arXiv:2304.14700v1 [cs.CC] 28 Apr 2023

半比特拉伸与抗非确定性攻击的伪随机性

Iddo Tzameret* Lu-MingZhang†伦敦帝国学院 伦敦政治经济学院

摘要

我们将密码学中抗非确定性攻击的伪随机性理论推进到超越 Rudich 原始工作的程度 [Rud97], , 并将其用于推导平均情形复杂性和证明复杂性中的新结果。具体而言，我们证明了以下结论：

半比特扩展：超比特和半比特是密码学伪随机数生成器的变体，能够抵御非确定性统计检验[Rud97]。它们最初被提出，是为了排除某些证明强复杂性下界的方法，这些下界超出了 Rudich 和 Razborov 提出的自然证明障碍所设定的限制 [RR97]。自半比特提出以来，它们能否扩展一直是一个开放问题。我们通过证明以下结论，肯定地回答了这个问题：每个半比特 b：$\{0,1\}^n \to \{0,1\}^{n+1}$ 都可以扩展为次线性数量的半比特 b'： $\{0,1\}^n \to \{0,1\}^{n+n^c}$，对每个常数 $0 < c < 1$。

平均情形困难性：利用 Santhanam 的工作 [San20], 我们将结果应用于新的平均情形柯尔莫哥洛夫复杂度结果：我们证明，$\mathrm{K}^{\mathsf{poly}}[n - O(1)]$ 是针对 NP/poly 机器的零错误平均情形困难，当且仅当 $\mathrm{K}^{\mathsf{poly}}[n - o(n)]$ 也是如此，其中对于函数 $s(n)$：$\mathbb{N} \to \mathbb{N}$，$\mathrm{K}^{\mathsf{poly}}[s(n)]$ 表示所有字符串组成的语言 $x \in \{0,1\}^n$ 对于这些字符串，存在描述长度至多为 $s(n)$ 的（固定）多项式时间图灵机能够输出 x。

用非确定性不可预测性刻画超比特：在确定性情形下，Yao [Yao82] 证明了伪随机生成器的超多项式困难性等价于“下一比特”不可预测性。不可预测性大致是指，给定随机字符串的任意一个真前缀，预测下一个比特是不可行的。我们首次研究了确定性情形之外（在密码学情形中）的不可预测性，并从不可预测性的角度刻画生成器的非确定性困难性。具体而言，我们提出了四种更强的不可预测性概念：NP/poly-不可预测性，coNP/poly-不可预测性， $\cap$-不可预测性和 $\cup$-不可预测性，并证明生成器的超多项式非确定性困难性介于 $\cap$-不可预测性与 $\cup$-不可预测性之间。

用非确定性硬核谓词刻画超比特：我们引入硬核谓词的一种非确定性变体，称为超核心谓词。我们证明，超比特的存在等价于某个非收缩

*计算机系。本工作的部分内容是在加州大学伯克利分校的西蒙斯计算理论研究所完成的。本项目的部分经费由欧洲研究委员会（ERC）通过欧盟地平线2020研究与创新计划资助（资助协议编号：101002742）。电子邮箱：iddo.tzameret@gmail.com

†本工作的部分内容是在伦敦帝国学院期间完成的。电子邮箱：l534zhang@uwaterloo.ca

函数的超核心的存在。这类似于强伪随机生成器的存在与某个单向函数的硬核存在之间的等价关系 [GL89，HILL99], 并首次提供了超比特的另一种刻画。我们还证明，某一类可能具有硬核的函数，不可能拥有任何超核心。

目录

1 引言 **3**
1.1 非确定性安全伪随机性的理论.31.2 与障碍结果的关系.5

2 贡献、意义和背景 **6**
2.1 半比特拉伸.72.1.1 半比特拉伸对障碍结果的讨论及其意义 . . . . . 72.1.2 在平均情形复杂性中的应用.72.1.3 在证明复杂性中的应用.82.1.4 技术概述.92.2 基于不可预测性的非确定性安全精细刻画 102.3 超核心：非确定性情形下的硬核谓词.112.4 半比特存在性及 PAC 学习的论证.13

3 预备知识与基本概念 **13**
3.1 记号与约定.133.2 计算模型.143.3 自然证明.153.4 伪随机生成器.153.5 超比特与半比特.163.6 无穷多次超比特与半比特.18

4 半比特拉伸 **19**
4.1 平均情形复杂性中的应用.224.2 在证明复杂性生成器中的应用.24

5 非确定性可预测性 **24**
5.1 基本确定性可预测性.255.2 非确定性变体.25

6 超核心谓词 **29**
6.1 单向函数和硬核谓词.296.2 非确定性变体.31

7 结论和未来方向 **36**

A 半比特存在，除非小型电路的 PAC 学习是可行的 **36**
A.1 PAC 学习.36A.2 基于不存在性假设的学习.37

1 引言

伪随机性是一个自然概念，用于衡量资源受限计算机器识别真正随机源的能力。它是算法中的一个重要概念，可以通过用更少的随机比特模拟大量真正的随机比特，对高效概率算法进行去随机化。伪随机性的另一个重要方面涉及计算复杂性和密码学，尤其是计算下界；它是许多密码学结果、困难性与随机性的权衡，以及若干证明强计算下界之障碍的基础。在这里，我们主要关注障碍结果的后一个方面。

1.1 非确定性安全伪随机性的理论 s

Razborov 和 Rudich 在其《自然证明》论文中，建立了伪随机性与证明布尔电路下界的能力之间的联系 [RR97]。他们指出，大多数电路复杂性中的下界论证都包含（可能是隐含地包含）一种用于判定困难性的高效算法，具体来说：给定一个布尔函数的真值表，该算法会判断该函数是否具有某些（组合）性质，而这些性质意味着它对于某个给定的电路类而言是困难的（他们将下界论证的这种特性称为“构造性”和“有用性”）。他们还指出，这种算法能正确识别非可忽略比例的函数的困难性（在 [RR97]中，这被称为“大性”）。另一方面，若存在这样一种能够判定布尔函数（针对一般电路）困难性的高效算法，就会与伪随机性方面的合理假设相矛盾，即强伪随机生成器的存在性。这可以说为使用自然证明证明下界的能力设置了一道障碍。

自然证明的概念对计算复杂性理论产生了深远影响。然而，某些类别的自然证明可能不存在这一事实，会阻碍非常构造性的下界论证（也就是说，那些隐含地包含一个高效算法、能够判断一个函数何时具有高复杂度的论证），这一点不那么令人满意。人们希望将这种障碍扩展到构造性较弱的证明，例如那些论证中只隐含包含函数难度的短见证的证明。

事实上，自然证明这一概念旨在解释证明下界的困难，而非高效判定布尔函数的难度。因此，除其他原因外，Rudich[Rud97] 着手扩展自然证明障碍，使其涵盖非构造性论证，也就是隐含地使用布尔函数难度的高效见证的论证，以区别于确定性算法。为此，他扩展了伪随机生成器的概念，使其能够抵御非确定性对手。

尽管从经验来看，已知的大多数下界证明至少隐含地都属于P/poly-构造性自然证明的范畴，但完全可以设想并且合乎情理地假定，某些下界方法需要NP/poly-构造性自然证明。事实上，找到这样一种NP/poly-构造性下界证明方法，本身就是一个非常有趣的开放问题。此外，在证明复杂性的研究中，非确定性安全伪随机性的作用十分重要（参见 Pich 和 Santhanam 的近期工作 [PS 19], 以及 Kraj´ıˇcek [Kra04]）。还要注意，在讨论障碍这一概念时，我们指的是不可能性结果，因此不能总是期待能找到我们希望排除的证明方法的恰当实例。

因此，为了扩展自然证明障碍，Rudich 引入了两种原语：超比特、

及其较弱的变体半比特。超比特和半比特是强伪随机生成器（PRG）的（非一致）非确定性变体。它们能够抵御非确定性对手（即 NP/poly 中的对手）。要破解半比特，非确定性对手必须达到比破解超比特更强的意义，因此，半比特的存在比超比特的存在构成了更好的（即更弱的）假设，可用于建立障碍结果；这也是半比特这一概念重要的原因之一。

更具体地说，超比特和半比特都要求非确定性对手通过认证真正随机字符串（即，对手认为给定字符串是真正随机过程的输出时输出 1，认为它是伪随机生成器的输出时输出 0），从而有意义地区分真正随机字符串和伪随机字符串。因此，非确定性区分器不能仅仅通过猜测生成器的种子来破解超比特或半比特。具体而言，这一点通过以下方式得到保证：对于半比特，我们要求伪随机生成器像中的字符串始终被拒绝；而对于超比特，我们允许接受其中一些伪随机字符串，但要求被接受的、位于伪随机生成器像之外的字符串数量远多于像之中的字符串。

形式化地，我们有如下定义（除非另有说明，本文讨论的所有分布默认都是均匀分布，且 U_n 表示 $\{0,1\}^n$ 上的均匀分布）：

定义 1.1 （非确定性困难性 [Rud97]）。 设 g_n： $\{0,1\}^n \to \{0,1\}^{l(n)}$，其中 l（n） $> n$，是一个属于 P复杂度类/多项式的函数。我们称这样的函数为生成器。 非确定性困难性H_{nh}（g_n） （也称为 超硬度）是 g_n 的最小值 s，使得存在一个大小至多为 D 的非确定性电路 s，满足

$$\mathop{\mathbb{P}}_{y\in\{0,1\}^{l(n)}}[D(y)=1] - \mathop{\mathbb{P}}_{x\in\{0,1\}^{n}}[D(g_n(x))=1] \geq \frac{1}{s}. \tag{1}$$

与标准的（确定性）困难性定义（定义 3.6）不同，左侧两种可能性的排列顺序至关重要。这个顺序迫使非确定性区分器证明给定输入具有随机性。颠倒顺序，或在左侧添加绝对值，都会使破坏 g的任务变得轻而易举（如同标准伪随机生成器的情形）：区分器D 只需猜测一个种子 x 并检查 g（x）是否等于给定输入。对于这样的 D，我们有$\mathbb{P}$[D（g（x））$= 1] = 1$ 且 $\mathbb{P}$[D（y）$= 1] \leq 1/2$。

超比特是指数级超难生成器：

定义 1.2 (超比特 [Rud97])。 生成器 $g\colon \{0,1\}^n \to \{0,1\}^{n+c}$ (可由 P/poly 计算)，对于某个 $c\colon \mathbb{N}\to\mathbb{N}$，若称其为 c 超比特 (或 c-超比特)，则当且仅当 $H_{\text{nh}}(g) \geq 2^{n^\varepsilon}$ 对某个常数$\varepsilon > 0$ 以及所有足够大的 n 均成立。特别地，若 $c = 1$，则称 g 其为超比特。

许多针对确定性机器的强伪随机生成器候选方案，都是通过利用被猜想为单向的函数和／或其硬核构造出来的（例如参见 [ACGS88、Kal05、IN89]）。文献 [ACGS88] 提出了基于整数分解困难性假设的伪随机生成器，而 [Kal05] 提出了基于离散对数问题被猜想为困难这一假设的伪随机生成器。[IN89] 提出了基于子集和问题的伪随机生成器。类似地，对于非确定性安全的伪随机生成器（即超比特），Rudich 猜想存在一种基于子集和问题困难性的超比特生成器 [IN89]。我们不知道还有其他被猜想的超比特构造。

如上所述，Rudich 引入了生成器的另一种困难性度量：

定义 1.3 (半硬度 [Rud97])。 设 g_n : $\{0,1\}^n \to \{0,1\}^{l(n)}$ 是一个生成器（可由 P/poly计算）。那么，半硬度 $H_{\mathrm{dh}}(g_n)$ 是 g_n 的最小值，满足存在一个大小至多为 s 的非确定性电路 D，使得其大小至多为 s，并且满足

$$\mathop{\mathbb{P}}_{y\in\{0,1\}^{l(n)}}[D(y)=1] \geq \frac{1}{s} \quad and \quad \mathop{\mathbb{P}}_{x\in\{0,1\}^{n}}[D(g_n(x))=1] = 0. \tag{2}$$

我们注意到，(2) 要求区分器对生成的字符串不能出错，因此比 (1) 的要求更强。因此， $H_{\mathrm{nh}}(g) \leq H_{\mathrm{dh}}(g)$ 对每个生成器 g 都成立。

半比特是指数级半硬生成器，其中“半”指的是“一半”：

定义 1.4 (半比特 [Rud97])。 g : $\{0,1\}^n \to \{0,1\}^{n+c}$ 对于某个 c : $\mathbb{N} \to \mathbb{N}$ 称为 c半比特 (或称 c-半比特)，如果 $H_{\mathrm{dh}}(g) \geq 2^{n^{\varepsilon}}$ 对某个 $\varepsilon > 0$ 和所有足够大的 n 都成立。特别地，如果 $c = 1$，我们称 g 为一个半比特。

值得一提的是，按照 g 的半比特，如 定义 1.4 所述，可以视为一个 击中集生成器，用于对抗 NP/poly (见下文 第2.1.2节 以及 Santhanam [San20])。

超比特与半比特的区别在于，半比特要求区分器以更强的意义破解它们：半比特区分器必须始终正确识别伪随机字符串（即，对于生成器像中的字符串，始终输出 0）。因此，如果 g 是超比特（此处的复数表示 g 的扩展长度可能大于 1；如果扩展长度恰好为 1，我们称 g 是一个超比特），那么NP/poly 中的任何算法都无法以较弱的意义破解 g （1），因此NP/poly 中的任何算法都无法以较强的意义破解 g （2），这意味着 g 也是半比特。因此，超比特的存在意味着半比特的存在（尽管这两者是否存在仍未有定论）。

注释 (密码学情形与复杂性理论情形)。 在本文中，我们只关注伪随机性的密码学情形。在这种情形下，生成器试图欺骗的对手可以比生成器更强大，具体而言，它拥有足够的计算资源来运行生成器。在复杂性理论情形下，对手无法模拟生成器，非确定性安全伪随机性的概念是在例如 Klivans 和 van Melkebeek [KvM02] 以及 Shaltiel 和 Umans [SU05] 的研究中提出的（另见 Sdroievski 和 van Melkebeek 的近期工作 [SvM23] 及其中的参考文献）。这些复杂性理论思想也在密码学中找到了多种应用（源自 Barak、Ong 和 Vadhan 的工作 [BOV07]）。还值得一提的是，在复杂性理论情形下，即使面对非确定性对手，也可以使用伪随机生成器困难性的原始定义（定义 3.6）；而在密码学情形下则不然，因为符合定义 3.6的伪随机生成器永远无法抵御猜测种子的非确定性对手。

1.2 与障碍结果的关系

回顾一下，自然证明 [RR97] 用于证明电路下界，是指使用布尔函数某种自然组合性质的证明。布尔函数的组合性质（简称性质）C 是一个布尔函数集合。我们称函数 f 具有性质 C，如果 f 属于C。设 Γ 和 Λ 是复杂性类， F_n 是所有 f 的集合： $\{0,1\}^n \to \{0,1\}$。如果子集 $C' \subseteq C$ 满足构造性（即可以在 Γ时间内判定 f 是否属于 C'）和大性（即 C' 占据了 F_n中不可忽略的一部分），则称 C 是Γ-自然的。若 C 能够用于对付 Λ

如果每个无穷多次具有性质 C 的函数族 f 都无法在 Λ中计算。自然证明的思路是：如果我们想证明某个函数族 f （例如布尔可满足性问题 SAT）不属于 P/poly（或一般地，不属于其他某个复杂性类 Λ），就找出 f 的某种自然组合性质 C，并证明所有具有性质 C 的函数族都无法在 P/poly 中计算（也就是说，该性质可用于对付 P/poly）。如果 f 是 NP 完全（例如 SAT），那么这样的证明就能得出 P $\neq$NP。

Razborov 和 Rudich 指出，基于强伪随机生成器的存在，不存在任何 P/poly-自然证明能够用于对付 P/poly。他们证明，许多已知的针对（非单调）布尔电路的下界证明都是自然证明，或者可以通过某种方式表述为自然证明。

在这一背景下，非确定性安全生成器理论可以排除一类更大的（或许也更自然的）下界论证：

˜定理 1.5 ([Rud97])。 如果超比特存在，那么就不存在任何 $NP/qpoly$-自然性质能够用˜于对付 P/poly，其中 $NP/qpoly$ 表示由非一致、准多项式规模电路族识别的语言类。

该定理基于如下能力得以证明：能够对超比特进行扩展，也就是说，将一个把 n 比特映射为 $n+1$ 比特的生成器（我们称之为扩展长度为 1）扩展为一个把 n比特映射为 $n+N$比特的生成器，其中 $N>1$，我们称其扩展长度为 N。在标准的伪随机理论中，已证明困难比特（即扩展长度为 1 的强伪随机生成器）可以扩展为多项式数量的困难比特（即扩展长度为多项式） [BM84、Yao82] ，并且可以用来构造难以攻破的伪随机函数生成器（粗略地说，这类生成器产生的伪随机函数与真正随机函数不可区分） [GGM86]。和困难比特一样，超比特也可以使用类似的拉伸算法扩展为多项式数量的超比特，并构造出对非确定性对手安全的伪随机函数生成器 [Rud97]。这类拉伸算法正确性的证明基于一种称为混合论证[GM84] 的技术，下面将对此进行回顾。相比之下，在本研究之前，人们尚不清楚半比特能否扩展为哪怕两个半比特，因为直接应用标准混合论证也无法得出这一结论。

2 贡献、意义与背景

我们奠定了抗非确定性攻击的伪随机性的基础。据我们所知，这是首次系统研究（密码学情形下的）非确定性伪随机性，并以 Rudich 提出的原语为基础 [Rud97]。我们进一步理解了该理论的基本原语，即超比特和半比特，并引入了新的概念，说明它们与已有概念之间的关系。我们由此在平均情形复杂性和证明复杂性方面得出若干结论。我们还在为障碍结果奠定更可靠的基础方面取得了一些初步进展：例如，通过证明半比特可以（适度地）扩展，我们为加强半比特与不可证明性结果之间的联系带来了一线希望（目前已知的只有超比特的存在能够导出障碍结果，而我们希望证明，较弱的半比特存在性假设也足以导出障碍结果）。

2.1 半比特拉伸

在 半比特拉伸算法 4.1中，我们给出了一种算法，能够对任意给定的半比特实现次线性拉伸。这解决了半比特能否扩展为 2 比特这一开放问题 [Rud97，开放问题 2] (另见 Santhanam [San20，问题 4])。

定理 (非正式；定理 4.2)。 每个半比特 $b:\ \{0,1\}^n \to \{0,1\}^{n+1}$ 都可以高效地转换（拉伸）为半比特 $g:\ \{0,1\}^n \to \{0,1\}^{n+n^c}$，对每个常数 $0 < c < 1$。

2.1.1 将半比特拉伸至障碍结果的讨论及其意义

半比特拉伸可以视为迈向证明以下结论的第一步：半比特˜的存在排除了对 P/poly 有用的 $NP/qpoly$-自然性质，具体解释如下。基于半比特的存在性，为 $NP/qpoly$-自然性质提供这样的障碍十分重要，因为假设半比特存在，比假设超比特存在更弱。

为什么说半比特拉伸是迈向证明半比特的存在会排除以下性质的第一步？˜排除 $NP/qpoly$对 P/poly 有用的自然性质？原因在于，拉伸是基于超比特的存在性建立障碍结果这一论证的第一步，具体来说：超比特的存在蕴含障碍结果，因为可以拉伸超比特以获得伪随机函数生成器，由此便能得到障碍结果，正如上文的 定理 1.5（以及其后的文字）所述。更确切地说，拉伸半比特是迈向 Rudich 的开放问题 3 的第一步（也是必要的一步），而该问题也蕴含 Rudich 的开放问题 4：

开放问题 (Rudich 的开放问题 3 [Rud97])。 给定一个半比特，能否构造出一个具有指数级（2^{n^ε}）半硬度的伪随机函数生成器?

