下界。*Matem- aticheskie Zametki*, 37(6)：887–900，1985。引用于54 [Raz92] A.A. Razborov。关于不相交性的分布复杂度。*Theoretical Computer Sci- ence*, 106(2)：385–390，1992。引用于163 [Raz95a] A.A. Razborov。有界算术和布尔复杂度的下界。在*Feasible Mathematics II*中，第344–386页。Springer，1995。引用于68 [Raz95b] A.A. Razborov。在某些有界算术片段中电路大小下界的不可证明性。*Izvestiya of the Russian Academy of Science, Mathematics*, 59(1)：201–224，1995。引用于55，68 [Raz96] A.A. Razborov。命题证明和有界算术中的独立性结果的下界。在*Automata, Languages and Programming*中，第48–62页。Springer，1996。引用于68 [Raz98a] R. Raz。并行重复定理。*SIAM Journal on computing*, 27(3)：763–803，1998。引用于146 [Raz98b] A.A. Razborov。多项式演算的下界。*Computional Complexity*, 7(4)：291–324，1998。引用于63 [Raz04a] R. Raz。弱鸽巢原理的归结下界。*Journal of the ACM*, 51(2)：115–138，2004。引用于69，132

[Raz04b] A.A. Razborov. 完美匹配原理的解分辨率下界。*Journal of computer and system sciences*，69(1):3–27，2004。被引用于69 [Raz11] R. Raz. 强并行重复的反例。*SIAM Journal on computing*，40(3):771–777，2011。被引用于146 [Reg09] O. Regev. 关于格、错误学习、随机线性码和密码学。*Journal of the ACM*，56(6):34，2009。被引用于215 [Rei08] O. Reingold. 无向连通性在对数空间中。*Journal of the ACM*，55(4):17，2008。被引用于80，92，155 [Rem16] Z. Remscrim. 希尔伯特函数、代数提取器和递归傅里叶采样。在 *Proceedings of 57th annual IEEE Symposium on Foundations of Computer Science*，第197–208页。IEEE，2016。被引用于102 [RM99] R. Raz 和 P. McKenzie。单调 NC 层次的分离。*Combinatorica*，19(3):403– 435，1999。被引用于54 [Ros58] F. Rosenblatt. 感知器：大脑中信息存储和组织的一种概率模型。*Psychological review*，65(6):386，1958。被引用于187 [Ros97] A. Rosenbloom。单调实电路比单调布尔电路更强大。*Information Processing Letters*，61(3):161–164，1997。被引用于65 [Rot53] K.F. Roth。关于某些整数集。*Journal of the London Mathematical Society*，1(1):104– 109，1953。被引用于94 [Rot06] R. Roth。*Introduction to Coding Theory*。剑桥大学出版社，2006。被引用于84 [Rot14] T. Rothvoß。匹配多面体的指数扩展复杂性。在 *Proceedings of the 46th annual ACM symposium on Theory of Computing*，第263–272页。ACM，2014。被引用于171 [Rou16] T. Roughgarden。通信复杂性（针对算法设计师）。*Foundations and Trends in Theoretical Computer Science*，11(3–4):217–404，2016。被引用于161，164 [RPRC16] R. Robere，T. Pitassi，B. Rossman 和 S. A. Cook。单调跨度程序的指数下界。在 *2016 IEEE 57th Annual Symposium on Foundations of Computer Science*，第406–415页。IEEE，2016。被引用于54，167 [RPW04] P.W.K. Rothemund，N. Papadakis 和 E. Winfree。DNA Sier- pinski 三角形的算法自组装。*PLoS biology*，2(12):e424，2004。被引用于244 [RR97] A.A. Razborov 和 S. Rudich。自然证明。*Journal of computer and system sciences*，55(1):24–35，1997。被引用于55，86，241 [RR99] R. Raz 和 O. Reingold。关于在空间有界计算中回收状态随机性的问题。在 *Proceedings of the 31st annual ACM symposium on Theory of Computing*，第159–168页。ACM，1999。被引用于97 [RRR16] O. Reingold，G.N. Rothblum 和 R.D. Rothblum。委托计算的常轮交互证明。在 *Proceedings of the 48th annual ACM symposium on Theory of Computing*，第49–62页。ACM，2016。被引用于215 [RS59] M.O. Rabin 和 D. Scott。有限自动机和它们的决策问题。*IBM journal of research and development*，3(2):114–125，1959。被引用于158