开放问题 (Rudich 的开放问题 4 [Rud97])。 半比特的存在是否排除了˜$NP/qpoly$对 P/poly 有用的自然性质?

此外，对半比特可拉伸性的研究为解决 Rudich 的开放问题 1 提供了一个视角（若该问题得到肯定回答，开放问题 4 也将得到肯定解决）：

开放问题 (Rudich 的开放问题 1 [Rud97])。 半比特的存在是否意味着超比特也存在?

这是因为超比特的可拉伸性已得到充分理解，而此前我们对半比特的可拉伸性一无所知。我们预期，进一步理解半比特的基本性质（如可拉伸性），将有助于阐明半比特存在与超比特存在之间的关系（开放问题 1）。

2.1.2 平均情形复杂性中的应用

这里我们介绍定理 4.2 在时间受限柯尔莫戈洛夫复杂度的平均情形困难性方面的一项应用。

正如 Santhanam 所指出的 [San20], 一个击中集生成器 $g:\ \{0,1\}^n \to \{0,1\}^{n+1}$ 存在，当且仅当存在一个半比特 $b:\ \{0,1\}^n \to \{0,1\}^{n+1}$。回顾一下，针对一类判定问题 的击中集生成器 $\mathcal{C} \subseteq 2^{\{0,1\}^N}$是一个函数 $g:\ \{0,1\}^n \to \{0,1\}^N$，对于 $n < N$，满足

g 的像与每个足够稠密的集合 A 相交（即相交于） $\mathcal{C}$ （也就是说， $|A| \geq \frac{2^N}{N^{O(1)}}$ ）。并且我们有：

命题 (命题 4.3； [San20])。 设 $n < N$。存在一个击中集生成器 g： $\{0,1\}^n \to \{0,1\}^N$可在类 $\mathcal{D}$ 下针对 NP/poly 计算，当且仅当存在一个半比特 b： $\{0,1\}^n \to \{0,1\}^N$可在 $\mathcal{D}$ (针对 NP/poly).

Santhanam [San20, 命题 3] 建立了（简洁）击中集生成器与最小电路规模问题的平均情形困难性之间的等价关系，其中 MCSP 指的是 最小电路规模问题。然而，正如 Santhanam [San22], 向我们指出的那样，类似的论证可以证明，击中集生成器（不一定是简洁的）与针对 NP/poly 机器的多项式时间有界柯尔莫哥洛夫复杂度零错误平均情形困难性之间也存在等价关系，本文将对此进行证明。

我们将 t-受限字符串 x的柯尔莫哥洛夫复杂度，记作 $\mathrm{K}^t(x)$，定义为字符串 D 的最小长度；通用图灵机 $U(D)$（我们固定选取这样一台通用机）在至多 t 的时间内运行并输出 x。关于时间受限柯尔莫戈洛夫复杂度的更多细节，参见 [ABK+06] ；其中定义 9 给出了字符串的时间受限柯尔莫戈洛夫复杂度定义（该定义实际上是在将索引 i和 x作为 i 输入给 D 时，输出字符串的第 U位，但这不会改变我们的结果）。

定义 2.1（语言 $\mathrm{K}^t[s]$ 和 $\mathrm{K}^{\mathrm{poly}}[s(n)]$）。对于时间函数 $t(n)$： $\mathbb{N} \to \mathbb{N}$ 和规模函数 $s(n)$： $\mathbb{N} \to \mathbb{N}$，满足 $s(n) \leq n$，令 $\mathrm{K}^{t(n)}[s(n)]$ 为语言 $\{x \in \{0,1\}^*: |X| = n \wedge \mathrm{K}^{t(n)}(x) \leq s(n)\}$。我们将 $\mathrm{K}^{\mathrm{poly}}[s(n)]$ 定义为语言 $\bigcup_{c\in\mathbb{N}} \mathrm{K}^{n^c}[s(n)]$。

我们还需要定义相对于类 NP/poly 的零错误平均情形困难性概念（见 定义 4.4）。非正式地说，对于语言 L ，若要称其对于 容易，NP/poly，就应存在一台带辅助信息的非确定性多项式时间机器，使其在给定输入 x 时，猜测 $x \in L$ 的一个见证或 $x \notin L$的一个见证，并在找到见证后给出相应回答；此外，我们假设，对于多项式小比例的输入，存在这样的见证（用于判定是否属于 L）。若未找到见证，机器就输出“无法判定”。（我们还假设，不存在同时对应于 $x \in L$和 $x \notin L$ 的矛盾见证对。）

在第 4.1 节中，我们证明如下结论：

定理（平均情形时间受限柯尔莫哥洛夫复杂度的等价性；定理 4.5）。$\mathrm{K}^{\mathrm{poly}}[n - O(1)]$ 相对于NP/多项式时间机器具有零错误平均情形困难性，当且仅当$\mathrm{K}^{\mathrm{poly}}[n - o(n)]$ 相对于NP/多项式时间机器具有零错误平均情形困难性。

2.1.3 证明复杂性中的应用

在证明复杂性领域，Krajíček [Kra04、Kra10] 以及 Alekhnovich、Ben-Sasson、Razborov 和 Wigder- son [ABRW04] 发展了证明复杂性生成器理论。给定一个 P/poly 映射g： $\{0,1\}^n \to \{0,1\}^\ell$，其中 $n < \ell$，以及一个固定向量 $r \in \{0,1\}^\ell$，我们用 $\tau(g)_r$ 表示一个 poly(ℓ) 大小的命题公式，它以自然的方式编码命题 $r \notin \mathrm{Im}(g)$，因此，如果 r 不属于 g的像，那么 $\tau(g)_r$就是一个命题重言式。对于不属于 g像的 r，重言式 $\tau(g)_r$称为证明复杂性生成器。人们希望能够证明，对于强命题证明系统，在映射 g 具有足够伪随机性的假设下，不存在 $\tau(g)_r$的 poly(ℓ) 大小证明（至少可以在条件性意义上建立这一结论）（另见 [Raz15]）。Krajíček 注意到

证明复杂度生成器与半比特之间的联系（参见 [Kra04，推论 1。3] 以及其后的讨论。）

定理 4.2 的一个直接推论如下。

推论（拉伸证明复杂性生成器；推论 4.6）。设 $b\colon \{0,1\}^n \to \{0,1\}^{n+1}$ 是一个可在 P/poly 中计算的半比特。设 $0 < c < 1$ 是一个常数，且 $\ell = n + n^c$。那么，存在一个证明复杂性生成器 $g\colon \{0,1\}^n \to \{0,1\}^\ell$，它属于 P/poly，使得对于每个命题证明系统，在选择 $r \in \{0,1\}^\ell$ 时，以至少 $1 - \frac{1}{\ell^{\omega(1)}}$ 的概率，不存在大小为 poly(n) 的重言式 $\tau(g)_r$ 的证明。[1]

2.1.4 技术概述

我们通过一种新颖且更灵活地运用混合论证，并结合其他思路，证明半比特的可拉伸性。

混合论证（又称混合方法、混合技术等）是一种常见的证明技巧，源自 Goldwasser 和 Micali 的研究 [GM84]。该名称由 Leonid Levin 命名。（下文所用记号见 第 3.1 节 。）给定一个生成器 $g\colon \{0,1\}^n \to \{0,1\}^{m(n)}$、一个区分器 D和一个函数 p（通常是多项式），满足

$$\mathbb{P}\left[D(U_m) = 1\right] - \mathbb{P}\left[D(g(U_n)) = 1\right] \geq 1/p(n), \tag{$*$}$$

其中，U_m 表示真正随机的字符串，$g(U_n)$表示伪随机字符串。标准混合论证定义了一系列（即一个有序集合）随机变量 H_i，称为混合变量，从一个极端 U_m逐步过渡到另一个极端 $g(U_n)$。一个具体例子是 $H_i := g(U_n)[1...i] \cdot U_{m-i}, 0 \leq i \leq m$（其中 $\cdot$ 表示拼接）。在这个例子中，确实有 $H_0 = U_m$和 $H_m = g(U_n)$。于是，不等式（$*$）可以写成：

$$\begin{aligned} 1/p(n) &\leq \mathbb{P}\left[D(U_m) = 1\right] - \mathbb{P}\left[D(g(U_n)) = 1\right] \\ &= \sum_i (\mathbb{P}\left[D(H_i) = 1\right] - \mathbb{P}\left[D(H_{i+1}) = 1\right]). \end{aligned}$$

因此，通常下一步是声称存在某个 i 使得

$$\mathbb{P}\left[D(H_i) = 1\right] - \mathbb{P}\left[D(H_{i+1}) = 1\right] \geq \frac{1}{k \cdot p(n)},$$

其中 k 是混合体的总数（在上面的例子中，$k = m$）。简而言之，我们证明，如果我们能以 U_m 与 $g(U_n)$ 的 $1/p(n$比例将二者区分开来，那么我们就能以 H_i 与 H_{i+1} 之间的某个相邻混合体对 $1/(k \cdot p(n))$ 的比例将二者区分开来。关于混合论证的简单演示，请参见 引理 A.2 。

[1]点 r 是从 $\{0,1\}^\ell$中均匀选取的，并且公式 $\tau(g)_r$ 以概率 $1 - 1/2^{\ell-n}$ 是一个重言式，因为对所有 $r \in \{0,1\}^\ell \setminus \mathrm{Im}(g)$，公式 $\tau(g)_r$ 都是重言式。在一些研究中，证明复杂性生成器被假定对生成器 g像之外的 *every* r 而言是困难的；而在我们的表述中，困难性只是在 r上以高概率成立。目前尚不清楚这是否会对证明复杂性生成器理论产生影响，因为我们不知道有哪个情形用到了这样的性质：对每个 $r \notin \mathrm{Im}(g)$，$\tau(g)_r$ 都是困难的（尽管据我们所知，所有已证明相对于弱证明系统而言是困难的证明复杂性生成器，对每个$r \notin$Im(g)都困难）。

如上所述，标准混合论证无法用于证明：通过某个拉伸算法 g，由单个半比特 b 拉伸得到的生成器仍然是半比特。现在我们直观地解释其中的原因。通常的证明思路如下：我们反设 g 不是半比特。于是，存在一个区分器 D，用于区分 g，以及一个函数 p，使得$\mathbb{P}[D(U_m) = 1] - \mathbb{P}[D(g(U_n)) = 1] \geq 1/p(n)$；特别地，$\mathbb{P}[D(g(U_n)) = 1] = 0$，因为 D 能破坏半比特性质。我们希望基于 D构造一个新的、合适的区分器 C，用于区分 b。然而，正如上文所见，标准混合论证只能推出：对于某个函数 p'，有 $\mathbb{P}[D(H_i) = 1] - \mathbb{P}[D(H_{i+1}) = 1] \geq 1/p'(n)$，却无法推导出 $\mathbb{P}[D(H_{i+1}) = 1] = 0$。因此，我们不清楚如何继续这一构造。

定理 4.2 的证明采用逆否命题。我们假设存在一个区分器 D，能够以所需方式区分半比特 g（由半比特拉伸算法 4.1 从单个半比特 b 扩展而来），并希望构造一个能够区分 b 的区分器 C。我们没有直接对 D 应用混合论证，而是将混合论证应用于一个基于 D 定义的新区分器 D'：这个新区分器 D' 可以利用非确定性改变混合中的伪随机部分，从而“放大”随机性认证的概率（直观而言，这可以看作是将标准混合论证中的平均情况分析改为最坏情况分析或存在性分析）。通过对 D' 应用混合论证，我们得以确定证明中记为 S_2 的一个非空类，其中包含随机字符串 $y_{i+1} \dots y_m$，这些字符串不是随机见证（也就是说，对于该类中的每个 $y_{i+1} \dots y_m$，都不存在种子 x_1、.....、x_i，使得 $D(b(x_1) \dots b(x_i)\ y_{i+1} \dots y_m) = 1$）。因此，对于 S_2 中的 $y_{i+1} \dots y_m$，只有当 y_i 真正随机时（即不等于任何种子 x 对应的 $b(x)$），$y_i y_{i+1} \dots y_m$ 才能成为随机见证（也就是说，存在种子 x_1、.....、x_{i-1}，使得 $D(b(x_1) \dots b(x_{i-1})\ y_i \dots y_m) = 1$）。混合论证还意味着，在 S_2 中存在这样一个“良好”的 $y_{i+1} \dots y_m$，它可以识别足够多的真正随机 y_i。由此，我们可以构造一个新的区分器 C，用来区分真正随机的字符串和伪随机字符串。

使该证明成立的一个关键步骤是，非确定性地猜测 $x_1, \dots, x_{i-1}$in $b(x_1) \dots b(x_{i-1})z_i$，能够保留 $b(x_1) \dots b(x_{i-1})z_i$ 的“随机性结构”，具体来说：当 $z_i = b(\cdot)$ 是伪随机时，非确定性猜测保留形式 $b(\cdot) \dots b(\cdot)b(\cdot)$（即，$i$ 等长伪随机块）；而当 $z_i = y$ 真正随机时，它保留形式$b(\cdot) \dots b(\cdot)y$（即，$i-1$ 等长伪随机块，后接一个长度相同的真正随机块 y）。对于常见的、能够生成指数级多个新比特的拉伸算法（例如，递归地应用单比特生成器），目前尚不清楚如何利用非确定性来保留给定字符串的“随机性结构”。尽管如此，这种新的证明技术有望激励研究人员进一步探索半比特的可拉伸性。另一方面，也有可能存在这样一种情况：某个特定的半比特无法通过应用于超比特和强伪随机生成器的标准拉伸算法，拉伸为指数级多个半比特。

2.2 基于不可预测性的非确定性安全精细刻画

Yao [Yao82] 将伪随机生成器（PRG）定义为生成这样的序列：对于确定性对手而言，该序列与均匀序列计算不可区分；并证明，这种不可区分性的定义等价于确定性不可预测性。Blum 和 Micali 在早先提出的 PRG 定义中使用了确定性不可预测性 [BM84]。粗略地说，不可预测性是指：给定一个随机字符串的任意真前缀，预测其下一位是不可行的。

在 第 5 节中，我们通过引入非确定性不可预测性的概念，对非确定性困难性（定义 1.1）进行了更细致的刻画。这使我们能够为非确定性困难性建立新的下界和上界：我们将非确定性困难性夹在两种不可预测性性质之间。

具体而言，我们为概率系综提出了四种不可预测性概念：

1. NP/poly-不可预测性：无法被 NP/poly 预测器预测的能力。2. coNP/poly-不可预测性：无法被 coNP/poly 预测器预测的能力。3. $\cup$-不可预测性：无法被 NP/poly 与 coNP/poly 的并集中的预测器预测的能力。4. $\cap$-不可预测性：无法被非确定性函数计算预测器预测的能力。

名称 NP/poly-不可预测性、coNP/poly-不可预测性和 $\cup$-不可预测性（这是 NP/poly $\cup$coNP/poly-不可预测性的简写）都一目了然，而“$\cap$-不可预测性”的用法则不太直观。我们将在 第 5.2 节（其中我们会使用非确定性函数计算机器[2]）中看到，判定问题属于 NP/poly $\cap$coNP/poly 当且仅当它可由一个非确定性多项式规模函数计算算法判定。

我们从不可预测性的角度，确立了生成器非确定性困难性的如下刻画：

定理 (总结 5.9)。 这里， $A \leq B$ 表示，如果生成器具有性质 B，那么它也具有性质 A:

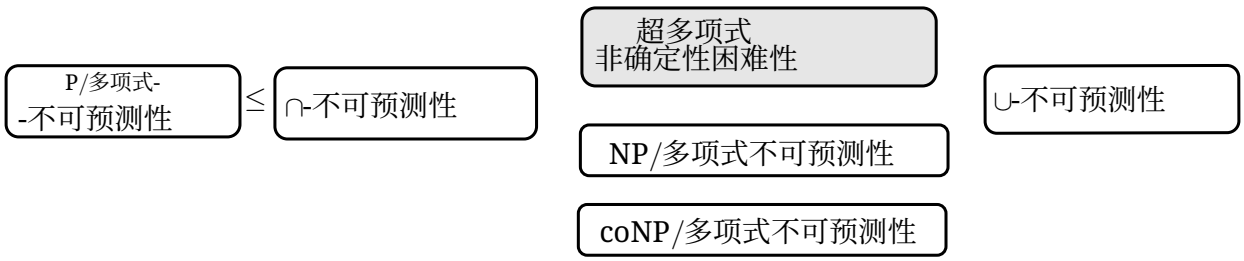

图1：此处的超多项式非确定性困难性指的是 定义 1.1。请注意， $\cup$- 不可预测性至少与 NP/多项式-不可预测性同样强，因为它排除了NPNP/多项式 和 coNP/多项式中的预测器。同样， $\cup$-不可预测性至少与 coNP/多项式- 不可预测性同样强。

2.3 超核心：非确定性情形下的硬核谓词

在确定性背景下，已知强伪随机生成器的存在等价于若干核心密码学原语的存在，例如单向函数、安全私钥加密

[2]如果对于每个输入 $x \in \{0,1\}^n$，每个计算分支都会产生 $\{0,1,\perp\}$ 中的某个值，其中 $\perp$ 表示失败，并且始终存在一个计算分支产生 0 或 1，那么我们就称非确定性算法 $\mathcal{A}$ 是函数计算算法。

方案、数字签名等。Liu 和 Pass [LP20] 最近证明，关于时间受限柯尔莫哥洛夫复杂度具有温和平均情形困难性的元复杂度假设，也等价于强伪随机生成器的存在。另一方面，在非确定性情形下，我们尚无已知的此类等价刻画（即非确定性安全伪随机生成器，也就是超比特）。在 第6节中，我们提出了超核心的定义，将其作为硬核的非确定性变体。随后，我们利用这一概念，首次给出了超比特的等价刻画。