[RS95] N. Robertson 和 P. Seymour. 图的 minors I–XIII. *Journal of Combinatorial Theory B*，1983–1995. 引用次数 20 [RS97] R. Raz 和 S. Safra. 一个亚常数的错误概率低度测试，以及 NP 的亚常数的错误概率 PCP 特化。在 *Proceedings of the 29th annual ACM symposium on Theory of Computing*，第 475–484 页。ACM，1997。引用次数 236 [RS06] V. Rödl 和 J. Skokan. 均匀超图正则性引理的应用。*Random Structures & Algorithms*，28(2):180–194，2006。引用次数 96 [RS10] P. Raghavendra 和 D. Steurer. 图扩张和唯一游戏猜想。在 *Proceedings of the 42nd annual ACM symposium on Theory of Computing*，第 755–764 页。ACM，2010。引用次数 43 [RS11] R. Rubinfeld 和 A. Shapira. 亚线性时间算法。*SIAM Journal on Discrete Mathe- matics*，25(4):1562–1588，2011。引用次数 112 [RSA78] R.L. Rivest，A. Shamir，和 L. Adleman. 获得数字签名和公钥密码系统的方法。*Communications of the ACM*，21(2):120–126，1978。引用次数 206 [RSD16] O. Regev 和 N. Stephens-Davidowitz. 反 Minkowski 定理。*arXiv preprint arXiv:1611.05979*，2016。引用次数 147 [RSW04] E. Rozenman，A. Shalev，和 A. Wigderson. 一个新的凯莱扩张器族 (?). 在 *Proceedings of the 36th annual ACM symposium on Theory of Computing*，第 445–454 页。ACM，2004。引用次数 92 [RT02] T. Roughgarden 和 É. Tardos. 自私路由有多糟糕? *Journal of the ACM*，49(2):236–259，2002。引用次数 250 [RT18] R. Raz 和 A. Tal. BQP 和 PH 的算子分离。*Electronic Colloquium on Computational Complexity (ECCC)*，25:107，2018。引用次数 79，116 [RTS00] J. Radhakrishnan 和 A. Ta-Shma. 分散器、提取器和深度二超级集中器的界限。*SIAM Journal on Discrete Mathematics*，13(1):2–24，2000。引用次数 100 [RTTV08] O. Reingold，L. Trevisan，M. Tulsiani，和 S. Vadhan. 伪随机集的密集子集。在 *Proceedings of 49th annual IEEE Symposium on Foundations of Computer Science*。IEEE，2008。引用次数 95 [RTV06] O. Reingold，L. Trevisan，和 S. Vadhan. 正则有向图上的伪随机游走和 RL vs. L 问题。在 *Proceedings of the 38th annual ACM symposium on Theory of Computing*，第 457–466 页。ACM，2006。引用次数 156 [RU01] T.J. Richardson 和 R.L. Urbanke. 在消息传递解码下的低密度奇偶校验码的容量。*IEEE Transactions on Information Theory*，47(2):599–618，2001。引用次数 236 [RVW02] O. Reingold，S. Vadhan，和 A. Wigderson. 熵波、之字形图积和新的常度扩张器。*Annals of Mathematics*，第 157–187 页，2002。引用次数 92，140 [RW89] R. Raz 和 A. Wigderson. 布尔关系的概率通信复杂性。在 *Proceedings of 30th annual IEEE Symposium on Foundations of Computer Science*，第 562–567 页。IEEE，1989。引用次数 167 [RW92] R. Raz 和 A. Wigderson. 匹配的单调电路需要线性深度。*J. ACM*，39:736–744，1992。引用次数 54，166，167

[RW00] S. Rudich 和 A. Wigderson，编者。*Computational complexity theory*，*IAS/Park-City Mathematics Series* 第 10 卷。高级研究所/美国数学会，2000。在 57 [Rys63] H.J. Ryser。组合数学。*Math. Assoc. America*，1963。Carus 数学专著，第 14 号。在 129 [SA90] H.D. Sherali 和 W.P. Adams。连续和凸包表示之间的松弛层次结构。*SIAM Journal on Discrete Mathe- matics*，3(3):411–430，1990。在 65 [San09] R. Santhanam。Merlin-Arthur 类的电路下界。*SIAM Journal on computing*，39(3):1038–1061，2009。在 48，56 [Sar90] P. Sarnak。*Some applications of modular forms*，第 99 卷。剑桥大学出版社，1990。在 91 [Sau72] N. Sauer。关于集合族的密度。*Journal of Combinatorial Theory, Series A*，13(1):145–147，1972。在 195 [Sav70] W.J. Savitch。非确定性 tape 复杂性与确定性 tape 复杂性之间的关系。*Journal of computer and system sciences*，4(2):177–192，1970。在 106，155 [Sch78] T.J. Schaefer。可满足性问题复杂性。在 *Proceedings of the 10th annual ACM symposium on Theory of Computing*，第 216–226 页。ACM，1978。在 41 [Sch80] J.T. Schwartz。用于验证多项式恒等式的快速概率算法。*Journal of the ACM*，27(4):701–717，1980。在 70 [Sch90] R.E. Schapire。弱学习能力的强度。*Machine learning*，5(2):197–227，1990。在 197，198 [Sch92] L.J. Schulman。关于噪声信道的通信：计算编码定理。在 *Proceedings of 33rd annual IEEE Symposium on Foundations of Computer Science*，第 724–733 页。IEEE，1992。在 176，254 [Sch93] L.J. Schulman。交互通信的确定性编码。在 *Proceedings of the 25th annual ACM symposium on Theory of Computing*，第 747–756 页。ACM，1993。在 176，254 [Sch96] L.J. Schulman。交互通信的编码。*IEEE Transactions on Information Theory*，42(6):1745–1756，1996。在 176，177 [Sch03] A. Schrijver。*Combinatorial Optimization: Polyhedra and Efficiency*。Springer-Verlag，柏林，2003。在 19 [Sch08] G. Schoenebeck。某些 k-CSP 的线性 Lasserre 下界。在 *Proceedings of 49th annual IEEE Symposium on Foundations of Computer Science*，第 593–602 页。IEEE，2008。在 66 [See04] N.C. Seeman。纳米技术与双螺旋。*Scientific American*，290(6):64–75，2004。在 244 [Seg07] N. Segerlind。命题证明的复杂性。*Bulletin of Symbolic Logic*，13(04):417– 481，2007。在 57 [Sel65] A. Selberg。关于模形式傅里叶系数的估计。在 *Proceedings of Symposia in Pure Mathematics*，第 8 卷，第 1–15 页，1965。在 91

[Ser03] Á. Seress. *Permutation group algorithms*，第152卷。剑桥大学出版社，2003年。在141处引用 [SF12] R.E. Schapire和Y. Freund. *Boosting: Foundations and Algorithms*。麻省理工学院出版社，2012年。在197处引用 [Sha48] C.E. Shannon. 通信数学理论。*Bell System Technical Journal*，27(3):379–423，1948年。在82、83、172、173、174、176、254处引用 [Sha49a] C.E. Shannon. 密码系统的通信理论。*Bell System Technical Journal*，28(4):656–715，1949年。在203、206处引用 [Sha49b] C.E. Shannon. 双端开关电路的综合。*Bell System Technical Journal*，28(1):59–98，1949年。在50处引用 [Sha79a] A. Shamir. 在$O(\log n)$算术步骤中分解数字。*Information Processing Letters*，8(1):28–31，1979年。在124处引用 [Sha79b] A. Shamir. 如何共享秘密。*Communications of the ACM*，22(11):612–613，1979年。在213处引用 [Sha83] A. Shamir. 关于生成密码学上强伪随机序列。*ACM Transactions on Computer Systems*，1(1):38–44，1983年。在77处引用 [Sha92] A. Shamir. IP = PSPACE。*Journal of the ACM*，39:869–877，1992年。在38、105、254处引用 [Sha99] Y. Shalom. 有界生成和Kazhdan的性质(T)。*Publications Mathématiques de l'IHÉS*，90:145–168，1999年。在92处引用 [Sha04] R. Shaltiel. 提取器显式构造的最近发展。*Current Trends in Theoretical Computer Science: The Challenge of the New Century*，1:229–264，2004年。在97处引用 [She72] S. Shelah. 组合问题；无穷语言中的模型和理论的稳定性和顺序。*Pacific Journal of Mathematics*，41(1):247–261，1972年。在195处引用 [She14] A.A. Sherstov. 通信复杂性理论：三十五年的集合不相交性。在*International symposium on Mathematical foundations of computer science*，第24–43页。Springer，2014年。在161处引用 [Sho88] N.Z. Shor. 获得多项式数学规划问题全局极值的方法。*Cybernetics*，23(5):695–700，1988年。在65处引用 [Sho94] P.W. Shor. 量子计算算法：离散对数和分解。在*Proceedings of 35th annual IEEE Symposium on Foundations of Computer Science*，第124–134页。IEEE，1994年。在115、137、149处引用 [Sho95] P.W. Shor. 降低量子计算机内存退相干性的方案。*Physical review A*，52(4):R2493，1995年。在117处引用 [Sim57] H.A. Simon. *Models of man; social and rational*。Wiley，1957年。在243处引用 [Sim70] C.C. Sims. 排列群研究中的计算方法。在*Computational problems in abstract algebra*，第169–183页，1970年。在20处引用 [Sim97] D.R. Simon. 关于量子计算的能力。*SIAM Journal on computing*，26(5):1474– 1483，1997年。在116处引用

[Sim10] D. Simon. 数论中LLL算法的应用选择。在 *The LLL Algorithm*，第265–282页。Springer，2010。引用于148 [Sin11] S. Singh.