我们首先回顾单向函数和硬核谓词的概念。粗略地说，单向函数（族） f 是指一种易于计算但平均而言难以求逆的函数（其中概率取自 f 的定义域）。更准确地说，“易于计算”意味着 f 属于 P 或 P/poly，而“难以求逆”意味着当 x 不可见时，任何高效的确定性算法只能对 $y = f(x)$ 的一小部分求逆。在一致性设置中，“高效”指的是有界错误概率多项式时间 (BPP) 中的算法；在非一致性设置中，指的是 P/poly 中的算法；而“求逆”是指：对于给定的 $y \in range(f)$，寻找一个 x'，使得 $f(x') = y$。对我们而言，可忽略部分是指：对于任意多项式 p 以及所有足够大的 n，其比例小于 $1/p(n)$。我们称 b：若 $\{0,1\}^n \to \{0,1\}$ 属于 P 或 P/poly，则 $\{0,1\}^n \to \{0,1\}$ 是函数 f 的硬核，当给定 $f(x)$ 时，不可能以至少 $1/2 + 1/\text{poly}(n)$ 的概率高效预测 $b(x)$。

已知硬核谓词的存在性（例如，$b(x) = x[-1]$，字符串 x 的最后一位，是函数 $f(x) = x[1]$（x 的第一位）的硬核），但单向函数的硬核是否存在，以及是否存在任何单向函数，这些问题目前尚无定论。Goldreich 和 Levin [GL89] 证明，内积 $mod\ 2$ 是任何形如 $g(x,r) = (f(x)\ r)$ 的函数的硬核，其中 f 是任意单向函数，且 $|x| = |r|$。随后，H˚astad、Impagliazzo、Levin 和 Luby [HILL99] 证明，强伪随机生成器存在当且仅当单向函数存在。这个定理也可以等价地表述为：强伪随机生成器存在当且仅当某个单向函数存在硬核。

既然强伪随机生成器存在当且仅当某个单向函数存在硬核，而超比特是强伪随机生成器的非确定性对应概念，那么一个值得提出的问题是：

单向函数和硬核的非确定性类比是什么 $s?$

要提出超单向函数的合理定义并非易事，因为对于任意函数 f，非确定性算法总能通过猜测某个 x 并检查 $f(x) = y$，来对值域元素 y 求逆。类似地，超核心谓词的合理定义同样并不简单：对于任意函数 f 和谓词 b，非确定性算法在以 $f(x)$ 作为输入时，可以通过猜测 x，然后应用 b，来预测 $b(x)$。

我们提出函数 f 的超核心定义，它们能够抵抗 NP/poly 和 coNP/poly 预测器，其含义遵循 定义 6.7，其中 $f(x)$ 被作为输入给出。借助这一定义，我们可以建立以下等价性（如果对于每个 n），我们称函数 $f\colon \{0,1\}^n \to \{0,1\}^{m(n)}$ 是 非收缩的，如果 $m(n) \geq n$）：

定理 (非正式；定理 6.13)。 存在某个非收缩函数的超核心，当且仅当存在超比特。

这一结果类似于已知的一个等价关系：某个单向函数存在硬核，当且仅当强伪随机生成器存在。我们还证明，某一类可能具有硬核的函数，不可能拥有任何超核心。这也表明，单向函数可能根本不拥有超核心。（“大部分为一对一”的定义见定理 6.16 之前的正文。）

定理 （非正式表述；定理 6.16）。如果一个非收缩函数 f 是“大部分为一对一”的，那么 f 就没有超核心。

我们在第6节中取得的成果有助于进一步理解伪随机生成器的非确定性困难性，并在非确定性情形下制定单向函数的合理定义。

2.4 半比特和 PAC 学习存在性的论证

这里我们提及 Pich [Pic20] 的工作。他通过证明：如果多项式规模布尔电路类无法由次指数规模电路 PAC 学习，那么半比特就存在（其中学习器可以生成一个用于近似目标函数的非确定性或余非确定性算法），从而说明了半比特存在的合理性。因此，如果多项式规模布尔电路类无法由次指数规模电路 PAC 学习，且半比特的存在意味着超比特的存在，那么就不存在可用于对抗 NP/qpoly 的自然性质，且这些自然性质对 P复杂度类/poly 有用。为保证内容自成一体，我们将阐述这些结果。

回想一下，PAC 学习 算法可以通过破解伪随机生成器来构造（例如，参见[BFKL94、OS17]）。PAC 学习模型（即“可能近似正确学习”的缩写）由 Valiant 于[Val84、VS84、Val85]中提出并加以发展。在 PAC 学习模型中，学习器的目标是学习从某个目标集合（例如，决策树类、布尔合取类、P复杂度类/poly等）中抽取的任意目标函数 f-预言机，并从某个假设类中选出一个泛化函数 f'，称为假设。所选函数必须以高概率具有较低的泛化误差（“PAC”中的“近似正确”）。此外，学习器应当高效，并输出能够针对任意给定输入进行高效求值的假设。

定理 (非正式；定理 A.3Pich [Pic20])。假设多项式规模布尔电路类无法由次指数规模电路 PAC 学习，且学习器可以生成用于逼近目标函数的非确定性或余非确定性电路，则半比特存在。

这一结果表明半比特存在是合理可信的，因为人们普遍认为（例如，参见 [RS21]），学习P/poly 是困难的。因此，假设即使允许学习器生成非确定性或余非确定性电路族，这项任务仍然困难，也是合理的。

与半比特不同，值得一提的是，目前尚不清楚如何将超比特的存在与 PAC 可学习性的困难联系起来（也就是说，当我们考虑超比特时，定理 A.3 的证明无法继续成立）。

3 预备知识与基本概念

3.1 记号与约定

我们遵循以下约定：

- $\mathbb{N}$ 表示正整数集合（不包括 0）。对于 $n \in \mathbb{N}$，$[n]$ 表示 $\{1, ..., n\}$。$[0]$ 是空集 $\emptyset$。

- 布尔电路 C 的大小，记作 $size(C)$ 或 $|C|$，是门的总数（包括输入门）。 $Circuit[s]$ 表示大小至多为 s 的布尔电路。如果$s: \mathbb{N} \to \mathbb{N}$ 是一个函数，那么 $Circuit[s]$包含所有满足以下条件的布尔电路族 C_n：对于所有足够大的 n，$|C_n| \leq s(n)$。

- 本文中考虑的所有分布默认均为均匀分布。除非另有说明， U_n 表示 $\{0,1\}^n$ 上的均匀分布。

- 对于函数 f、g：$\{0,1\}^n \to \{0,1\}$，我们称 g γ近似于 f，如果 $\mathbb{P}_x[f(x) = g(x)] \geq \gamma$。

- 对于字符串 x，若其位从左至右依次编号为 1 到 $|x|$，则 $x[i]$ 表示其第 i位，而 $x[i...j]$ 表示 x 中从 i 到 j 的子字符串（若 $i > j$，则为 $x[i...j] = \varepsilon$，即空字符串）。 $x[-i]$ 表示其倒数第 i位。

- 对于字符串 x、y，我们可以用以下任意一种方式表示 x 与 y的拼接： xy、(x, y)、 $x \cdot y$。

- 在不会引起歧义时，我们可以不在口头上区分函数及其字符串表示（例如真值表，或编码该函数的电路表示的字符串等）。

我们也可以遵循复杂性社区或文献中使用的其他常见约定。

3.2 计算模型

定义 3.1（随机电路；等价地，概率电路）。如果电路 C 除标准输入位（类似于非随机电路的输入位）外，还包含零个或多个随机输入位（即从随机分布中取出的位），则称其为随机电路（等价地，概率电路）。如果对于每个n， C_n 都是具有 n 个标准输入位的随机电路，则称 $\{C_n\}_{n=1}^{\infty}$ 为随机电路族。

注意，如果随机电路族 $\{C_n\}$ 属于 $Circuit[s(n)]$, ，这意味着对于每个充分大的 n，随机电路 C_n 的规模至多为 $s(n)$，这也就自动限制了 C_n 所允许拥有的随机输入位数量。

定义 3.2（非确定性电路和余非确定性电路）。如果电路 $C(\overline{x},\overline{r})$除标准输入位 $\overline{x}$外，还包含零个或多个非确定性输入位 $\overline{r}$ （即用于控制电路所作非确定性决策的位），则称其为（余）非确定性电路。具有单个输出位的非确定性电路称为接受输入 $\overline{\alpha} \in \{0,1\}^n$，当且仅当存在一个对 $\overline{r}$的赋值 $\overline{\beta} \in \{0,1\}^{|\overline{r}|}$，使得 $C(\overline{\alpha},\overline{\beta}) = 1$（否则称其拒绝 $\overline{x}$）。具有单个输出位的余非确定性电路称为拒绝输入 $\overline{\alpha} \in \{0,1\}^n$，当且仅当存在一个对 $\overline{r}$的赋值 $\overline{\beta} \in \{0,1\}^{|\overline{r}|}$，使得$C(\overline{\alpha},\overline{\beta}) = 0$ （否则称其接受 $\overline{x}$）。如果对于每个 n， C_n 都是具有 n 个标准输入位的（余）非确定性电路，则称 $\{C_n\}$ 为（余）非确定性电路族。

定义 3.3（预言机电路）。 C 是预言机电路，如果它可以使用预言机门。若 C 具有计算布尔函数 f_1、.....、 f_k 的预言机门，我们将 C 记作 $C^{f_1,...,f_k}$。

我们注意到，计算布尔函数 $f: \{0,1\}^n \to \{0,1\}$ 的预言机门具有 n 扇入；在我们的模型中，预言机门可以出现在电路的任何位置。

3.3 自然证明

设 F_n 为所有函数 f 的集合，$\{0,1\}^n \to \{0,1\}$ 和 Γ，且 Λ 为复杂性类。如果每个 $C_n \subseteq F_n$，我们就称 $C = (C_n)_{n\in\mathbb{N}}$ 是布尔函数的一种组合性质。

定义 3.4 (自然性质 [RR97])。我们称组合性质 $C = (C_n)_{n\in\mathbb{N}}$ 是**Γ-自然的**，如果存在某个 $C' = (C'_n)_{n\in\mathbb{N}}$ 满足 $C'_n \subseteq C_n$，对于每个 n 都满足：

- 构造性。判断当 f 以其真值表作为输入编码时，$f \in C'_n$ 是否可在 Γ 中计算。
- 大性。$|C'_n| \geq 2^{-O(n)} \cdot |F_n|$ 对于所有足够大的 n 都成立。

我们称 C 对 **Λ** 有用，当且仅当它满足：

- 有用性。对于任意函数族 $f = (f_n)_{n\in\mathbb{N}}$，如果 $f_n \in C_n$ 无穷多次成立，则 $f \notin \Lambda$。

电路下界证明（即证明某个函数族不属于 Λ）称为针对 Λ 的 Γ-自然证明，如果它显式或隐式地使用了某种针对 Λ 有用的 Γ-自然组合性质。特别地，针对 P/poly 的 P/poly-自然证明，是指使用针对 P/poly 有用的 P/poly-自然组合性质的证明。

我们注意到，与自然组合性质不同，自然证明这一概念并没有以数学上严格的方式定义。尽管如此，在陈述中使用“自然证明”这一术语，更直观地体现了我们的意图，也不会影响陈述的严谨性：每当我们说针对 Λ 的 Γ-自然证明存在或不存在时，从数学上严格的意义来说，我们指的是针对 Λ 的 Γ-自然组合性质存在或不存在。

3.4 伪随机生成器

这里我们回顾伪随机生成器的基本定义。如引言中所述，本文所考虑的所有分布默认都是均匀分布，且除非另有说明，U_n 表示 $\{0,1\}^n$ 上的均匀分布。

定义 3.5（生成器）。若函数族 $g_n : \{0,1\}^n \to \{0,1\}^{l(n)}$ 满足 $g_n \in$P/poly，且对每个 n都有 $l(n) > n$，则称其为生成器。我们称这样的 l 为扩展函数，并称 $l(n) - n$ 为 g_n 的扩展长度（有时也将 $l(n)$称为扩展长度）。

需要指出的是，本文中的所有生成器都可在P/poly中计算，不过在更一般的情形下，并不要求生成器可由P/poly计算（参见 [NW94]）。

定义 3.6（标准困难性）。设 $g_n : \{0,1\}^n \to \{0,1\}^{l(n)}$ 是一个生成器。那么，g_n 的困难性 $H(g_n)$是满足以下条件的最小 s：存在一个大小至多为 s的（确定性）电路 D，使得

$$\left| \mathop{\mathbb{P}}_{y\in\{0,1\}^{l(n)}}[D(y) = 1] - \mathop{\mathbb{P}}_{x\in\{0,1\}^n}[D(g_n(x)) = 1] \right| \geq 1/s(n).$$

绝对值中的两个项的顺序以及绝对值本身并不重要，因为在确定性情形下，我们总可以翻转区分器的输出位 D。

定义 3.7（（强）伪随机生成器（PRG））。若生成器 $g\colon \{0,1\}^n \to \{0,1\}^{l(n)}$，则称其为（强）PRG；如果对于每个 $D \in$P/poly、每个多项式 p，以及所有充分大的n，

$$\left| \mathbb{P}_{y\in\{0,1\}^{l(n)}}[D(y)=1] - \mathbb{P}_{x\in\{0,1\}^n}[D(g(x))=1] \right| < 1/p(n).$$

换言之，强伪随机生成器被定义为能够抵御所有多项式规模区分器的生成器。某些文献采用另一种定义：对于某个 $\varepsilon > 0$ 和所有足够大的 n，生成器的安全性至少为 2^{n^ε}；这种定义所描述的是更强的伪随机生成器。

若存在某个 $\varepsilon > 0$，使得对于所有充分大的 n，函数 $f(n)\colon \mathbb{N} \to \mathbb{R}^+$ 满足 $f(n) \geq 2^{n^\varepsilon}$，则称函数至少是指数级的。若对于每个 $\varepsilon > 0$，都存在无穷多个 n，使得 $f(n) < 2^{n^\varepsilon}$，则称函数不是（至少）指数级的；后一条件等价于：存在一个无限单调序列 $(n_i) \subseteq \mathbb{N}$，使得 $f(n_i)$ 相对于 n_i 是次指数级的（即，$f(n_i) = 2^{n_i^{o(1)}}$）。

强伪随机生成器的存在被认为相当可信，因为许多难解问题（例如整数分解）似乎为构造此类生成器提供了基础（参见 [Gol01]）。

猜想。强伪随机生成器存在。

Razborov 和 Rudich 证明了，强伪随机生成器的存在排除了P/poly-自然证明对以下问题有用：P/poly [RR97]。

目前尚无强伪随机生成器的具体实例，因为此类生成器的存在意味着，在一致性设置下P $\neq$ NP，而在非一致性设置下P/poly $\neq$NP/poly。尽管如此，人们构造了能够欺骗较弱区分器类的生成器（例如 Nisan 和 Wigderson [NW94]）。[3]

3.5 超比特和半比特

这里简要回顾[Rud97] 中的主要结果和开放问题，这些内容与我们的工作相关。（部分文本重复自引言。）

定义 3.8（非确定性困难性）。设 $g_n\colon \{0,1\}^n \to \{0,1\}^{l(n)}$ 是一个生成器。那么，g_n 的非确定性困难性 $H_{\text{nh}}(g_n)$（也称为超硬度）是满足以下条件的最小 s：存在一个大小至多为 s 的非确定性电路 D，使得

$$\mathbb{P}_{y\in\{0,1\}^{l(n)}}[D(y)=1] - \mathbb{P}_{x\in\{0,1\}^n}[D(g_n(x))=1] \geq \frac{1}{s}. \tag{1}$$

与确定性硬度的定义不同，左侧两种可能性的顺序至关重要。这种顺序迫使非确定性区分器对给定输入的随机性进行认证。颠倒顺序或保留绝对值都会使破解g变得轻而易举：区分器 D 只需猜测一个种子 x，并检查 $g(x)$ 是否等于给定输入。对于这样的 D，我们有 $\mathbb{P}[D(g(x)) = 1] = 1$ 且 $\mathbb{P}[D(y) = 1] \leq 1/2$。

我们将指数级超难生成器称为超比特：

[3]对于弱模型，已知复杂性理论生成器和密码学生成器。能够欺骗 AC^0 的复杂性理论生成器由 Nisan 展示（即以奇偶函数作为困难函数的 Nisan-Wigderson 生成器，并且早于 [NW94]）。密码学生成器例如在 [DVV16]中构造。对于P复杂度类/poly，复杂性理论生成器和密码学生成器都尚未知晓。然而，复杂性理论生成器所需的假设（例如，E 需要指数规模）远弱于密码学生成器所需的假设（例如，单向函数存在）。

定义 3.9（超比特）。若某个生成器（属于 P/poly） $g_n : \{0,1\}^n \to \{0,1\}^{n+c}$ for some $c : \mathbb{N} \to \mathbb{N}$满足 $H_{\text{nh}}(g_n) \geq 2^{n^\varepsilon}$ for some $\varepsilon > 0$ and all sufficiently large n's，则称其为 c 超比特（或 c-超比特）。特别地，若 $c = 1$，则称 g_n 为超比特。

因此，超比特一词指的是能够骗过“超级”强大对手的伪随机比特。Rudich 基于子集和问题构造了一个候选超比特，并猜想：

猜想（超比特猜想）。存在超比特。

Rud [ich97] 中的主要定理如下，该定理基于上文讨论的超比特的可拉伸性得证。

˜定理 3.10（[Rud97]）。如果存在超比特，那么不存在对 P/poly 有用的 $NP/qpoly$-自然性质，其中 $NP/qpoly$是由非一致准多项式规模电路族识别的语言类（这里准多项式指 $n^{log^{O(1)}(n)}$）。

˜我们指出，在此定理中， $NP/qpoly$-自然性质的“大性”要求实际上可以放宽为 $|C'_n| \geq 2^{-n^{O(1)}} \cdot |F_n|$（参见 定义 3.4）。

Rudich 还提出了另一种概念，称为半硬度；他认为这种概念比超硬度更直观：

定义 3.11（半硬度）。设 $g_n : \{0,1\}^n \to \{0,1\}^{l(n)}$ 是一个生成器（属于 P/poly）。那么 g_n 的半硬度 $H_{\text{dh}}(g_n)$ 是满足以下条件的最小 s ： 存在一个大小至多为 s 的非确定性电路D ，使得

$$\mathbb{P}_{y \in \{0,1\}^{l(n)}}[D(y) = 1] \geq \frac{1}{s} \quad and \quad \mathbb{P}_{x \in \{0,1\}^n}[D(g_n(x)) = 1] = 0. \tag{2}$$

注意，条件 (2) 要求区分器对所有生成的字符串都不出错，因此比条件 (1) 更强。所以，对于每个生成器 g，都有 $H_{\text{nh}}(g) \leq H_{\text{dh}}(g)$。

我们将指数级半硬生成器称为半比特；这里的“半”意为“一半”：

定义 3.12（半比特）。如果某个生成器（属于 P/poly） $g_n : \{0,1\}^n \to \{0,1\}^{n+c}$ 对某个 $c : \mathbb{N} \to \mathbb{N}$ 成立，则称其为 c 半比特（或 c-半比特），如果对于某个 $\varepsilon > 0$ 以及所有充分大的 n，都有 $H_{\text{dh}}(g_n) \geq 2^{n^\varepsilon}$ 。特别地，如果 $c = 1$，我们称 g_n 为一个半比特。

正如 $H_{\text{nh}}(g) \leq H_{\text{dh}}(g)$，自然可以猜想：

猜想 (半比特猜想 [Rud97])。存在一个半比特。

因此，自然会提出这样的问题：

开放问题 ([Rud97])。半比特的存在是否意味着超比特的存在?