The code book: the science of secrecy from ancient Egypt to quantum cryptography。Anchor，2011。引用于202 [Sip88] M. Sipser. 扩展器、随机性或时间与空间。*Journal of computer and system sciences*，36(3):379–383，1988。引用于98，100 [Sip92] M. Sipser. P与NP问题的历史和现状。在 *Proceedings of the 24th annual ACM symposium on Theory of Computing*，第603–618页。ACM，1992。引用于23 [Sip97] M. Sipser. *Introduction to the theory of computation*。PWS Publishing Co.，波士顿，马萨诸塞州，1997。引用于158 [SJ89] A. Sinclair和M. Jerrum. 近似计数、均匀生成和快速混合马尔可夫链。*Information and Computation*，82(1):93–133，1989。引用于90，143，237 [SK14] S. Saraf和M. Kumar。关于同质深度4算术电路的能力。`http://www.math.rutgers.edu/~ss1984/`，2014。引用于133 [Sly10] A. Sly。唯一性阈值处的计算转变。在 *Proceedings of 51st annual IEEE Symposium on Foundations of Computer Science*，第287–296页。IEEE，2010。引用于144，238 [Spe28] E. Sperner。关于维度数和区域不变性的新证明。在 *Abhandlungen aus dem Mathematischen Seminar der Universität Hamburg*，第6卷，第265–272页。Springer，1928。引用于226 [Spi71] P.M. Spira。布尔函数的时间-硬件复杂度权衡。在 *Proceedings of the 4th Hawaii symposium on system sciences*，第525–527页，1971。引用于166 [Spi95] D.A. Spielman。线性时间可编码和可解码的错误纠正码。在 *Proceedings of the 27th annual ACM symposium on Theory of Computing*，第388–397页。ACM，1995。引用于176，236 [SS71] A. Schönhage和V. Strassen。大数快速乘法。*Computing*，7:281– 292，1971。引用于51 [SS77] R.M. Solovay和V. Strassen。快速蒙特卡洛素性测试。*SIAM Journal on Computing*，6(1):84–85，1977。引用于71，87，136 [SS79] E. Shamir和M. Snir。关于公式的深度复杂性。*Mathematical Systems Theory*，13(1):301–322，1979。引用于132 [SS96] M. Sipser和D.A. Spielman。扩展器码。
IEEE Transactions on Information Theory，42(6):1710–1722，1996。引用于236 [SS12] D.A. Spielman和N. Srivastava。限制可逆性定理的简单证明。*Israel Journal of Mathematics*，190(1):83–91，2012。引用于138 [ST85] D.D. Sleator和R.E. Tarjan。列表更新和页面规则的摊销效率。*Commu- nications of the ACM*，28(2):202–208，1985。引用于179，181 [ST04a] D.A. Spielman和S. Teng。图划分、图稀疏化和求解线性系统的近似线性时间算法。在 *Proceedings of the 36th annual ACM symposium on Theory of Computing*，第81–90页。ACM，2004。引用于235

[ST04b] D.A. Spielman 和 S. Teng. 算法平滑分析：为什么单纯形算法通常需要多项式时间。*Journal of the ACM*，51(3)：385–463，2004。在43，235，259处引用 [ST11] D.A. Spielman 和 S. Teng. 图的谱稀疏化。*SIAM Journal on computing*，40(4)：981–1025，2011。在138处引用 [Ste74] G. Stengle. 半代数几何中的零点定理和正点定理。*Mathematis- che Annalen*，207(2)：87–97，1974。在65处引用 [STJ83] E. Szemerédi 和 W.T. Trotter Jr. 离散几何中的极值问题。*Combinatorica*，3(3-4)：381–392，1983。在138处引用 [Sto73] L.J. Stockmeyer. 平面3着色是多项式完备的。*ACM Sigact News*，5(3)：19–25，1973。在32，33处引用 [Sto76] L.J. Stockmeyer. 多项式时间层次。*Theoretical Computer Science*，3(1)：1–22，1976。在37，38处引用 [Sto10] J. Stothers. *On the Complexity of Matrix Multiplication*。博士论文，爱丁堡大学，2010。可在`http://www.maths.ed.ac.uk/pg/thesis/stothers.pdf`获取。在128处引用 [Str69] V. Strassen. 高斯消元不是最优的。*Numerische Mathematik*，13(4)：354–356，1969。在128处引用 [Str73a] V. Strassen. 元对称函数和插值系数的计算复杂性。¨*Numerische Mathematik*，20(3)：238–251，1973。在126处引用 [Str73b] V. Strassen. 避免除法。*Journal für die reine und angewandte Mathematik*，264：184–202，1973。在127，129处引用 [Str86] V. Strassen. Lesley G. Valiant 的工作。在*International Congress of Mathematicians*，第16页，1986。在35处引用 [Str87] V. Strassen. 相对双线性复杂性和矩阵乘法。*Journal für die reine und angewandte Mathematik*，375：406–443，1987。在150处引用 [Stu08] B. Sturmfels. *Algorithms in invariant theory*。Springer Science & Business Media，2008。在149处引用 [STV99] M. Sudan，L. Trevisan，和 S. Vadhan. 无xor引理的伪随机生成器。在 *Proceedings of the 31st annual ACM symposium on Theory of Computing*，第537–546页。ACM，1999。在236处引用 [Sub61] B.A. Subbotovskaya. 使用$+, \cdot, -$通过公式实现线性函数。*Doklady Akademii Nauk SSSR*，136(3)：553–555，1961。在51处引用 [Sud96] M. Sudan. *Efficient Checking of Polynomials and Proofs and the Hardness of Approximation Problems*。Springer-Verlag，1996。ACM杰出论文，计算机科学讲座笔记。在110处引用 [Sud97] M. Sudan. 超越纠错界限的Reed-Solomon码解码。*Journal of complexity*，13(1)：180–193，1997。在235处引用 [Sud00] M. Sudan. 列解码：算法和应用。*Theoretical Computer Science: Ex- ploring New Frontiers of Theoretical Informatics*，第25–41页，2000。在236处引用

[SV86] M. Santha 和 U. Vazirani. 从半随机源生成准随机序列。*Journal of Computer and System Sciences*，33(1):75–87，1986。被引用于98 [SVdB01] A. Schofield 和 M. Van den Bergh. 任意维向量下箭头的半不变量。*Indagationes Mathematicae*，12(1):125–138，2001。被引用于152 [SW73] D. Slepian 和 J. Wolf. 相关信息源的噪声编码。*IEEE Transactions on Information Theory*，19(4):471–480，1973。被引用于174 [SW01] A. Shpilka 和 A. Wigderson. 特征零域上的深度-3算术电路。*Computational complexity*，10(1):1–27，2001。被引用于126 [SW14] A. Sahai 和 B. Waters. 如何使用不可区分混淆：可否认加密等。在*Proceedings of the 46th annual ACM symposium on Theory of Computing*，第475–484页。ACM，2014。被引用于216 [Swa86] E.R. Swart. P=NP。*Report No. CIS86-02, Department of Computer and Information Science, University of Guelph, Ontario, Canada*，1986。被引用于169 [SY10] A. Shpilka 和 A. Yehudayoff. 算术电路：最近结果和开放问题的综述。*Foundations and Trends in Theoretical Computer Science*，5(3–4):207–388，2010。被引用于70，124，132，134 [SZ99] M. Saks 和 S. Zhou. $BP_HSPACE(S) \subseteq DSPACE(S^{3/2})$。*Journal of computer and system sciences*，58(2):376–403，1999。被引用于156 [SZ00] M. Saks 和 F. Zaharoglou. 无等待k集一致性问题不可能：公共知识拓扑。*SIAM Journal on computing*，29(5):1449–1483，2000。被引用于225，226，227，228 [Sze88] R. Szelepcsényi. 非确定性自动机的强制枚举法。*Acta Informatica*，26(3):279–284，1988。被引用于155 [Sze12] B. Szegedy. 关于高阶傅里叶分析。*arXiv preprint arXiv:1203.2260*，2012。被引用于96 [Tal14] A. Tal. 使用谱技术压缩德摩根公式。在*Proceedings of 55th annual IEEE Symposium on Foundations of Computer Science*，第551–560页。IEEE，2014。被引用于52 [Tan84] R.M. Tanner. 从广义N边形生成显式集中器。*SIAM Journal on Algebraic Discrete Methods*，5(3):287–293，1984。被引用于90 [Tao06] T. Tao. 结构与随机性的二分法，算术数列和素数。在*Proceedings of the International Congress of Mathematicians*，第581–608页，2006。被引用于93 [Tao08] T. Tao. *Structure and randomness: pages from year one of a mathematical blog*。美国数学会，普罗维登斯，RI，2008。被引用于93 [Tao09] T. Tao. 2009年Kakeya猜想研究进展。`http://terrytao.wordpress.com/2009/05/11/`。被引用于137 [Tar51] A. Tarski. *A decision method for elementary algebra and geometry*。加州大学出版社，1951。被引用于13 [Tar87] E. Tardos. 单调与非单调电路复杂度之间的差距是指数级的。*Combinatorica*，7(4):141–142，1987。被引用于54