如上所述，超比特可以扩展为多项式数量的超比特，也可以扩展为能够抵御非确定性对手的伪随机函数生成器。相比之下，在我们的工作之前，半比特能否扩展为哪怕两个半比特尚属未知：

开放问题 ([Rud97]；已在 第 4 节中解决。 给定一个半比特，是否可以将其扩展为 2 个半比特?

仍然悬而未决的问题是：

开放问题 ([Rud97])。 给定一个半比特，是否可以构造一个具有指数级半比特硬度的伪随机函数生成器?

对上一个问题的肯定回答将回答以下问题：

˜开放问题 ([Rud97]).半比特的存在是否排除了针对 $NP/qpoly$- 自然性质P复杂度类/多项式?

3.6 无穷多次超比特和半比特

这里，我们对超比特和半比特的一种较弱变体进行形式化定义，该变体仅要求存在无穷多个 n 满足“困难性”（回顾一下，超比特/半比特要求对所有充分大的 n 都满足困难性）。这种变体偶尔会在文献中隐式出现，但可能尚未得到正式定义。

定义 3.13（无穷多次超比特/半比特）。若存在某个 $c:\mathbb{N}\to\mathbb{N}$，生成器（属于 P/poly） $g_n:\{0,1\}^n\to\{0,1\}^{n+c}$ 被称为无穷多次（i.o.）超比特/半比特，如果对于某个 $\varepsilon>0$ 和无穷多个 n，有 $H_{\mathrm{nh}}(g_n)\geq 2^{n^\varepsilon}/H_{\mathrm{dh}}(g_n)\geq 2^{n^\varepsilon}$。特别地，若 $c=1$，则称 g_n 为无穷多次超比特/半比特。

事实上，一个简单的观察是：对于一个出现频率足够高的无穷多次超比特/半比特，我们可以通过恰当地选取给定输入 n 的一个前缀，并对该前缀应用无穷多次算法，构造出一个超比特/半比特。更多细节如下。不过，我们尚不清楚是否能从任意无穷多次超比特/半比特构造出一个超比特/半比特。

引理 3.14。假设 $g_n:\{0,1\}^n\to\{0,1\}^{n+c}$ 对于某个 $c\in\mathbb{N}$ 是 c 无穷多次超比特/半比特。若存在一个多项式 p 和一个无限单调序列 $(n_i)_{i\in\mathbb{N}}\subseteq\mathbb{N}$，使得：(1)$n_{i+1}\leq p(n_i)$，且 (2) 对某个 $\varepsilon>0$ 及每个 $n\in(n_i)_{i\in\mathbb{N}}$，$H_{\mathrm{nh}}(g_n)\geq 2^{n^\varepsilon}/H_{\mathrm{dh}}(g_n)\geq 2^{n^\varepsilon}$，那么存在一个由 g_n构造出的 c-超比特/c-半比特。

证明。我们给出由无穷多次超比特构造超比特的证明；由无穷多次半比特构造半比特的证明几乎完全相同。

假设 g、(n_i)、p 如引理陈述所述。记 $m=n+c$ 和 $m_i=n_i+c$。我们按如下方式构造一个新的生成器 G：

Giv对于 $x_n\in\{0,1\}^n$，存在一个 i 使得 $n_i\leq n<n_{i+1}$。定义 $G(x_n)=g(a)\cdot b$，其中 $a=x_n[1...n_i]$、$b=x_n[n_i+1...n]$。注意到 $|G(x_n)|=|g(a)|+|b|=n+c$。

我们要证明 G 确实是超比特。反设 G 不是。于是存在一个无限单调序列 $S\subseteq\mathbb{N}$、一个次指数函数 s以及一个大小为 s 的区分器D，使得对于每个 $n\in S$，

$$1/s(n)\leq\mathbb{P}\left[D(U_m)=1\right]-\mathbb{P}\left[D(G(U_n))=1\right]=\mathbb{P}\left[D(U_{m_i}U_{m-m_i})=1\right]-\mathbb{P}\left[D(g(U_{n_i})U_{m-m_i})=1\right]$$

因此，对于每个 $n\in S$，都存在一个固定字符串 $w=w(n)$，使得

$$1/s(n)\leq\mathbb{P}\left[D(U_{m_i}w)=1\right]-\mathbb{P}\left[D(g(U_{n_i})w)=1\right].$$

现在我们构造一个新的区分器 D'，用于 g，如下所示：

给定作为输入的 $Y \in \{0,1\}^{m_i}$，如果存在一个 $n \in S$ 使得 $n_i \le n < n_{i+1}(\le p(n_i))$，则 D' 输出 $D(Yw(n))$；否则 D' 始终输出 0（这意味着对于这样的 n_i，D' 无法发挥任何作用）。

注意，假设 (1) $n_{i+1} \le p(n_i)$ 保证了 D' 的效率。由于 S 是无限的，且每个 $n \in S$ 都介于某个 n_i 和 n_{i+1} 之间，因此存在无穷多个 n_i 满足：

$$\mathbb{P}\left[D'(U_{m_i}) = 1\right] - \mathbb{P}\left[D'(g(U_{n_i})) = 1\right] = \mathbb{P}\left[D(U_{m_i}w) = 1\right] - \mathbb{P}\left[D(g(U_{n_i})w) = 1\right] \ge 1/s(n)$$

这表明，对于无穷多个 n_i，存在某个次指数 s'，使得 $H_{\text{nh}}(g(n_i)) \le 1/s'(n)$，这与引理陈述中的假设 (2) 矛盾。 □

备注。引理 3.14 常被隐含地用于超比特和半比特的构造中 .

4 半比特拉伸

在本节中，我们肯定地回答半比特是否可以拉伸这一问题；这个问题自 Rudich 的原始工作以来一直是一个开放问题 [Rud97]。

我们提出半比特拉伸算法 4.1，它将单个半比特拉伸为 n^c 个半比特，其中任意 $c < 1$，并验证该拉伸算法的正确性（即，验证新延展生成器的指数级半比特硬度）。直观而言，半比特拉伸算法 4.1将给定种子划分为互不相交的部分，并对每一部分应用一半比特生成器。此处提出的算法与文献中出现的其他拉伸或放大算法非常相似（例如，参见 [Yao82、Lub96]）。半比特拉伸真正的难点在于证明所尝试的拉伸是正确的（即，证明所应用的拉伸算法能够保持指数级半比特硬度）。

半比特拉伸算法 4.1。设 $b_n : \{0,1\}^n \to \{0,1\}^{n+1}$ 是一个半比特，且 $0 < c < 1$ 是一个常数。我们定义一个新的生成器 $g : \{0,1\}^N \to \{0,1\}^{N+m}$，其输入长度为$N$、扩展长度为 $m = \lceil N^c \rceil$，如下所示：给定输入 x（长度为 N），令 $n = \lfloor \frac{N}{m} \rfloor$ 和$x = x_1 x_2 \dots x_m r$，其中每个 x_i 的长度为 n，并定义 $g(x) = b(x_1) \dots b(x_m)r$。

正确性证明采用逆否命题。也就是说，我们假设存在一个区分器 D，它能以所需的方式区分半比特 g（由半比特拉伸算法 4.1 从单个半比特 b扩展而来），并希望构造一个区分器 C 来区分 b。我们不直接对 D 应用混合论证，而是对一个基于 D 定义的新区分器 D' 应用混合论证：新区分器 D' 可以利用非确定性改变混合中的伪随机部分，从而“放大”随机性认证的概率（直观地说，这可以视为将标准混合论证中的平均情况分析改为最坏情况分析，或“存在性”分析）。通过对 D' 应用混合论证，我们能够找出一个非空类 S_2，其中包含随机字符串 $y_{i+1} \dots y_m$，但这些字符串不是随机见证（也就是说，对于该类中的每个 $y_{i+1} \dots y_m$，都不存在种子 x_1，.....，x_i，使得 $D(b(x_1) \dots b(x_i)\ y_{i+1} \dots y_m) = 1$）。因此，对于 $y_{i+1} \dots y_m \in S_2$，只有当 y_i是真正随机的（也就是说，它不等于某个种子 x对应的 $b(x)$）时，$y_i y_{i+1} \dots y_m$才能成为随机见证（即存在种子x_1，.....，x_{i-1}，使得 $D(b(x_1) \dots b(x_{i-1})\ y_i \dots y_m) = 1$）。混合论证还意味着，在 S_2中存在这样一个“良好”的 $y_{i+1} \dots y_m$，能够识别出足够一部分真正随机的 y_i。由此，我们可以构造一个新的区分器 C，用来区分真正随机的字符串与伪随机字符串。

在证明中，我们用粗体表示随机变量。

定理 4.2。（半比特拉伸的主定理）生成器 g（具有次线性拉伸长度），如 半比特拉伸算法 4.1 中所定义，至少具有指数级半比特硬度。

证明。半比特 b 和常数 c 如半比特拉伸算法 4.1中所述。为导出矛盾，假设 g 不具有指数级半比特硬度。也就是说，存在一个规模为次指数级（相对于输入长度，记为 N）的非确定性电路 D，使得对于无穷多个 N（回顾 $m = \lceil N^c \rceil$ 和 $n = \lfloor \frac{N}{m} \rfloor$，且不失一般性，我们可以假设$m|N$），有 $\mathbb{P}[D(\mathbf{y_1} \dots \mathbf{y_m}) = 1] \geq 1/|D|$ 且 $\mathbb{P}[D(b(\mathbf{x_1}) \dots b(\mathbf{x_m})) = 1] = 0$，其中 $\mathbf{x_1}$、……、$\mathbf{x_m}$ 是完全独立的长度为-n 的随机字符串，而 $\mathbf{y_1}$、……、$\mathbf{y_m}$ 是完全独立的长度为-$(n+1)$的随机字符串。我们的目标是以所需的方式高效地攻破 b。

我们定义一个新的非确定性电路 D'，它接受一对输入：第一个与 D 的输入相同，即一个 $(N+m)$ 位字符串 $y_1 \dots y_m$；第二个是 $i \in \{0, 1, \dots, m\}$（经过适当编码 i）：

> 给定输入 $(y_1 \dots y_m, i)$，D' 猜测 n 位字符串 x_1、……、x_i，然后执行 D 在 $b(x_1) \dots b(x_i)$ 上所做的操作 $y_{i+1} \dots y_m$。

我们注意到：

- $\mathbb{P}[D'(\mathbf{y_1} \dots \mathbf{y_m}, 0) = 1] = \mathbb{P}[D(\mathbf{y_1} \dots \mathbf{y_m}) = 1] \geq 1/|D|$（根据 D' 的定义）；
- $\mathbb{P}[D'(b(\mathbf{x_1}) \dots b(\mathbf{x_m})\ m) = 1] = \mathbb{P}[D(b(\mathbf{x_1}) \dots b(\mathbf{x_m})) = 1] = 0$（根据关于 D 的假设；否则 $\mathbb{P}[D'(b(\mathbf{x_1}) \dots b(\mathbf{x_m})\ m) = 1] > 0$ 就意味着存在 x_1、……、x_m，使得 $D(b(x_1) \dots b(x_m)) = 1$）；
- D' 也具有亚指数规模。

现在，我们可以对 D'应用混合论证：

$$\begin{aligned} 1/|D| &\leq \mathbb{P}\left[D'(\mathbf{y_1} \dots \mathbf{y_m}, 0) = 1\right] - \mathbb{P}\left[D'(b(\mathbf{x_1}) \dots b(\mathbf{x_m}), m) = 1\right] \\ &= \sum_i (\mathbb{P}\left[D'(b(\mathbf{x_1}) \dots b(\mathbf{x_{i-1}})\, \mathbf{y_i} \dots \mathbf{y_m}, i-1)\right] - \mathbb{P}\left[D'(b(\mathbf{x_1}) \dots b(\mathbf{x_i})\, \mathbf{y_{i+1}} \dots \mathbf{y_m}, i)\right]) \end{aligned}$$

可得存在一个 $i = i(N)$（对无穷多个 N而言），使得

$$P_{i-1} - P_i \geq 1/(m \cdot |D|),$$

其中 $P_{i-1} := \mathbb{P}[D'(b(\mathbf{x_1}) \dots b(\mathbf{x_{i-1}})\mathbf{y_i}\mathbf{y_{i+1}} \cdots \mathbf{y_m}, i-1) = 1]$ 且
$P_i := \mathbb{P}[D'(b(\mathbf{x_1}) \dots b(\mathbf{x_{i-1}})b(\mathbf{x_i})\mathbf{y_{i+1}} \cdots \mathbf{y_m}, i) = 1]$。由于 m 关于 n是次线性的，$m \cdot |D|$ 在 N中仍然是次指数级的。

> 我们用 $S = \{0,1\}^{(n+1)\times(m-i)}$ 表示由$(m-i)$元组组成的集合，其中每个元组包含$(n+1)$位字符串，并令
>
> $$S_1 = \{(y_{i+1}, \dots, y_m) \in S : \exists (x_1, \dots, x_i) \in \{0,1\}^{n \times i}\ D(b(x_1) \dots b(x_i) y_{i+1} \dots y_m) = 1\},$$
>
> and
>
> $$S_2 = S \setminus S_1.$$
>
> 注意：

- $D'(b(x_1)\dots b(x_{i-1})\ y_iy_{i+1}\dots y_m, i-1) = 1$ 当且仅当 $D'(O\ y_iy_{i+1}\dots y_m, i-1) = 1$，其中 $O = 0^{(n+1)\cdot(i-1)}$（因为根据构造，$D'(\dots, i-1)$ 会忽略其前 $i-1$ 个输入字符串）；
- $D'(b(x_1)\dots b(x_{i-1})b(x_i)\ y_{i+1}\dots y_m, i) = 1$ 当且仅当 $(y_{i+1}, \dots, y_m) \in S_1$，因此 $\mathbb{P}[\mathbf{y_{i+1}}\cdots\mathbf{y_m} \in S_1] = P_i$。

因此，

$$\begin{aligned}
P_{i-1} &= \mathbb{P}\left[D'(b(\mathbf{x_1})\dots b(\mathbf{x_{i-1}})\mathbf{y_i}\mathbf{y_{i+1}}\cdots\mathbf{y_m}, i-1) = 1\right] \\
&= \mathbb{P}\left[D'(b(\mathbf{x_1})\dots b(\mathbf{x_{i-1}})\mathbf{y_i}\mathbf{y_{i+1}}\cdots\mathbf{y_m}, i-1) = 1 | \mathbf{y_{i+1}}\cdots\mathbf{y_m} \in S_1\right]\cdot\mathbb{P}\left[\mathbf{y_{i+1}}\cdots\mathbf{y_m} \in S_1\right] + \\
&\quad \mathbb{P}\left[D'(b(\mathbf{x_1})\dots b(\mathbf{x_{i-1}})\mathbf{y_i}\mathbf{y_{i+1}}\cdots\mathbf{y_m}, i-1) = 1 | \mathbf{y_{i+1}}\cdots\mathbf{y_m} \in S_2\right]\cdot\mathbb{P}\left[\mathbf{y_{i+1}}\cdots\mathbf{y_m} \in S_2\right] \\
&\le 1\cdot\mathbb{P}\left[\mathbf{y_{i+1}}\cdots\mathbf{y_m} \in S_1\right] + \\
&\quad \mathbb{P}\left[D'(O\ \mathbf{y_i}\mathbf{y_{i+1}}\cdots\mathbf{y_m}, i-1) = 1 | \mathbf{y_{i+1}}\cdots\mathbf{y_m} \in S_2\right]\cdot\mathbb{P}\left[\mathbf{y_{i+1}}\cdots\mathbf{y_m} \in S_2\right] \\
&= P_i + \mathbb{P}\left[D'(O\ \mathbf{y_i}\mathbf{y_{i+1}}\cdots\mathbf{y_m}, i-1) = 1 | \mathbf{y_{i+1}}\cdots\mathbf{y_m} \in S_2\right]\cdot\mathbb{P}\left[\mathbf{y_{i+1}}\cdots\mathbf{y_m} \in S_2\right],
\end{aligned}$$

因此

$$\mathbb{P}\left[D'(O\ \mathbf{y_i}\mathbf{y_{i+1}}\cdots\mathbf{y_m}, i-1) = 1 | \mathbf{y_{i+1}}\cdots\mathbf{y_m} \in S_2\right]\cdot\mathbb{P}\left[\mathbf{y_{i+1}}\cdots\mathbf{y_m} \in S_2\right] \ge P_{i-1} - P_i \ge 1/(m\cdot|D|).$$

特别地，$\mathbb{P}[\mathbf{y_{i+1}}\cdots\mathbf{y_m} \in S_2] > 0$ 和 $\mathbb{P}[D'(O\,\mathbf{y_i}\,\mathbf{y_{i+1}}\cdots\mathbf{y_m}, i-1) = 1 | \mathbf{y_{i+1}}\cdots\mathbf{y_m} \in S_2] \ge 1/(m\cdot|D|)$，这意味着存在一个固定的 $(y_{i+1}\dots y_m) \in S_2$ 使得

$$\mathbb{P}\left[D'(O\ \mathbf{y_i}\ y_{i+1}\dots y_m, i-1) = 1\right] \ge 1/(m\cdot|D|).$$

现在，我们通过 $C(\mathbf{y_i}) := D'(O\,\mathbf{y_i}y_{i+1}\dots y_m, i-1)$ 定义另一个非确定性电路 C，其中 $\mathbf{y_i} \in \{0,1\}^{n+1}$ 是输入变量。由于 D' 在 N 中规模为次指数，且 n 与 N 多项式相关，因此 C 在 n 中规模为次指数。