[Tav13] S. Tavenas. 将深度降低到4和3的改进界限。在 *Mathematical founda- tions of computer science 2013*，第813–824页。Springer，2013。引用于133 [Tel18] R. Tell. 证明prBPP=prP与“几乎”证明P ≠ NP一样困难。*Electronic Colloquium on Computational Complexity (ECCC)*，25:3，2018。引用于81 [Tho87] Sir W. Thomson. 关于以最小分区面积划分空间。*Acta mathematica*，11(1-4):121–134，1887。引用于146 [Tho79] C. D. Thompson. VLSI的面积-时间复杂性。在 *Proceedings of the 11th annual ACM symposium on Theory of Computing*，第81–88页。ACM，1979。引用于164 [Tho80] C.D. Thompson. *A complexity theory for VLSI*。博士论文，卡内基梅隆大学，1980。引用于164 [Tho87] A. Thomason. 伪随机图。*North-Holland Mathematics Studies*，144:307–331，1987。引用于89 [TKRR13] Y. Tauman Kalai，R. Raz，和R.D. Rothblum. 有限空间的委托。在 *Proceedings of the 44th annual ACM symposium on Theory of Computing*，第565–574页。ACM，2013。引用于215 [Tod91] S. Toda. PP与多项式时间层次一样困难。*SIAM Journal on computing*，20(5):865–877，1991。引用于38，106 [TPC15] A. Tiwari，H.K. Patra，和J. Choi. *Advanced theranostic materials*。John Wiley & Sons，2015。引用于244 [Tra84] B.A. Trakhtenbrot. 俄罗斯对穷举（暴力）算法方法的综述。*Annals of the History of Computing*，6(4):384–400，1984。引用于23 [Tre99] L. Trevisan. 使用伪随机生成器构建提取器。在 *Proceedings of the 31st annual ACM symposium on Theory of Computing*，第141–148页。ACM，1999。引用于80，100，214 [Tsi93] B. Tsirelson. 量子贝尔型不等式。*Hadronic Journal Supplement*，8:329–345，1993。引用于123 [TSZ04] A. Ta-Shma和D. Zuckerman. 提取码。*IEEE Transactions on Information Theory*，50(12):3015–3025，2004。引用于97 [TT94] P. Tiwari和M. Tompa. Shamir和Snir关于单调电路深度的下界的一个直接版本。*Information Processing Letters*，49(5):243–248，1994。引用于132 [TTV09] L. Trevisan，M. Tulsiani，和S. Vadhan. 规则性、提升和高效模拟每个高熵分布。在 *Proceedings of the 24th annual IEEE conference on Computa- tional Complexity*，第126–136页。IEEE，2009。引用于95 [Tur36] A.M Turing. 关于可计算数，及其在Entscheidungsproblem中的应用。*Journal of Mathematics*，58(345-363):5，1936。引用于1，12 [Tur50] A.M. Turing. 计算机与智能。*Mind*，59(236):433–460，1950。引用于253 [Tur52] A.M. Turing. 形态发生的化学基础。*Philosophical Transactions of the Royal Society of London. Series B, Biological Sciences*，237(641):37–72，1952。引用于243

[TV00] L. Trevisan 和 S. Vadhan. 从可抽样分布中提取随机性。在 *Proceedings of 41st annual IEEE Symposium on Foundations of Computer Science*，第 32 页。IEEE，2000。引用于 102 [TV02] L. Trevisan 和 S. Vadhan. 通过均匀归约的伪随机性和平均情况复杂性。在 *Proceedings of the 17th annual IEEE conference on computational complexity*，第 129–138 页。IEEE，2002。引用于 81 [TZ08] T. Tao 和 T. Ziegler. 质数包含任意长的多项式级数。*Acta Mathematica*，201(2):213–305，2008。引用于 95 [Vad04] S.P. Vadhan. 在有限存储模型中构建局部可计算提取器和密码系统。*Journal of Cryptology*，17(1):43–77，2004。引用于 97 [Vad09] S. Vadhan. 随机提取器的密码学应用，2009。`http://people.seas.harvard.edu/~salil/research/extractors-clouds09.ppt`。引用于 97 [Vad11] S.P. Vadhan. 伪随机性。*Foundations and Trends in Theoretical Computer Science*，7(1–3):1–336，2011。引用于 70，77，97，102，137 [Vad17] S. Vadhan. 差分隐私的复杂性。在 *Tutorials on the foundations of Cryptography*，第 347–450 页。Springer，2017。`http://privacytools.seas.harvard.edu/publications/complexity-differential-privacy`。引用于 202 [Val79a] L.G. Valiant. 代数中的完备类。在 *Proceedings of the 11th annual ACM sym- posium on Theory of Computing*，第 249–261 页。ACM，1979。引用于 38，130，131 [Val79b] L.G. Valiant. 计算永久的复杂性。*Theoretical Computer Science*，8(2):189–201，1979。引用于 144，150，237 [Val79c] L.G. Valiant. 枚举和可靠性问题的复杂性。*SIAM Journal on computing*，8(3):410–421，1979。引用于 38，237 [Val80] L.G. Valiant. 否定可以指数级强大。*Theoretical Computer Science*，12(3):303–314，1980。引用于 132 [Val84a] L.G. Valiant. 大多数函数的短单调公式。*Journal of Algorithms*，5(3):363–366，1984。引用于 55，198 [Val84b] L.G. Valiant. 可学习理论。*Communications of the ACM*，27(11):1134–1142，1984。引用于 192，193，195 [Val00] L.G. Valiant. *Circuits of the Mind*。牛津大学出版社按需，2000。引用于 36，247 [Val06] L.G. Valiant. 神经计算的定量理论。*Biological cybernetics*，95(3):205– 211，2006。引用于 247 [Val09] L.G. Valiant. 可进化性。*Journal of the ACM*，56(1):3，2009。引用于 245 [Val12] L.G. Valiant. 海马体作为皮质稳定内存分配器的理论。*Neural computation*，24(11):2873–2899，2012。引用于 247 [Val13] L.G. Valiant. *Probably Approximately Correct: Nature's Algorithms for Learning and Pros- pering in a Complex World*。基础书籍，2013。引用于 36，192，245 [Val14] L.G. Valiant. 全球皮质理论必须解释什么? *Current opinion in neurobiology*，25:15–19，2014。引用于 247

[Vap98] V. Vapnik. *Statistical learning theory*，第1卷。Wiley，1998。引用于192 [Vap13] V. Vapnik. *The nature of statistical learning theory*。Springer Science & Business Media，2013。引用于192 [Var57] R.R. Varshamov. 在纠错码中错误信号的估计。在 *Doklady Akademia Nauk SSSR*，第117卷，第739–741页，1957。俄语。英文翻译在 I. F. Blake，代数编码理论：历史与发展，Dowden，Hutchinson and Ross，1973，第68–71页。引用于83，176 [Vav09] S.A. Vavasis. 非负矩阵分解的复杂性。*SIAM Journal on Optimization*，20(3):1364–1377，2009。引用于170 [VC74] V.N. Vapnik 和 A.J. Chervonenkis. *Theory of pattern recognition*。Nauka，1974。引用于192 [VC15] V.N. Vapnik 和 A.Y. Chervonenkis. 事件相对频率收敛到其概率的均匀性。在 *Measures of Complexity*，第11–30页。Springer，2015。引用于192，193，194，195 [Vem05] S. Vempala. 几何随机游走：综述。*Combinatorial and computational geometry*，52(573–612):2，2005。引用于144 [Vin04] N.V. Vinodchandran. $\mathrm{AM}_{\exp} \not\subseteq (\mathrm{NP} \cap \mathrm{coNP})/\mathrm{poly}$。*Information Processing Letters*，89:43–47，2004。引用于48，56 [Vio15] E. Viola. 加法通信复杂性。*Combinatorica*，35(6):703–747，2015。引用于163 [vN28] J. von Neumann. 关于社会游戏理论。*Mathematische Annalen*，100(1):295–320，1928。引用于163 [vN51] J. von Neumann. 与随机数字相关的各种技术。*Applied Math Series*，12(36-38):1，1951。引用于77 [VN58] J. Von Neumann. *The computer and the brain*。耶鲁大学出版社，1958。2012年再版，附雷·库兹韦尔序言。引用于243，247 [VNB+66] J. Von Neumann，A.W. Burks，等。自我复制自动机理论。*IEEE Transactions on Neural Networks*，5(1):3–14，1966。引用于243 [VSBR83] L.G. Valiant，S. Skyum，S. Berkowitz，和 C. Rackoff。使用少量处理器的多项式快速并行计算。*SIAM Journal on computing*，12(4):641–644，1983。引用于126 [VV85] U. Vazirani 和 V. Vazirani。随机多项式时间等于略随机多项式时间。在 *Proceedings of 26th annual IEEE Symposium on Foundations of Computer Science*，第417–428页。IEEE，1985。引用于98 [VW18] E. Viola 和 A. Wigderson。局部扩张器。*Computational Complexity*，27(2):225–244，2018。引用于91 [vzGG13] J. von zur Gathen 和 J. Gerhard. *Modern computer algebra*。剑桥大学出版社，2013。引用于124 [Wei49] A. Weil. 有限域中方程的解的数目。*Bulletin Amer. Math. Soc.*，55(5):497–508，1949。引用于84，85 [Wei06] D. Weitz. 最多达到树阈值的独立集计数。在 *Proceedings of the 38th annual ACM symposium on Theory of Computing*，第140–149页。ACM，2006。引用于144，238

[Wer74] P.J. Werbos. 超越回归：行为科学预测和分析的新工具。*Ph.D. dissertation, Harvard University*，1974。引用于127 [Wer94] P.J. Werbos。
The roots of backpropagation: from ordered derivatives to neural networks and political forecasting，第1卷。John Wiley & Sons，1994。引用于127 [Whi92] S.R. White。量子重整化群的密度矩阵公式。
Physical Review Letters，69(19):2863，1992。引用于120 [Wig60] E.P. Wigner。数学在自然科学中的不合理有效性。*Com- munications on pure and applied mathematics*，13(1):1–14，1960。理查德·库尔特数学科学讲座，纽约大学，1959年5月11日。引用于242 [Wig93] A. Wigderson。电路复杂度下界融合方法。