现在我们论证，C 以所需的意义破坏了 b。对于无穷多个 n，

(1) $\mathbb{P}[C(\mathbf{y_i}) = 1] = \mathbb{P}[D'(O\,\mathbf{y_i}\ y_{i+1}\dots y_m, i-1) = 1] \ge 1/(m\cdot|D|)$，其中 $m\cdot|D|$ 是次指数级的 n；(2) $\mathbb{P}[C(b(\mathbf{x_i})) = 1] = \mathbb{P}[D'(O\ b(\mathbf{x_i})\ y_{i+1}\dots y_m, i-1) = 1] = 0$，因为 $y_{i+1}\dots y_m \in S_2$ 意味着不存在 $x_1, \dots, x_i$ 使得 $D(b(x_1)\dots b(x_i)y_{i+1}\dots y_m) = 1$。

由于 C 以以上意义破坏了 b，因此我们得出矛盾，这与 b是半比特这一假设相矛盾。

□

条件 $c < 1$ 在半比特拉伸算法 4.1中保证 $n = N^{1-c}$ 与 N 存在一个无穷单调序列 N 会产生一个无穷单调序列 n。

使这个证明成立的一个关键步骤是，在 $b(x_1)$ 中非确定性地猜测种子 x_1、.....、$x_{i-1}\dots b(x_{i-1})z_i$，能够保留 $b(x_1)\dots b(x_{i-1})z_i$ 的“随机性结构”，具体来说：当 $z_i = b(\cdot)$ 是伪随机时，非确定性猜测保留了 $b(\cdot)\dots b(\cdot)b(\cdot)$ 的形式（即，i 等长的伪随机块）；当 $z_i = y$ 真正随机时，则保留了 $b(\cdot)\dots b(\cdot)y$ 的形式（即，$i-1$ 等长的伪随机块后接一个长度相同的真正随机块 y）。对于常见的拉伸算法（例如递归地应用单比特生成器）而言，这类算法会生成指数多个新比特，但目前尚不清楚如何以尊重给定字符串“随机性结构”的方式利用非确定性。尽管如此，这种新的证明技巧或许能启发研究人员进一步探索半比特的可拉伸性。另一方面，也可能存在某个特定的半比特，无法通过通常用于超比特和强伪随机生成器的标准拉伸算法，拉伸为指数多个半比特。

4.1 在平均情形复杂性中的应用

本节将证明定理 4.2 蕴含时间受限柯尔莫哥洛夫复杂度零错误平均情形困难性的两个不同参数范围之间存在等价关系，针对的对象是NP/多项式时间 机器。

我们需要如下定义：针对一类判定问题$\mathcal{C} \subseteq 2^{\{0,1\}^N}$的击中集生成器是一个函数 $g\colon \{0,1\}^n \to \{0,1\}^N$，其中 $n < N$，使得 g的像能够击中（即与之相交） $\mathcal{C}$ 中所有足够稠密的集合 A （也就是说， $|A| \geq \frac{2^N}{N^{O(1)}}$）。

正如 Santhanam 所指出的 [San20]:

命题 4.3 ([San20])**.** 设 $n < N$。存在一个针对 $g\colon \{0,1\}^n \to \{0,1\}^N$且可在类 $\mathcal{D}$ 中计算的击中集生成器，针对 NP/poly 当且仅当存在一个半比特 $b\colon \{0,1\}^n \to \{0,1\}^N$， 可在 $\mathcal{D}$ (针对NP/poly) 中计算。

证明。若 $g\colon \{0,1\}^n \to \{0,1\}^N$ 是针对由非确定性多项式规模电路可判定的判定问题类 $\mathcal{C}$ 的击中集生成器，那么它也是一个半比特，因为不存在多项式规模 $|C| = N^{O(1)}$的机器 $C \in \mathcal{C}$能够破解 g；否则 $\mathbb{P}[C(U_N) = 1] \geq 1/N^{O(1)}$ 和 $\mathbb{P}[C(g(U_n)) = 1] = 0$，这与 g 是针对 $\mathcal{C}$的击中集生成器这一假设相矛盾。反过来，如果 b 是一个半比特，那么它也是针对 $\mathcal{C}$的击中集生成器，因为如果一个 $\mathcal{C}$ 中的电路 C 对足够稠密的一组输入输出 1，它也必定会对 b的像中的某个字符串输出 1；否则 C 就会攻破这个半比特。□

注意，半比特拉伸算法 4.1 也适用于可由一致多项式时间计算的半比特（拉伸算法是一致的，前提是原始半比特也是如此，因为它只需在输入的不同部分上应用半比特）。这对我们很重要，因为要讨论柯尔莫哥洛夫复杂度，我们需要机器的大小固定，即使输入长度发生变化也是如此。但请注意，证明拉伸算法能保持其困难性的论证要求对手 D 是非一致的（这就是在定理 4.5 中我们针对NP/多项式时间对手进行研究的原因）。

我们将字符串 t-受限柯尔莫哥洛夫复杂度定义为 x，记作 $\mathrm{K}^t(x)$，即满足以下条件的字符串 D 的最小长度：通用图灵机 $U(D)$（我们固定一台这样的通用机）在至多 t 的时间内运行并输出 x。有关时间受限柯尔莫哥洛夫复杂度的更多细节，以及字符串的时间受限柯尔莫哥洛夫复杂度的定义，请参见 [ABK+06] 中的定义 9（该定义实际上是在输入索引 i和字符串 x第 i 位 D 作为输入时，输出 U；但这不会改变我们的结果）。

回顾定义 2.1 中语言 $\mathrm{K}^t[s]$和 $\mathrm{K}^{\mathrm{poly}}[s(n)]$。

我们还需要精确定义针对NP/poly 类的零错误平均情形困难性（等价地，如定义 3.2所述，针对非确定性电路的零错误平均情形困难性）。

定义 4.4（针对 NP/poly 的零错误平均情形困难性）。如果存在一台 NP/poly 机器满足以下所有条件，我们就称语言 $L \in \{0,1\}^*$ 对 NP/poly 而言是零错误平均情形容易的：(i) 每条计算路径都会以 Yes、No 或未知状态终止；(ii) 对于给定输入 x ，不存在两条不同的计算路径分别以 Yes 和 No 终止；(iii) 如果给定 x 时存在一条以 Yes（分别为 No）终止的计算路径，我们就称该机器在输入 x 上回答 Yes（No）；(iv) 否则（也就是说，给定输入 x 时，所有计算路径都以未知状态终止），我们就称该机器不知道 x 的答案；(v) 机器在回答 Yes 或 No 时绝不会出错（另一方面，它可以回答未知状态

对于 L 中的成员或非成员均可）；(vi) 机器（正确地）至少在 L 中占多项式比例的输入上回答 Yes 或 No（即，对于每个充分大的 n，以及某个与 n 无关的固定常数 c，至少有 $2^n/n^c$ 比例的字符串属于 $L \cap \{0,1\}^n$）。如果 L 对 NP/poly 而言并非零错误平均情形容易，我们就称 L 是针对 NP/poly 的零错误平均情形困难。

主要的等价关系如下：

定理 4.5（平均情形时间受限柯尔莫哥洛夫复杂度的等价性）。$\mathrm{K}^{\mathrm{poly}}[n - O(1)]$ 相对于 NP/poly 机器是零错误平均情形困难的，当且仅当 $\mathrm{K}^{\mathrm{poly}}[n - o(n)]$相对于 NP/poly 机器是零错误平均情形困难的。

证明。 根据 定理 4.2 以及针对 NP/poly (命题 4.3) 的（一致多项式时间可计算）半比特与击中集生成器（HSGs）之间的等价关系，只需证明：对于规模函数s： $\mathbb{N} \to \mathbb{N}$ 满足 $s(n) < n - O(1)$ 时，$\mathrm{K}^{\mathrm{poly}}[s(n)]$ 相对于 NP/poly机器是零错误平均情形困难的，当且仅当存在一个一致多项式时间可计算的 HSG H： $\{0,1\}^{s(n)} \to \{0,1\}^n$，针对NP/poly。

($\Longleftarrow$) 假设 H: $\{0,1\}^{s(n)} \to \{0,1\}^n$是一个针对 NP/poly 的 HSG，且可在一致多项式时间内计算。那么，对于每个常数 k（与 n 无关）以及充分大的 n，$\mathrm{Im}(H) \cap \{0,1\}^n$与所有满足以下条件的 NP/poly 可计算集合 $A_n \subseteq \{0,1\}^n$ 相交：对于所有 n，均有 $|A_n| \geq 2^n/n^k$。我们需要证明，对于每个常数 c，$\mathrm{K}^{n^c}[s(n)] \subseteq \{0,1\}^n$ 对 NP/poly 而言是零错误平均情形困难的。我们证明，不存在 NP/poly 机器能在至少 $2^n/n^k$ 个来自 $\{0,1\}^n$的输入字符串上给出 Yes 或 No 中的一个正确答案，而在其余来自 $\{0,1\}^n$ 的输入字符串上回答“不知道”，并且不犯任何错误。反设存在这样的机器，那么就有一台 NP/poly 机器能对至少 $2^n/n^{O(k)}$ 个来自 $\{0,1\}^n$的输入字符串正确地回答 No；这是因为大多数输入字符串不具有较短的时间受限柯尔莫哥洛夫复杂度，也就是说，对于每个 n，有多项式比例的输入 $x \in \{0,1\}^n$不属于 $\mathrm{K}^{\mathrm{poly}}[s(n)]$, ，其中 $s(n)$介于 $|x|^\varepsilon$ 和 $|x|$ 之间（对某个常数 $0 < \varepsilon < 1$而言；参见 [ABK+06，第 2.6] 节及其中的参考文献）。因此，存在一台 NP/poly/ 机器，能够接受来自 $\{0,1\}^n$ 的至少 $2^n/n^{O(k)}$ 个“困难字符串”（即不属于 $\mathrm{K}^{\mathrm{poly}}[n]$的字符串），并拒绝 $\{0,1\}^n$中的所有其他字符串：具体而言，这台 NP/poly/ 机器会猜测一个见证，证明输入字符串是困难的；如果见证正确，就接受，否则就拒绝。

因此，我们得到一个与 H 作为针对 NP 的 HSG 相矛盾的结论：存在一种稠密 NP 语言，其中至少包含来自 $\{0,1\}^n$ 的 $2^n/n^{O(k)}$ 个“困难字符串”。但是，如果 D 是针对该语言的 NP 机器，那么 D 就破坏了 HSG H：对于 $\mathrm{Im}(H)$ 中的每个字符串，机器 D 都会拒绝，因为根据 H 可在一致多项式时间内计算这一假设，它具有较小的 $\mathrm{K}^{\mathrm{poly}}$ 复杂度（换句话说，根据 H 可在一致多项式时间内计算这一假设，$\mathrm{Im}(H)$ 中每个长度为 n 的字符串 x 都满足：对于某个与 n 无关的常数 r， $x \in \mathrm{K}^{n^r}[s(n) + O(1)],$ ）。因此， H 无法命中由 D 定义的稠密 NP/poly 集合，矛盾。

($\Longrightarrow$) 我们假设 $\mathrm{K}^{\mathrm{poly}}[s(n)]$ 针对 NP/poly 是零错误平均情形困难的。令 H :$\{0,1\}^{s(n)} \to \{0,1\}^n$ 为如下定义的映射：将输入 $x \in \{0,1\}^{s(n)}$ 送入一台通用图灵机，运行时间为 n^k，其中某个常数 k 与 n无关；通用机的输出是一个长度为 n 的字符串（如果算法在 n^k 步后仍未终止，则输出 n 个零组成的字符串 $0 \cdots 0$）。我们证明， H 是针对 NP/poly 的 HSG（根据假设，可在 uniform$n^{O(1)}$-时间内计算）。

反设 H 不是针对 NP/poly 的 HSG。因此，存在一台 NP/poly 机器 D，它接受 $\{0,1\}^n$ 中至少 $2^n/n^{O(1)}$ 个字符串，但拒绝 Im(H) 中的每个字符串。因此，存在一台 NP/poly 机器 D，它正确接受满足 $x \notin \mathrm{K}^{\mathrm{poly}}[s(|x|)]$, 的至少 $2^n/n^{O(1)}$ 个字符串$x \in \{0,1\}^n$，并拒绝 $\{0,1\}^n$ 中的所有其他字符串。这与我们的假设矛盾，因为我们可以构造一台 NP/poly 机器 D'，以零错误方式在平均意义下判定 $\mathrm{K}^{\mathrm{poly}}[s(|x|)]$：在 D 中，只需将肯定状态替换为肯定状态，并将拒绝状态替换为未知状态。

□

4.2 对证明复杂性生成器的应用

推论 4.6（拉伸证明复杂性生成器）。设 $b: \{0,1\}^n \to \{0,1\}^{n+1}$ 是一个可由 P/poly 计算的半位。设 $0 < c < 1$ 是一个常数，且 $\ell = n + n^c$。那么，存在一个证明复杂性生成器 $g: \{0,1\}^n \to \{0,1\}^\ell$，可由 P/poly 计算，使得对于每个命题证明系统，以至少 $1 - \frac{1}{\ell^{\omega(1)}}$ 的概率（取值于 $r \in \{0,1\}^\ell$），重言式 $\tau(g)_r$ 不存在大小为 poly(n) 的证明。

证明。 根据 定理 4.2，我们可以扩展半比特 b，从而得到新的半比特 $g: \{0,1\}^n \to \{0,1\}^\ell$。根据假设， b可在 P复杂度类/poly 中计算，并且根据 半比特拉伸算法 4.1 （即扩展半比特的算法；显然，该算法仅涉及在种子的不同子部分上多次应用原始半比特函数） g 也属于 P复杂度类/poly。

反设存在一个常数 k 和一个命题证明系统 R，对于集合 $S \subseteq \{0,1\}^\ell \setminus \mathrm{Im}(g)$ 中的所有 r，它都允许对重言式 $\tau(g)_r$给出大小为 poly(n) 的证明；也就是说，对于某个满足 $|T| \geq 2^\ell/\ell^{\omega(1)}$ 的 T，并非对 $\{0,1\}^\ell \setminus (\mathrm{Im}(g) \uplus T)$ 中的所有 r，重言式 $\tau(g)_r$ 都没有多项式规模的 R-证明；注意，$|\{0,1\}^\ell \setminus (\mathrm{Im}(g) \uplus T)| = 2^\ell(1 - 1/\ell^{\omega(1)} - 1/2^{\ell-n}) = 2^\ell(1 - 1/\ell^{\omega(1)})$。

由于 g 可在 P复杂度类/poly 中计算，重言式 $\tau(g)_r$ 的大小也为 poly(n)。令 D 为一个非确定性电路，它接收输入 $r \in S$，“猜测”一个 R-证明来证明 $\tau(g)_r$，并验证该证明是否正确。于是， D 的大小为 $n^{O(1)}$，并且我们有

$$\mathbb{P}_{y \in \{0,1\}^\ell}[D(y) = 1] \geq \frac{1}{\ell^k} \quad \text{and} \quad \mathbb{P}_{x \in \{0,1\}^n}[D(g(x)) = 1] = 0, \tag{3}$$

这与我们关于 g 是半比特的假设相矛盾。 □

5 非确定性可预测性

在 第 5.1 节中，我们回顾确定性情形下可预测性的概念。在第 5.2 节中，我们将介绍超越确定性情形的全新可预测性概念，并从可预测性的角度研究非确定性困难性。

在本节和下一节中，我们将采用略作修改的超比特定义，该定义类似于确定性情形下的定义 3.7：若生成器 $g: \{0,1\}^n \to \{0,1\}^{m(n)}$ 对于每个 D 属于 NP/poly中的每个 p以及所有充分大的 n，都满足

$$\mathbb{P}\left[D(U_{m(n)}) = 1\right] - \mathbb{P}\left[D(g(U_n)) = 1\right] < 1/p(n).$$

也就是说，如果 g 能抵御所有 NP/ 多项式区分器，那么 g 就是超比特（超比特原始定义中的安全性是针对所有次指数规模非确定性电路而言的，而这里的定义则是针对所有多项式规模非确定性电路）。

本节的贡献推动了我们对生成器非确定性困难性的理解。

5.1 基本确定性可预测性

我们首先回顾确定性情形下可预测性的概念，以及确定性不可预测性与生成器的超多项式困难性之间的等价关系。本节回顾的结果改编自 第 5.1 节 所述内容，源自 [Gol08]。

定义 5.1（概率系综）。概率系综（简称系综）是一个随机变量无限序列 $(Z_n)_{n\in\mathbb{N}}$。每个 Z_n 的取值范围为 $\{0,1\}^{l(n)}$，其中 $l(n)$ 与 n 呈多项式关系（即存在一个多项式 p，使得对每个 n，都有 $l(n) \le p(n)$ 且 $p(l(n)) \ge n$。

如果存在 $g \in P/poly$ 使得 $g(U_n) = Z_n$，我们就称一个系综是多项式生成的。从密码学的角度来看，本文只关注多项式生成的系综，因为它们易于生成，可用于模拟其他概率系综（例如均匀系综，下一定义将对此进行介绍）。因此，除非另有说明，从现在起我们考虑的每个系综都默认是多项式生成的。

定义 5.2（强伪随机系综）。概率系综 $(Z_n)_{n\in\mathbb{N}}$ 是强伪随机的，如果对于每个属于 P/poly 的算法 D、每个多项式 p，以及所有充分大的n，

$$|\mathbb{P}_{U_{m(n)}}[D(U_{m(n)}, 1^n) = 1] - \mathbb{P}_{Z_n}[D(Z_n, 1^n) = 1]| \le 1/p(n),$$

其中 $m = |Z_n|$ 和 U_m 是均匀分布

输入 1^n 允许 D 在 n 中以多项式时间运行，并将值 n告知 D。不过，为了简化记号，通常会省略 1^n，从现在起也将省略它。

我们注意到，如果 g 是强伪随机生成器，那么 $g(U_n)$ 就是一个强伪随机系综。我们将看到，对于 $g(U_n)$，强伪随机等价于不可预测。

定义 5.3（（确定性）可预测性）。如果存在一个属于 P/poly 的算法 $\mathcal{A}$、一个多项式 p、无穷多个 n，以及对其中每个 n，一个 $i(n) < |Z_n|$，使得系综 $(Z_n)_{n\in\mathbb{N}}$ 称为可预测的，或称为 P/poly-可预测的

$$\mathbb{P}\left[\mathcal{A}(Z_n[1\ldots i]) = Z_n[i+1]\right] \ge \frac{1}{2} + 1/p(n).$$