Combinatorics, Paul Erdos is Eighty，1:453–468，1993。引用于41，54，56 [Wig06] A. Wigderson。P, NP和数学——计算复杂性的视角。在*Proceed- ings of the 2006 International Congress of Mathematicians*中，2006。引用于xiii [Wig09] A. Wigderson。随机提取器——应用和构造。在*LIPIcs-Leibniz International Proceedings in Informatics*，第4卷。Schloss Dagstuhl − Leibniz-Zentrum für Informatik，2009。引用于97 [Wig10a] A. Wigderson。扩展图——应用和组合构造，2010年6月14–18日。3小时教程，数学结构伪随机性研讨会。可在`http://www.math.ias.edu/~avi/TALKS/Expander_tutorial_2010.ppt`获取。引用于91 [Wig10b] A. Wigderson。数学中的哥德尔现象：现代观点。*Kurt Gödel and the Foundations of Mathematics: Horizons of Truth*，2010。引用于23 [Wil75] M.V. Wilkes。
Time sharing computer systems。Elsevier Science Inc.，1975。引用于44 [Wil14] R. Williams。非均匀ACC电路下界。*Journal of the ACM*，61(1):2，2014。引用于56 [Wil15] V.V. Williams。简单问题的困难性：基于流行的猜想，如强指数时间假设。在*Proceedings of the International symposium on Parameterized and Exact Computation*中，第16–28页，2015。引用于235 [Wil18] V.V. Williams。关于算法和复杂性中的一些精细问题。在*Proceedings of the International Congress of Mathematicians (ICM)*中，2018。引用于51 [Win06] E. Winfree。自修复瓦片集。*Nanotechnology: science and computation*，第55–78页，2006。引用于244 [Wol99] T. Wolff。与Kakeya问题相关的研究。*Prospects in mathematics*，2:129– 162，1999。引用于137 [Yan91] M. Yannakakis。用线性规划表达组合优化问题。*Journal of computer and system sciences*，43(3):441–466，1991。引用于169，170，235 [Yao77] A.C. Yao。概率计算：向统一复杂度度量迈进。在*Proceedings of 18th annual IEEE Symposium on Foundations of Computer Science*中，第222–227页。IEEE，1977。引用于163 [Yao79] A.C. Yao。与分布式计算相关的某些复杂性问题（初步报告）。在*Proceedings of the 11th annual ACM symposium on Theory of Computing*中，第209–213页。ACM，1979。引用于161

[Yao82a] A.C. Yao. 安全计算协议。在 *Proceedings of 23rd annual IEEE Symposium on Foundations of Computer Science*，第 160–164 页。IEEE，1982。引用于 211 [Yao82b] A.C. Yao. 陷门函数的理论与应用。在 *Proceedings of 23rd annual IEEE Symposium on Foundations of Computer Science*，第 80–91 页。IEEE，1982。引用于 73，74，77，78，207 [Yao86] A.C. Yao. 如何生成和交换秘密。在 *Proceedings of 27th annual IEEE Sym- posium on Foundations of Computer Science*，第 162–167 页。IEEE，1986。引用于 46，108，209，210，211，213，251 [Yao93] A.C. Yao. 量子电路复杂性。在 *Proceedings of 34th annual IEEE Symposium on Foundations of Computer Science*，第 352–361 页。IEEE，1993。引用于 114 [Yeh11] A. Yehudayoff. 素域上的仿射提取器。*Combinatorica*，31(2):245，2011。引用于 102 [Yek12] S. Yekhanin. 局部可解码码。*Foundations and Trends in Theoretical Computer science*，6(3):139–255，2012。引用于 112，236 [Zhu17] D. Zhuk. CSP 二分猜想证明。在 *Proceedings of 58th annual IEEE Symposium on Foundations of Computer Science*，第 331–342 页。IEEE，2017。引用于 41 [Zip79] R.E. Zippel. 稀疏多项式的概率算法。*Symbolic and algebraic computation (EUROSCAM '79), Lecture Notes in Computer Science*，72:216–226，1979。引用于 70 [ZL78] J. Ziv 和 A. Lempel。通过可变速率编码压缩单个序列。*IEEE Transactions on Information Theory*，24(5):530–536，1978。引用于 97 [Zuc90] D. Zuckerman。一般弱随机源。在 *Proceedings of 31st Annual IEEE Symposium on Foundations of Computer Science*，第 534–543 页。IEEE 计算机协会，1990。引用于 98，100 [Zuc91] D. Zuckerman。*Computing Efficiently Using General Weak Random Sources*。博士论文，加州大学伯克利分校，1991。引用于 100 [Zuc97] D. Zuckerman。随机最优无记忆采样。*Random Structures and Algorithms*，11(4):345–367，1997。引用于 102 [Zuc06] D. Zuckerman。线性度提取器与最大团和色数的不近似性。在 *Proceedings of the 38th annual ACM symposium on Theory of Computing*，第 681–690 页。ACM，2006。引用于 97