如果一个系综 $(Z_n)_{n\in\mathbb{N}}$ 不可预测，则称其具有不可预测性。

定理 5.4. 一个系综是强伪随机的，当且仅当它具有不可预测性。

我们将在下一节介绍上一定理的一个非确定性变体。

5.2 非确定性变体

本节中，我们提出四种超越确定性情形的不可预测性新概念，并基于这些新概念刻画生成器的超硬度。

在定义非确定性可预测性之前，我们强调在非确定性情形下，判定问题与单比特输出计算问题之间有一项重要区别。

如果对于每个输入$x \in \{0,1\}^n$，非确定性算法 $\mathcal{A}$ 的每个计算分支都会产生 $\{0,1,\perp\}$中的一个结果，其中 $\perp$ 表示失败，并且始终存在一个计算分支会产生 0 或 1，我们就称 $\mathcal{A}$ 是函数计算算法。如果对于某个 $c \in \{0,1\}$ ，在输入x时，每个计算分支都会产生 c 或 $\perp$，则 $\mathcal{A}(x) = c$ ；否则， $\mathcal{A}(x) = \perp$。因此，如果 $\mathcal{A}(x) = \perp$，那么就存在一个产生 0 的计算分支，以及另一个产生 1 的计算分支。如果对于每个 $x \in \{0,1\}^n$， $\mathcal{A}(x) \in \{0,1\}$，我们就称$\mathcal{A}$是全的。显然，全函数计算算法构成函数计算算法的一个子类。如果在定义定义 5.5 中的 $\cap$-不可预测性时，将算法限制为全算法，那么本节末尾总结 5.9 中的图示将保持不变。不作此限制的好处在于，$\cap$-不可预测性这一性质会略微增强，因此，作出此限制后，我们可以在总结 5.9 中得到对非确定性困难性的更紧确的“夹逼式”刻画。

我们还指出，判定问题属于 NP/poly ∩coNP/poly，当且仅当它可由一个（全的）非确定性多项式规模函数计算算法判定。

定义 5.5（非确定性可预测性）。

NP/poly-可预测性。若存在一个属于 NP/poly 的算法$\mathcal{A}$ 、一个多项式 p、无穷多个 n，以及对其中每个 n 都有一个 $i(n) < |Z_n|$ ，使得

$$\mathbb{P}\left[\mathcal{A}(Z_n[1 \ldots i]) = Z_n[i+1]\right] \geq 1/2 + 1/p(n).$$

若系综 $(Z_n)_{n\in\mathbb{N}}$ 不具备 NP/poly-可预测性，则称其为 NP/poly-不可预测的。.

coNP/poly-可预测性。若存在一个属于 coNP/poly 的算法 $\mathcal{A}$ 、一个多项式 p、无穷多个 n，以及对其中每个 n 都有一个 $i(n) < |Z_n|$ ，使得

$$\mathbb{P}\left[\mathcal{A}(Z_n[1 \ldots i]) = Z_n[i+1]\right] \geq 1/2 + 1/p(n).$$

若系综 $(Z_n)_{n\in\mathbb{N}}$ 不具备 coNP/poly-可预测性，则称其为 coNP/poly-不可预测的。

∪-可预测性。如果一个系综（Z_n）$_{n\in\mathbb{N}}$ 被称为 ∪-可预测的（ ∪是（NP/poly∪coNP/poly）-可预测的缩写），则它是 NP/poly-可预测的或 coNP/poly-可预测的。如果一个系综（Z_n）$_{n\in\mathbb{N}}$ 是∪-不可预测的，则它不是 ∪-可预测的。

∩-可预测性。如果存在一个多项式时间非确定性函数计算算法 $\mathcal{A}$、一个多项式 p、无穷多个 n，以及对于其中每个 n，都有一个 i（n） $< |Z_n|$ ，使得一个系综（Z_n）$_{n\in\mathbb{N}}$ 被称为 ∩-可预测的（ ∩是（NP/poly∩coNP/poly）-可预测的缩写），则

$$\mathbb{P}\left[\mathcal{A}(Z_n[1 \ldots i]) = Z_n[i+1]\right] \geq 1/2 + 1/p(n).$$

如果一个系综（Z_n）$_{n\in\mathbb{N}}$ 是 ∩-不可预测的，则它不是 ∩-可预测的。

备注。读者不应将具备 ∩可预测性误认为具备 NP/poly-可预测性 ∧ coNP/poly-可预测性。事实上，具备 ∩可预测性比具备 NP/poly-可预测性 ∧ coNP/poly-可预测性更强（见引理 引理 5.6）。换言之，对于一个系综 (Z_n)，具备 ∩不可预测性比具备 NP/poly-不可预测性∨ coNP/poly-不可预测性是更弱的性质。尽管如此，具备 ∩不可预测性仍然是一个非平凡性质，因为它比具备 P/poly-不可预测性更强（确定性算法是一个计算全函数的算法）。

引理 5.6。如果一个系综 (Z_n) 具有 $\cap$可预测性，那么它同时具有 NP/poly-可预测性和 coNP/poly-可预测性。

证明。假设存在一个多项式时间非确定性函数计算算法 $\mathcal{A}$、一个多项式 p、无穷多个 n，以及对于其中每个 n 都存在一个 $i(n) < |Z_n|$，使得

$$\mathbb{P}\left[\mathcal{A}(Z_n[1\dots i]) = Z_n[i+1]\right] \geq 1/2 + 1/p(n).$$

我们定义 $\mathcal{A}_1$ 属于 NP/多项式和 $\mathcal{A}_0$ 属于 coNP/多项式如下：

> 给定输入 $Y \in \{0,1\}^i$，$\mathcal{A}_j(j=0,1)$ 在输入 Y 上模拟 $\mathcal{A}$，以获得一个输出位 c。如果$c = \perp$，则 $\mathcal{A}_j$ 输出 $1-j$，否则 $\mathcal{A}_j$ 输出 c。

根据 $\mathcal{A}_j$ 的定义，如果 $\mathcal{A}(Y) \in \{0,1\}$，则 $\mathcal{A}_j(Y) = \mathcal{A}(Y)$。因此，对于 $j \in \{0,1\}$，我们有

$$\begin{aligned}
\mathbb{P}\left[\mathcal{A}_j(Z_n[1\dots i]) = Z_n[i+1]\right] &\geq \mathbb{P}\left[\mathcal{A}_j(Z_n[1\dots i]) = Z_n[i+1] \wedge \mathcal{A}(Z_n[1\dots i]) \in \{0,1\}\right] \\
&= \mathbb{P}\left[\mathcal{A}(Z_n[1\dots i]) = Z_n[i+1] \wedge \mathcal{A}(Z_n[1\dots i]) \in \{0,1\}\right] \\
&= \mathbb{P}\left[\mathcal{A}(Z_n[1\dots i]) = Z_n[i+1]\right] \\
&\geq 1/2 + 1/p(n).
\end{aligned}$$

□

命题 5.7。如果 $g: \{0,1\}^n \to \{0,1\}^{m(n)}$ 是超比特，则 $g(U_n)$ 是 $\cap$-不可预测的。

证明。假设存在矛盾，$g(U_n)$ 是 $\cap$-可预测的。存在一个多项式时间非确定性函数计算算法 $\mathcal{A}$、一个多项式 p、无穷多个 n，以及对于其中每个 n 都存在一个 $i(n) < |g(U_n)|$，使得

$$\mathbb{P}\left[\mathcal{A}(Z_i) = z\right] \geq 1/2 + 1/p(n),$$

其中 $Z_i = g(U_n)[1\dots i]$ 和 $z = g(U_n)[i+1]$。

我们构造非确定性算法 D 来区分 U_m 与 $g(U_n)$：

> 给定输入 $Y \in \{0,1\}^m$，D 在 $Y[1\dots i]$ 上运行 $\mathcal{A}$，以获得一个输出位 c。如果 $c = \perp$，D输出 0。当 $c \in \{0,1\}$时，如果 $c \neq Y[i+1]$，D 输出 1；否则，D 输出 0。

对于固定的 n，我们令 $f = \mathbb{P}\left[\mathcal{A}(U_i) = \perp\right]$，并用 b 表示一个随机位。回忆一下，$\mathcal{A}(U_i) = \perp$ 意味着 $\mathcal{A}$ 在 U_i 上的计算树以 0 和 1 为叶子。根据 D的定义，我们有：

$$\begin{aligned}
\mathbb{P}\left[D(U_m) = 1\right] &= \mathbb{P}\left[\mathcal{A}(U_i) = \perp\right] + \mathbb{P}\left[\mathcal{A}(U_i) \in \{0,1\} \wedge \mathcal{A}(U_i) \neq b\right] \\
&= \mathbb{P}\left[\mathcal{A}(U_i) = \perp\right] + \mathbb{P}\left[\mathcal{A}(U_i) \in \{0,1\}\right] \cdot \mathbb{P}\left[\mathcal{A}(U_i) \neq b | \mathcal{A}(U_i) \in \{0,1\}\right] \\
&= f + (1-f) \cdot 1/2 \\
&\geq 1/2.
\end{aligned}$$

另一方面：

$$\begin{aligned}
\mathbb{P}\left[D(g(U_n)) = 1\right] &= \mathbb{P}\left[\mathcal{A}(Z_i) = \perp\right] + \mathbb{P}\left[\mathcal{A}(Z_i) \in \{0,1\} \wedge \mathcal{A}(Z_i) \neq z\right] \\
&= \mathbb{P}\left[\mathcal{A}(Z_i) = \perp\right] + \left(\mathbb{P}\left[\mathcal{A}(Z_i) \in \{0,1\}\right] - \mathbb{P}\left[\mathcal{A}(Z_i) \in \{0,1\} \wedge \mathcal{A}(Z_i) = z\right]\right) \\
&= \mathbb{P}\left[\mathcal{A}(Z_i) = \perp\right] + \mathbb{P}\left[\mathcal{A}(Z_i) \in \{0,1\}\right] - \mathbb{P}\left[\mathcal{A}(Z_i) = z\right] \\
&= 1 - \mathbb{P}\left[\mathcal{A}(Z_i) = z\right] \\
&\leq 1/2 - 1/p(n).
\end{aligned}$$

因此，

$$\mathbb{P}\left[D(U_m)=1\right]-\mathbb{P}\left[D(g(U_n))=1\right]\geq 1/p(n).$$

□

命题 5.8. 若 $g(U_n)$ 是 $\cup$-不可预测的（即 NP/poly-不可预测或 coNP/poly-不可预测），其中 $g:\{0,1\}^n\to\{0,1\}^{m(n)}\in P/poly$ 且 $m(n)>n$，那么 g 是超比特。

证明。反设 g 不是超比特。对任意固定的 n，定义混合序列 $H_i=g(U_n)[0\ldots i]\cdot U_m[i+1\ldots m]$，并记 $Z_i=g(U_n)[1\ldots i]$ 和 $z_i=g(U_n)[i]$。于是，根据混合论证，存在一个属于 NP/poly 的算法 D、一个多项式 p、无穷多个 n，以及对其中每个 n，一个 i，使得

$$\mathbb{P}\left[D(H_i)=1\right]-\mathbb{P}\left[D(H_{i+1})=1\right]\geq 1/p(n).$$

因此，对每个这样的 n，都存在一个固定字符串 w，使得

$$\mathbb{P}\left[D(Z_ib,w)=1\right]-\mathbb{P}\left[D(Z_{i+1},w)=1\right]\geq 1/p(n),\qquad(*)$$

其中我们用 b 表示一个随机比特。此后可以省略书写 w n.

我们构造一个算法 $\mathcal{A}_1$ 属于 NP/poly，且 $\mathcal{A}_2$ 属于 coNP/poly，用来预测 z_{i+1} 基于 Z_i，如下所示:

Giv将 $Y_i\in\{0,1\}^i$ 作为输入， $\mathcal{A}_1$ 对 Y_i0w执行与 D 相同的操作； $\mathcal{A}_2$ 执行与 D 对 Y_i1w 执行相同的操作，但随后将得到的比特取反。

这些定义 of $\mathcal{A}_1,\mathcal{A}_2$ 意味着 $\mathcal{A}_1(Y_i)=D(Y_i0)$，且 $\mathcal{A}_2(Y_i)=D(\overline{Y_i1})$（$w$ 省略）。因此 e,

$$\begin{aligned}(1)&:=\mathbb{P}\left[\mathcal{A}_1(Z_i)=z_{i+1}\right]\\&=\mathbb{P}\left[D(Z_i0)=z_{i+1}\right]\\&=\mathbb{P}\left[D(Z_i0)=0\wedge z_{i+1}=0\right]+\mathbb{P}\left[D(Z_i0)=1\wedge z_{i+1}=1\right]\\&=\mathbb{P}\left[D(Z_iz_{i+1})=0\wedge z_{i+1}=0\right]+\mathbb{P}\left[D(Z_i\overline{z_{i+1}})=1\wedge z_{i+1}=1\right],\end{aligned}$$

and

$$\begin{aligned}(2)&:=\mathbb{P}\left[\mathcal{A}_2(Z_i)=z_{i+1}\right]\\&=\mathbb{P}\left[D(Z_i1)\neq z_{i+1}\right]\\&=\mathbb{P}\left[D(Z_i1)=0\wedge z_{i+1}=1\right]+\mathbb{P}\left[D(Z_i1)=1\wedge z_{i+1}=0\right]\\&=\mathbb{P}\left[D(Z_iz_{i+1})=0\wedge z_{i+1}=1\right]+\mathbb{P}\left[D(Z_i\overline{z_{i+1}})=1\wedge z_{i+1}=0\right].\end{aligned}$$

因此，

$$(1)+(2)=\mathbb{P}\left[D(Z_iz_{i+1})=0\right]+\mathbb{P}\left[D(Z_i\overline{z_{i+1}})=1\right].$$

As

$$\mathbb{P}\left[D(Z_iz_{i+1})=0\right]=1-\mathbb{P}\left[D(Z_{i+1})=1\right]$$

and

$$\begin{aligned}2\mathbb{P}\left[D(Z_i,b)=1\right]&=2\mathbb{P}\left[D(Z_i,b)=1\wedge b=z_{i+1}\right]+2\mathbb{P}\left[D(Z_i,b)=1\wedge b\neq z_{i+1}\right]\\&=2\mathbb{P}\left[b=z_{i+1}\right]\mathbb{P}\left[D(Z_i,b)=1|b=z_{i+1}\right]+2\mathbb{P}\left[b\neq z_{i+1}\right]\mathbb{P}\left[D(Z_i,b)=1|b\neq z_{i+1}\right]\\&=\mathbb{P}\left[D(Z_i,z_{i+1})=1\right]+\mathbb{P}\left[D(Z_i,\overline{z_{i+1}})=1\right],\end{aligned}$$

$$\begin{aligned}(1)+(2) &= (1-\mathbb{P}[D(Z_{i+1})=1]) + (2\mathbb{P}[D(Z_i b)=1] - \mathbb{P}[D(Z_{i+1})=1]) \\ &= 1 + 2(\mathbb{P}[D(Z_i b)=1] - \mathbb{P}[D(Z_{i+1})=1]) \\ &\geq 1 + 2/p(n) \text{ by } (*).\end{aligned}$$

因此，以下两种情况必有其一：(1) 对无穷多个 n，$= \mathbb{P}[\mathcal{A}_1(Z_i) = z_{i+1}] \geq 1/2 + 1/p(n)$；或 (2) 对无穷多个 n，$=\mathbb{P}[\mathcal{A}_2(Z_i) = z_{i+1}] \geq 1/2 + 1/p(n)$。也就是说，系综 (Z_n) 要么是 NP/poly 可预测的，要么是 coNP/poly 可预测的。 □

总结 5.9。 我们将 引理 5.6、命题 5.7 和 命题 5.8 总结为以下不等式链。这里，$A \leq B$ 表示：如果一个系综 $g(U_n)$ 具有性质 B，那么它也具有性质 A (参见引言中的 第 2.2 节，其中以更直观的方式展示了这些不等式)：

$$\text{P/多项式不可预测性} \leq \cap\text{-不可预测性} \tag{4}$$
$$\cap\text{-不可预测性} \leq \text{超多项式非确定性困难性} \leq \cup\text{-不可预测性} \tag{5}$$
$$\cap\text{-不可预测性} \leq \text{NP/多项式不可预测性} \leq \cup\text{-不可预测性。} \tag{6}$$
$$\cap\text{-不可预测性} \leq \text{coNP/多项式不可预测性} \leq \cup\text{-不可预测性。} \tag{7}$$

与 定理 5.4不同，后者从可预测性的角度精确刻画了标准困难性，而迄今为止，我们对非确定性情形下超硬度的刻画仍不准确。我倾向于认为，我们可能无法用可预测性对超硬度作出精确刻画，因为我们将在第 6.2 节看到，超硬度不仅涉及算法（例如用于证明随机性的算法，或用于预测的算法）在值域元素（即 y 使得 $\mathbb{P}[Z_n = y] > 0$）上的行为，也可能涉及在非值域元素（即 y 使得 $\mathbb{P}[Z_n = y] = 0$）上的行为。换言之，上述不等式中至少有一些很可能是严格不等式。

开放问题。摘要中的不等式能否进一步细化或分类？例如，超硬度与 NP/poly-不可预测性和 $\vee$coNP/poly-不可预测性之间是否存在某种关系？

6 超核心谓词

在下一小节中，我们将回顾单向函数和硬核谓词的概念，以及它们与强伪随机生成器之间已知的联系。在 第 6.2 节 中，我们将介绍超核心谓词的概念，并探讨它与超比特之间的联系。这将推动我们提出一个合理的定义，用于刻画非确定性情形下的单向函数。

6.1 单向函数和硬核谓词

我们首先回顾单向函数和硬核谓词的标准概念，以及强伪随机生成器存在与某个单向函数存在硬核之间的等价性（详情请参见 [Gol08]）。

定义 6.1（单向函数）。如果对于每个 $\mathcal{A}$ in P/poly、每个多项式 $p(\cdot)$，以及所有充分大的 n，函数 $f\colon \{0,1\}^* \to \{0,1\}^*$ in P/poly 被称为单向函数，当且仅当

$$\mathbb{P}_{x\in\{0,1\}^n}[\mathcal{A}(f(x),1^n) \in f^{-1}(f(x))] < \frac{1}{p(n)}.$$

输入 1^n 出于技术原因而存在。这使算法 $\mathcal{A}$ 能够在$n = |x|$上以多项式时间运行；当 f 大幅缩短其输入时（例如，当 $|f(x)| = O(log|x|)$ 时），这一点十分重要。当不需要辅助输入 1^n 时（例如，当 f 是长度保持的），我们可以将其省略。直观地说，对于所有高效算法，如果概率取自输入分布，则当函数 f in P/poly 的逆通常“难以”求出时，它就是单向的。

众所周知，单向函数存在与强伪随机生成器存在是等价的：

定理 6.2. 单向函数存在，当且仅当强伪随机生成器存在。

由于我们可以从任意单向函数构造一个保长度单向函数，定理 6.2 也可以表述为：

定理 6.3. 保长度单向函数存在，当且仅当强伪随机生成器存在。

定理 6.2 的逆向推论相当直接，而正向推论实际上依赖于一个与单向函数密切相关的概念，称为硬核谓词。即使假设硬核谓词可以从种子 x中额外生成一位，已知的定理 6.3 正向推论的证明仍相当复杂。

定义 6.4（硬核谓词）。如果不存在算法 $\mathcal{A} \in$ P/poly、多项式 $p(\cdot)$以及无穷多个 n，使得满足以下条件，那么谓词 $b : \{0,1\}^* \to \{0,1\} \in$ P/poly 称为函数 f 的硬核：

$$\mathop{\mathbb{P}}_{x\in\{0,1\}^n}[\mathcal{A}(f(x), 1^n) = b(x)] \geq \frac{1}{2} + \frac{1}{p(n)}.$$

换句话说，如果 b 能抵御所有 P/poly 算法的（预测意义上的）攻击，那么它就是 f 的硬核；这种安全性可能源于 f 的信息损失（例如，$b(x) = x[n]$ 是 $f(x) = x[1 \dots (n-1)]$的硬核），也可能源于求逆 f 的困难。

如果 b 是任意 f 的硬核，那么 $\mathbb{P}[b(x) = 0] \approx \mathbb{P}[b(x) = 1] \approx 1/2$（否则，我们只需输出常数 $argmax_{b\in\{0,1\}}(\mathbb{P}[b(x) = b]$，就能预测 b）。

单向函数与硬核谓词通过构造“通用”硬核而“配对”：

定理 6.5. 对于任意单向函数 f，x 与 y 的模 2 内积记作 $\langle x, y\rangle$，是 $f'(x, y) := (f(x)\ y)$ 的硬核谓词。

特别地，如果 f 是长度保持的，那么 f' 也是。实际上，在该定理中，我们还应在奇数长度的输入 $(x,\ y,\ b)$ 上定义 f' 和 b，其中 $|x| = |y|$ 和 b 是单比特；但这种情况通常会被省略，因为我们可以轻而易举地“忽略”最后一位（采用引理 3.14 中的相同思路），并定义 $f'(x,\ y,\ b) = (f(x)\ y,\ b)$ 和 $b(x,\ y,\ b) = \langle x,\ y\rangle$。

该定理指出，每个单向函数 f“本质上”都有相同的硬核谓词（这里具有硬核谓词的并非 f 本身，而是上述的 f' ）。很容易证明，不存在一个对每个单向函数都具有硬核性质的谓词 b。

因此，定理 6.3 可以重新表述为：

定理 6.6。存在某个保长度单向函数的硬核 b，当且仅当存在一个强伪随机生成器 g。

我们将在下一节中给出上一条定理的一个非确定性变体（定理 6.13）。

6.2 非确定性变体

本节将介绍超核心谓词的概念，并探讨它与超比特之间的联系，以及哪些函数可能具有超核心、哪些函数可能没有超核心。

定义 6.7（超核心谓词）。若谓词 $b:\{0,1\}^n \to \{0,1\}$ 属于 P/poly，则称其为函数 $f:\{0,1\}^n \to \{0,1\}^{m(n)}$ 的超核心，条件是不存在算法 $\mathcal{A}_1$ 属于 NP/poly、算法 $\mathcal{A}_2$ 属于 coNP/poly、多项式 $p(\cdot)$，以及无穷多个 n，使得

$$\mathbb{P}_{x\in\{0,1\}^n}[\mathcal{A}_1(f(x),1^n)=b(x)=0]+\frac{1}{2}\mathbb{P}_{y\in\{0,1\}^m}[\mathcal{A}_1(y,1^n)=1]\geq\frac{1}{2}+\frac{1}{p(n)} \qquad (\star)$$

$$nor$$

$$\mathbb{P}_{x\in\{0,1\}^n}[\mathcal{A}_2(f(x),1^n)=b(x)=1]+\frac{1}{2}\mathbb{P}_{y\in\{0,1\}^m}[\mathcal{A}_2(y,1^n)=0]\geq\frac{1}{2}+\frac{1}{p(n)}. \qquad (\diamond)$$

为方便起见，我们为上述公式定义一些缩写（省略输入 1^n）：

$$\begin{aligned}
&①(\mathcal{A}_1) := \mathbb{P}_{x\in\{0,1\}^n}[\mathcal{A}_1(f(x))=b(x)=0],\\
&②(\mathcal{A}_2) := \mathbb{P}_{x\in\{0,1\}^n}[\mathcal{A}_2(f(x))=b(x)=1],\\
&③(\mathcal{A}_1) := \tfrac{1}{2}\mathbb{P}_{y\in\{0,1\}^{m(n)}}[\mathcal{A}_1(y)=1],\\
&④(\mathcal{A}_2) := \tfrac{1}{2}\mathbb{P}_{y\in\{0,1\}^{m(n)}}[\mathcal{A}_2(y)=0].
\end{aligned}$$

我们在此旨在提出硬核谓词的非确定性变体，并建立超比特的存在性与这种非确定性变体的存在性之间的关系。按照上述规定的意义，超核心能够抵御非确定性和余非确定性预测器。 不等式（$\star$）可以改写为

$$\mathbb{P}_{x\in\{0,1\}^n}[\mathcal{A}_1(f(x),1^n)b(x)=0]\geq\frac{1}{2}\mathbb{P}_{y\in\{0,1\}^m}[\mathcal{A}_1(y,1^n)=0]+\frac{1}{p(n)},$$

并且 s 将左侧按关于 $b(x)=0$ 的条件概率进行拆分，得到

$$\mathbb{P}_{x\in\{0,1\}^n}[b(x)=0]\mathbb{P}_{x\in\{0,1\}^n}[\mathcal{A}_1(f(x),1^n)=0|b(x)=0]\geq\frac{1}{2}\mathbb{P}_{y\in\{0,1\}^m}[\mathcal{A}_1(y,1^n)=0]+\frac{1}{p(n)}.$$

正如我们将在引理 6.8中看到的，超核心也是硬核，这意味着$\mathbb{P}_{x\in\{0,1\}^n}[b(x)=0]\approx 1/2$。因此，对于一个NP/poly-预测器 $\mathcal{A}_1$ 要满足（$\star$），它必须满足以下等价条件，其中某个多项式为 $p(n)$:

$$\mathbb{P}_{x\in\{0,1\}^n}[\mathcal{A}_1(f(x),1^n)=0|b(x)=0]\geq\mathbb{P}_{y\in\{0,1\}^m}[\mathcal{A}_1(y,1^n)=0]+\frac{1}{p(n)}.$$

直观而言，这个条件意味着，如果提供给 NP/poly-预测器 $\mathcal{A}_1$ 的输入 y 确实表示某个 $f(x)$，使得 $b(x)=0$，那么 NP/poly-预测器 $\mathcal{A}_1$ 就必须对其更加敏感，通过输出 0 来检测到它（相比接收到随机输入时）。($\diamond$) 与 ($\star$) 互为对偶。请注意，在此定义下，非确定性对手和余非确定性对手无法以任何平凡的方式（例如，输出常数 0 或 1）满足相应的不等式。

我们注意到，如果 b是 f的超核心，那么 b也是任意“缩短后的” f的超核心。具体而言，如果 $i:\mathbb{N}\to\mathbb{N}$ 满足对每个 n，都有 $i(n)\leq|f(1^n)|$，那么 b就是 $f'(x):=f(x)[1\dots i(|x|)]$的超核心，因为 $f'(x)$提供的信息少于 $f(x)$。

正如超比特也是强伪随机生成器一样，我们有:

引理 6.8. 如果 b 是函数 $f\colon \{0,1\}^n \to \{0,1\}^{m(n)}$, b 的超核心，那么它是 f 的硬核。

证明。假设为了导出矛盾， b 不是 f的硬核。那么存在 $C \in$P/poly、一个多项式 p以及无穷多个 n, 使得 $\mathbb{P}\left[C(f(U_n)) = b(U_n)\right] \geq 1/2 + 1/p(n)$。注意， C同时属于 NP/poly 和 coNP/poly。由于

$$①(C) + ②(C) + ③(C) + ④(C) = \mathbb{P}\left[C(f(U_n)) = b(U_n)\right] + 1/2 \geq 1 + 1/p(n),$$

要么 1 (①(C) $+$ ③(C) $\geq 1/2 + 1/(2p(n))$, 要么 2 (②(C) $+$ ④(C) $\geq 1/2 + 1/(2p(n))$). □

现在我们证明，对于某个保长度函数，可以从一个超核心构造出一个超比特:

命题 6.9。如果 b 是函数 $f\colon \{0,1\}^n \to \{0,1\}^n \in$P/poly 的超核心，那么 $g(x)$:=$f(x)b(x)$是一个超比特。

证明。假设为了导出矛盾， g 不是超比特。那么存在一个区分器 $D \in$NP/poly、一个多项式 p以及无穷多个 n，使得:

$$\mathbb{P}\left[D(U_{n+1}) = 1\right] - \mathbb{P}\left[D(g(U_n)) = 1\right] \geq 1/p(n).$$

我们定义算法 $\mathcal{A}_1$ in NP/poly 和算法 $\mathcal{A}_2$ in coNP/poly 来预测 b ，如下所示：假设 $Y \in \{0,1\}^n$ 作为输入给定。 $\mathcal{A}_1$ 在 $Y0$ 上运行 D ，并在 D 接受时接受。 $\mathcal{A}_2$ 在 $Y1$ 上运行 D ，并在 D 拒绝时接受。

定义 的意思是 $\mathcal{A}_1(Y) = D(Y0)$ 且 $\mathcal{A}_2(Y) = D(Y1)$。现在，我们有 e:

$$\begin{aligned}
①(\mathcal{A}_1) + ②(\mathcal{A}_2) &= \mathbb{P}\left[\mathcal{A}_1(f(U_n)) = b(U_n) = 0\right] + \mathbb{P}\left[\mathcal{A}_2(f(U_n)) = b(U_n) = 1\right] \\
&= \mathbb{P}\left[D(f(U_n)0) = 0 \wedge b(U_n) = 0\right] + \mathbb{P}\left[D(f(U_n)1) = 0 \wedge b(U_n) = 1\right] \\
&= \mathbb{P}\left[D(f(U_n)b(U_n)) = 0 \wedge b(U_n) = 0\right] + \mathbb{P}\left[D(f(U_n)b(U_n)) = 0 \wedge b(U_n) = 1\right] \\
&= \mathbb{P}\left[D(f(U_n)b(U_n)) = 0\right] \\
&= 1 - \mathbb{P}\left[D(g(U_n)) = 1\right],
\end{aligned}$$

and

$$\begin{aligned}
③(\mathcal{A}_1) + ④(\mathcal{A}_2) &= \frac{1}{2}\mathbb{P}\left[\mathcal{A}_1(U_n) = 1\right] + \frac{1}{2}\mathbb{P}\left[\mathcal{A}_2(U_n) = 0\right] \\
&= \frac{1}{2}\mathbb{P}\left[D(U_n0) = 1\right] + \frac{1}{2}\mathbb{P}\left[D(U_n1) = 1\right] \\
&= \mathbb{P}\left[D(U_{n+1}) = 1\right].
\end{aligned}$$

因此，对于无穷多个 n,

$$①(\mathcal{A}_1) + ②(\mathcal{A}_2) + ③(\mathcal{A}_1) + ④(\mathcal{A}_2) = 1 + \mathbb{P}\left[D(U_{n+1}) = 1\right] - \mathbb{P}\left[D(g(U_n)) = 1\right] \geq 1 + 1/p(n),$$

这意味着以下两种情况之一成立：对于无穷多个 n, 1 ($\mathcal{A}_1$) + ③($\mathcal{A}_1$) $\geq 1/2 + 1/(2p(n))$; ②或者对于无穷多个 n, 2 ($\mathcal{A}_2$) +4 ($\mathcal{A}_2$) $\geq 1/2+1/(2p(n))$。这与 b 是 f的超核心这一假设矛盾。 □

在证明命题 6.11（它是命题 6.9的逆命题）之前，我们需要以下辅助引理:

引理 6.10. 若 $g\colon \{0,1\}^n \to \{0,1\}^{n+1}$ 是一个强伪随机生成器，并定义 $f(x)b(x) := g(x)$，其中 $b(x)$ 是 $g(x)$ 的最后一位，那么 b 是 $f \in P/poly$的硬核。

证明。反设， b 不是 f的硬核。那么存在 $\mathcal{A} \in$P/poly、一个多项式 p以及无穷多个 n，使得

$$\mathbb{P}\left[\mathcal{A}(f(U_n)) = b(U_n)\right] \geq 1/2 + 1/p(n).$$

我们按如下方式构造 D 来攻破 g：

给定输入 $Y \in \{0,1\}^{n+1}$，当且仅当 $\mathcal{A}(Y[1..n]) = Y[n+1]$时， D 输出 1。令 b 表示一个随机比特。那么，对于无穷多个 n,

$$\begin{aligned}\mathbb{P}\left[D(g(U_n)) = 1\right] - \mathbb{P}\left[D(U_{n+1}) = 1\right] &= \mathbb{P}\left[\mathcal{A}(f(U_n)) = b(U_n)\right] - \mathbb{P}\left[b = \mathcal{A}(U_n)\right] \\ &\geq 1/2 + 1/p(n) - 1/2 \\ &= 1/p(n).\end{aligned}$$

□

现在，我们可以证明:

命题 6.11。若 $g\colon \{0,1\}^n \to \{0,1\}^{n+1}$ 是一个超比特，并定义 $f(x)b(x) := g(x)$，其中$b(x)$是 $g(x)$的最后一位，则 b 是 $f \in$P/poly的超核心。

证明。反设， b 不是 f的超核心。我们给出以下情形的证明：存在一个 $\mathcal{A} \in$ NP/poly、一个多项式 p以及无穷多个 n，使得1 $(\mathcal{A})$ +3 $(\mathcal{A})$ $\geq 1/2 + 1/p(n$。另一种情形的证明类似。

我们定义一个区分器 D，如下所示，用来攻破 g：给定 $Y \in \{0,1\}^{n+1}$ 作为输入， D 在 $Y[1\ldots n]$ 上运行 $\mathcal{A}$，得到 $\mathcal{A}$ 的一个输出位 c。然后，D 当且仅当 $Y[n+1] = 0$ 和 $c = 1$ 时输出 1。

于是，对于无穷多个 n，我们有

$$\begin{aligned}&\mathbb{P}\left[D(U_{n+1}) = 1\right] - \mathbb{P}\left[D(f(U_n)b(U_n)) = 1\right] \\ &= \mathbb{P}\left[\mathcal{A}(U_n) = 1 \wedge U_1 = 0\right] - \mathbb{P}\left[\mathcal{A}(f(U_n)) = 1 \wedge b(U_n) = 0\right] \\ &= 1/2 \cdot \mathbb{P}\left[\mathcal{A}(U_n) = 1\right] - \left(\mathbb{P}\left[b(U_n) = 0\right] - \mathbb{P}\left[\mathcal{A}(f(U_n)) = 0 \wedge b(U_n) = 0\right]\right) \\ &= ①(\mathcal{A}) + ③(\mathcal{A}) - \mathbb{P}\left[b(U_n) = 0\right] \\ &\geq 1/2 + 1/p(n) - \mathbb{P}\left[b(U_n) = 0\right].\end{aligned}$$

令 $1/s(n) = |\mathbb{P}\left[b(U_n) = 0\right] - 1/2|$。由于 g 是一个强伪随机生成器，根据引理 6.10， b 是 g 的硬核。因此，对于每个多项式 q （特别是对于 $q(n) = 2p(n)$）以及每个充分大的 n，$1/s(n) \leq 1/q(n)$。因此，对于无穷多个 n,

$$\mathbb{P}\left[D(U_{n+1}) = 1\right] - \mathbb{P}\left[D(f(U_n)b(U_n)) = 1\right] \geq 1/2 + 1/p(n) - \mathbb{P}\left[b(U_n) = 0\right] \geq 1/2p(n)。$$

□

结合命题 6.9 和 命题 6.11，我们得出了定理 6.6的以下非确定性变体。

定理 6.12. 存在某个长度保持的 $f \in$P复杂度类/poly的超核心 b，当且仅当存在超比特 g。

我们称函数 $f\colon \{0,1\}^n \to \{0,1\}^{m(n)}$ 是非收缩的，$m(n) \geq n$ 对每个 n 都成立。由于 (1) b 是某个长度保持函数的超核心，当且仅当 b 是某个非收缩函数的超核心；并且 (2) 存在一个超比特，当且仅当存在超比特（也就是说，我们可以通过 [Rud97], 拉伸超比特），因此我们也可以将定理 6.12 表述为：

定理 6.13。P/poly 中存在某个非收缩函数的超核心，当且仅当存在超比特。

虽然我们可以将“长度保持”条件放宽为“非收缩”，但“非收缩”这一要求并非多余，因为很容易构造出某个函数 f 的超核心 b，而该函数会缩短其输入；然而，是否存在超比特仍然未知。定义 $f(x) = x[1]$和 $b(x) = x[-1]$，那么 b 是 f 的超核心，因为从 $\{0,1\}$ 到 $\{0,1\}$ 只有四个函数，而且没有一个能以所需的意义根据 $f(x) \in \{0,1\}$ 预测 b（事实上，对于任意 $c\colon \{0,1\} \to \{0,1\}$，$①(c) + ③(c) = ② (c) + ④(c) = 1/2$）。

此时，我们可能想进一步了解函数 f 的单向性与拥有超核心之间的关系。回顾一下我们在确定性情形中已知的结论：（1）如果 f 拥有硬核 b，那么 f 不一定是单向函数，因为拥有硬核可能是由于 f 的信息损失；（2）如果 f 是单向函数，那么我们可以构造出$f'(x,y) = (f(x)\ y)$ 的硬核 b。

第一点在非确定性情形中也同样成立。前面提到的 $f(x) = x[1]$显然不是单向函数，却拥有超核心 $b(x) = x[-1]$，这是由信息的大幅损失造成的。一个更有趣的问题是，由超比特 g 构造出的 命题 6.11 中的 f 是否拥有超核心 $b(x) = g(x)[-1]$，以及它是否为单向函数。（我们已证明 f 拥有超核心 $b(x) = g(x)[-1]$。）

开放问题 6.14. 如果 g是一个超比特，那么 $f(x) := g(x)[1 \ldots n]$ $(n = |x|)$ 是单向函数吗？ ?

然而，关于第二点，由于成为超核心的要求比成为硬核更强，因此未必存在意义上的“通用”超核心：也就是说，某个谓词 b（例如，$b(x,y) = \langle x,y\rangle$）未必对每个 $f'(x,y) = (f(x)\ y)$ 都是超核心，其中 $f\colon \{0,1\}^n \to \{0,1\}^{m(n)}$ 是一个单向函数。那么，一个自然的问题是：

开放问题 6.15. 假设 f 是一个单向函数。如果我们希望 $\langle x,y\rangle$ 成为$f'(x,y) = (f(x)\ y)$的超核心，那么 f 还需要具备哪些其他性质（如果有的话）？

相反，关于第二点，我们可以证明，即使某些 f’ 可能是单向的，它们也不可能拥有超核心。这些 f’ 包括某些无穷多次“主要为一对一”的函数。更精确地说，若 x 满足 $|f^{-1}(f(x))| = 1$，则称其为类型 1；否则，x 为类型 2。我们将 $T_1(n)$ 定义为类型 1 的 $x \in \{0,1\}^n$ 集合，并将 $T_2(n)$ 定义为类型 2 的 $x \in \{0,1\}^n$ 集合。现在，我们来证明：

定理 6.16。给定常数整数 $c \geq 0$ 和 f：若 $\{0,1\}^n \to \{0,1\}^{n+c}$ 属于 P复杂度类/poly，则存在无穷多个 n’ 以及一个多项式 p，使得，

$$\frac{|T_1|}{2^n} \geq \frac{2^{1+c}}{2^{1+c}+1} + \frac{1}{p(n)},$$

那么 f 不具有超核心。

特别是，当 f 是长度保持的（即 $c = 0$）时，不等式变为：

$$\frac{|T_1|}{2^n} \geq \frac{2}{3} + \frac{1}{p(n)}.$$

证明。假设 f 具有所述性质。反设 b 是 f的超比特。我们按如下方式构造两个算法 $\mathcal{A}_j(j = 0, 1)$，其中 $\mathcal{A}_0$ 是余非确定性的，而 $\mathcal{A}_1$ 是非确定性的，用来预测 b：

> 给定输入 $y \in \{0,1\}^n$， $\mathcal{A}_j$ 猜测一个 x'，使得 $f(x') = y$。如果 $\mathcal{A}_j$ 未能猜中这样的 x'，它就输出 $1 - j$。否则，它输出 $b(x')$。

注意，如果对某个 $x \in T_1$， $y = f(x)$ 成立，那么总是恰好存在一个正确猜测 x'（即 x 本身），使得 $f(x') = y = f(x)$，而且每当猜中这样的 x'时， $b(x') = b(x)$。因此，1 $(\mathcal{A}_1) = \mathbb{P}[\mathcal{A}_1(f(x)) = b(x) = 0] \geq \mathbb{P}[\mathcal{A}_1(f(x)) = b(x) = 0 \wedge x \in T_1] = \mathbb{P}[b(x) = 0 \wedge x \in T_1]$，并且对于某个， ③ $\mathcal{A}_1 = \frac{1}{2}\mathbb{P}[\mathcal{A}_1\ y = 1] \geq \frac{1}{2}\mathbb{P}[y = f\ x \wedge b\ x = 1 \quad x \in T_1]$ ② $\mathcal{A}_0 \geq \mathbb{P}[b\ x = 1 \wedge x \in T_1]$ ④ $\mathcal{A}_0 \geq \frac{1}{2}\mathbb{P}[y = f\ x \wedge b\ x = 0 \quad x \in T_1]$ ( ) ( ) ( ) ( )。类似地，( ) ( )，并且对于某个，( ) ( ) ( )。

因此，对于无穷多个 n， 1 $(\mathcal{A}_1)$ + ②$(\mathcal{A}_0)$ + ③$(\mathcal{A}_1)$ + ④$(\mathcal{A}_0) = \mathbb{P}[x \in T_1] + \frac{1}{2}\mathbb{P}[y \in f(T_1)] = \frac{|T_1|}{2^n} + \frac{1}{2} \cdot \frac{|T_1|}{2^{n+c}} = \frac{2^{1+c}+1}{2^{1+c}} \cdot \frac{|T_1|}{2^n} \geq 1 + \frac{1}{p\ n()}$ 成立，但这意味着 1 $(\mathcal{A}_1)$ + ③$(\mathcal{A}_1) \geq \frac{1}{2} + \frac{1}{2p\ (n)}$或 2 $(\mathcal{A}_0)$ + ④$(\mathcal{A}_0) \geq \frac{1}{2} + \frac{1}{2p\ (n)}$成立。 □

直观地说，这类 f 没有超核心的原因是：这样的 f 保留了大部分信息（因此可以猜出唯一的原像）。在该定理中，我们不能简单地将 c 放宽为 n 的任意多项式，因为这样做的话，命题中不等式的右侧可能会超过 1。当 f 是长度保持的情况下，还可以利用前面提到的其他结果来证明该结论，具体如下。假设，为了导出矛盾， b 是 f 的超核心，则根据命题 6.9， $g(x) := f(x)b(x)$ 是一个超比特。然而，我们随后可以轻易地破解 g：通过猜测一个 $x \in \{0,1\}^n$ 来证明给定 $y \in \{0,1\}^{n+1}$ 的随机性，使得$f(x) = y[1 \ldots n]$，并检验 $b(x) = y[n+1]$；如果 $b(x) \neq y[n+1]$，则 y 是随机的。

我们给出一个较弱但更简洁的推论，来自定理 6.16：

推论 6.17。如果 $f \in$P/poly 是长度保持且一对一的（或者至少对无穷多个n都一对一），那么 f 没有超核心。

这一推论与以下结果形成对比：长度保持且一对1 $f \in P/poly$ 的函数可能具有硬核（事实上，如果 b 是长度保持且一对1 $f \in P/poly$函数的硬核，那么 $g(x) := f(x)b(x)$ 是一个强伪随机生成器 [Gol08]）。因此，这一推论表明，可能存在并非超比特的强伪随机生成器。

除了上述关于哪些函数可以或不可以具有超核心，以及单向函数与具有超核心之间关系的探讨之外，另一个核心问题是：

开放问题。什么是对“非确定性单向函数”的合理定义（如果存在这样的定义）？ ?

对开放问题 6.14 和开放问题 6.15 给出令人满意的答案，将有助于阐明这一问题。一个可能的候选是“具有超核心的单向函数”，但这个问题仍需进一步探索。

7 结论与未来方向

本工作首次对（密码学情形下的）非确定性伪随机性进行了系统研究，并以 Rudich 提出的原语为基础[Rud97]。我们的研究揭示了一个富有潜力的研究方向。接下来，我们提出或概述一些值得探索的未来方向和开放问题。

- 我们给出了一种算法，能够针对任意给定的半比特实现次线性扩展。如果我们想要更多，也许会想知道能否扩展得更长。具体而言，我们能否将一个半比特扩展为线性数量的半比特、多项式数量的半比特，甚至扩展为具有指数级半比特硬度的伪随机函数生成器（PRFG）？即使假设我们确实拥有某些 n-半比特 b：$\{0,1\}^n \to \{0,1\}^{2n}$，我们仍不清楚能否通过任何用于构造 PRFG 的标准算法或一种新颖算法，构造出具有指数级半比特硬度的 PRFG。

- 我们能否进一步细化或分类总结 5.9中的不等式？我们还能否用不可预测性来刻画半硬度（即，半硬度在该格中的正确位置是什么？）

- 单向函数是最核心的密码学原语之一。因此，一个自然的问题是：非确定性安全单向函数的合理定义是什么？（一种初步尝试可能是“具有超核心的单向函数”，但这一提议还需要进一步验证。）

- 如果 g 是一个超比特，那么 $f(x) := g(x)[1...n]$ $(n = |x|)$ 是单向函数吗？

- 假设 f 是一个单向函数。如果我们希望 $\langle x, y\rangle$ 成为 $f'(x, y) = (f(x)\ y)$ 的超核心，那么 f 还需要具备哪些其他性质（如果有的话）？

更好地回答上述问题，将有助于我们更好地理解非确定性情形下生成器的困难性，并为开放问题“半比特的存在是否意味着超比特也存在”提供启示。

除非对小型电路进行 PAC 学习是可行的，否则半比特存在

本节将完整阐述 Pich 关于 [Pic20], 的研究成果。他提出了通过攻破伪随机生成器来构造 PAC 学习算法。这些成果对于非确定性对手下伪随机性的理论基础十分重要，因为它们为半比特的存在提供了依据。

A.1 PAC 学习

我们提出 PAC 学习的以下非一致版本：

定义 A.1（非一致 PAC 学习）。如果存在一个随机化预言机族 $L = \{D_n\} \in \mathcal{D}$，使得对于每个由 $\mathcal{C}$ 计算的族 f：$\{0,1\}^n \to \{0,1\}$，以及每个足够大的 n，均有：那么电路类 $\mathcal{C}$ 可由电路类 $\mathcal{D}$（在均匀分布下）以误差 ε、置信度 δ 学习：

1. $\mathbb{P}_{w}[L^f(1^n, w)\ (1-\varepsilon)-approximates\ f] \geq \delta$，其中 w 是输入到 L^f 的随机输入位，而 L^f 的输出 $L^f(1^n, w)$ 是 f 的近似器 f' 的字符串表示。$L^f(1^n, w)$ 可以是以下类型的一致或非一致算法的描述：确定性、非确定性、余非确定性和随机化。

2. 对于任意 f 和 w，$L^f(1^n, w)$ 都是 $\mathcal{D}$-可评估的：存在另一个电路族 $E \in \mathcal{D}$，使得对于 L 的每个可能输出 $L^f(1^n, w)$，以及作为输入提供给 E 的任意 $x \in \{0,1\}^n$，E 都能在 x 上计算 $L^f(1^n, w)$。当 $L^f(1^n, w)$ 是（余）非确定性或随机化的时，允许 E 分别是（余）非确定性或随机化的。

对于这样的学习器 L，我们也称 $\mathcal{C}$ 可由 L 学习，或者称每个 $C \in \mathcal{C}$ 都可由 L 学习。

备注。

1. 如果 L^f 以某种方式选择向预言机门发出的查询集合，我们就称它使用成员查询。如果 L^f 发出的查询集合是从均匀分布中随机抽取的，我们就称它使用均匀分布的随机样本。上述学习模型足够通用，既能涵盖这两类学习器，也能涵盖任何形式的混合。在本文中，我们只考虑使用均匀分布的随机样本的学习器。因此，“学习”始终指使用均匀分布的随机样本进行学习。2. 所定义的学习模型也足够通用，可以涵盖关于其他分布的学习（即我们改变条件 (1) 中 w 的分布）。当 $\mathcal{D}$ 包含 P/poly 时，在任意多项式生成的分布上学习等价于在均匀分布上学习。3. 一个可能引起困惑之处是省略条件 (2)。然而，条件 (2) 在此不可或缺。原因是，如果没有这一限制，P/poly 就可以被高效学习，而人们普遍认为事实并非如此（例如，参见 [RS21]）。通过应用 [BEHW87] 中确立的奥卡姆剃刀定理，可以很容易地证明，若没有第二个条件，P/poly 是可学习的。直观地说，学习器可以通过记住样本并将所有“学习”推迟到假设评估阶段，从而高效地学习 P/poly。4. 当学习器在 $\mathcal{D}$ 中的置信度 δ 和准确率 $1-\varepsilon$ 相对于 $\mathcal{D}$ 非可忽略时，可以用标准方法将它们提升到小于 1 的常数（参见 [KV94]）。例如，当 $\mathcal{D} = P/poly$ 时，只需让 δ 和 $1-\varepsilon$ 分别达到 $p(n)$ 和 $1/2 + q(n)$，其中 p 和 q 是任意多项式。除此处的备注外，增强并非本文主题，故不再赘述。

A.2 基于不存在性假设的学习

我们回顾 [BFKL94] 中的一种构造：对于正整数 m 和电路 $C: \{0,1\}^n \to \{0,1\}$，定义生成器 $G_{m,C}: \{0,1\}^{mn} \to \{0,1\}^{mn+m}$，它将 m 个 n 位字符串 $x_1, ..., x_m$ 映射为 $x_1, C(x_1), ..., x_m, C(x_m)$。

我们将 [BFKL94] 中关于平均情形学习的定理 7，重新表述为适用于我们 PAC 学习框架的如下引理：

引理 A.2。给定一个电路类 $\mathcal{C}$，如果存在一个 m 和一个大小为 $s(n)$ 的电路 D，使得对于每个 $C \in \mathcal{C}$

$$\mathbb{P}\left[D(y)=1\right]-\mathbb{P}\left[D(G_C(x))=1\right] \geq 1/s,\ where\ G_C=G_{m,C},$$

那么，存在一个随机多项式时间（关于 n 和 $|\langle D\rangle|$，其中 $\langle D\rangle$ 是 D的给定字符串表示）的算法 L，它以 $1/2m^2s$ 的置信度学习 $\mathcal{C}$，误差至多为 $1/2-1/2ms$。

特别地，如果 D 是非确定性电路或余非确定性电路，则允许 L 的输出是非确定性算法或余非确定性算法。

证明。给定任意 $C \in \mathcal{C}$，L 随机选择一个 $i \in [m]$、比特 r_1、……、r_m以及 n 位字符串 x_1、……、x_m（x_i除外），在 x_1、……、x_{i-1}上查询 C，得到 $C(x_1)$、……、$C(x_{i-1})$，并输出 C'，它按如下方式预测 C：给定任意 n 位输入 x_i，C' 在 $(x_1$，$C(x_1)$、……、x_{i-1}，$C(x_{i-1})$ x_i，r_i、……、x_m，$r_m)$ 上模拟 D，得到一个输出位 p_i。如果 $p_i=1$，C' 输出 $\bar{r_i}$；如果 $p_i=0$，C' 输出 r_i。

接下来，我们要证明 L 确实以所需的方式学习 C。给定随机比特 $r_1,...,r_m$以及随机的 n位字符串 $x_1,...,x_m$，我们定义 $p_i := D(x_1, C(x_1)\ ..., x_{i-1}, C(x_{i-1})\ x_i, r_i, ..., x_m, r_m)$。那么（注意：这里我们应用了混合论证），

$$\begin{aligned}
1/s &\leq \mathbb{P}\left[D(x)=1\right]-\mathbb{P}\left[D(G_C(x))=1\right] \\
&= \mathbb{P}\left[p_1=1\right]-\mathbb{P}\left[p_m=1\right] \\
&= \sum_{i=1}^{m-1}(\mathbb{P}\left[p_i=1\right]-\mathbb{P}\left[p_{i+1}=1\right])
\end{aligned}$$

因此，存在某个 i 使得 $\mathbb{P}\left[p_i=1\right]-\mathbb{P}\left[p_{i+1}=1\right] \geq 1/ms$，所以 i L 选择的对象满足 $\mathbb{P}\left[p_i=1\right]-\mathbb{P}\left[p_{i+1}=1\right] \geq 1/ms$ 的概率为 $\geq 1/m$。当 L 成功选择了这样一个 i 时，

$$\begin{aligned}
&\mathop{\mathbb{P}}_{\substack{r_1,\ldots,r_m \\ x_1,\ldots,x_m}}[C'(x_i)=C(x_i)] \\
&= \mathbb{P}\left[p_i=1, r_i \neq C(x_i)\right]+\mathbb{P}\left[p_i=0, r_i=C(x_i)\right] \\
&= \frac{1}{2}\mathbb{P}\left[p_i=1|r_i \neq C(x_i)\right]+\frac{1}{2}\mathbb{P}\left[p_i=0|r_i=C(x_i)\right] \\
&= \frac{1}{2}\mathbb{P}\left[p_i=1|r_i \neq C(x_i)\right]+\frac{1}{2}(1-\mathbb{P}\left[p_i=1|r_i=C(x_i)\right]) \\
&= \frac{1}{2}+\frac{1}{2}\mathbb{P}\left[p_i=1|r_i \neq C(x_i)\right]-\frac{1}{2}\mathbb{P}\left[p_i=1|r_i=C(x_i)\right] \\
&= \frac{1}{2}+\frac{1}{2}\mathbb{P}\left[p_i=1|r_i \neq C(x_i)\right]+(\frac{1}{2}\mathbb{P}\left[p_i=1|r_i=C(x_i)\right]-\mathbb{P}\left[p_i=1|r_i=C(x_i)\right]) \\
&= \frac{1}{2}+(\mathbb{P}\left[p_i=1, r_i \neq C(x_i)\right]+\mathbb{P}\left[p_i=1, r_i=C(x_i)\right])-\mathbb{P}\left[p_i=1|r_i=C(x_i)\right] \\
&= \frac{1}{2}+\mathbb{P}\left[p_i=1\right]-\mathbb{P}\left[p_{i+1}=1\right] \\
&\geq \frac{1}{2}+\frac{1}{ms}
\end{aligned}$$

令 $p = \mathop{\mathbb{P}}_{r_1 \ldots、、x_1 \ldots \quad r_m、、x_m\, except\, x_i}[\mathop{\mathbb{P}}_{x_i}[C'(x_i) = C(x_i)] >= 1/2 + 1/2ms]$。由于存在 $0 \leq a < 1/2$ 使得 $1/2 + 1/2ms + a$ 表示预测器 C' 在以下情况下的平均准确率

r_1、…、r_m、x_1、…、x_{i-1}、x_{i+1}、…、x_m 满足 $\mathbb{P}[C'(x_i) = C(x_i)] >= 1/2 + 1/2ms$，我们有$x_i$

$$p\left(\frac{1}{2}+\frac{1}{2ms}+a\right)+(1-p)\left(\frac{1}{2}+\frac{1}{2ms}\right) \geq \mathop{\mathbb{P}}_{\substack{r_1,\ldots,r_m \\ x_1,\ldots,x_m}}[C'(x_i)=C(x_i)] \geq \frac{1}{2}+\frac{1}{ms},$$

因此

$$p \geq \frac{1}{2ms}\cdot\frac{1}{a} \geq \frac{1}{ms}.$$

因此，L 输出一个$(1/2 + 1/2ms)$-准确的 C'，置信度为 $1/m^2s$。 □

注。当 D 是非确定性时：(1) 若 $r_i = 0$，则 L^C 学得的 C' 也是非确定性的；(2) 若 $r_i = 0$，则 L^C 学得的 C' 是余非确定性的。

本节余下部分将分别在不存在无穷多次半比特和不存在半比特的假设下，推导学习算法。显然，不存在无穷多次半比特的假设强于不存在半比特的假设。我们还将看到，根据我们的证明策略，更强的假设会带来更好的学习结果，其含义稍后将会明了。

以下定理的证明曾在[Pic20]中作过概述：

定理 A.3. 假设不存在无穷多次半比特。那么，对于每个 $c \in \mathbb{N}$，$Circuit[n^c]$都可由 $Circuit[2^{n^{o(1)}}]$ 学习（通过获取随机样本），置信度为 $1/2^{n^{o(1)}}$，误差不超过$1/2 - 1/2^{n^{o(1)}}$；其中，学习器可以生成一个非确定性或余非确定性算法来近似目标函数。

证明。假设不存在无穷多次半比特，且 $c \in \mathbb{N}$。存在一种针对 $Circuit[n^c]$ 的编码方案，使得每个具有 n 位输入的 $C \in Circuit[n^c]$ 都可以编码为长度为 $\leq d = d(n)$ 的字符串 $\langle C\rangle$，其中 $d(n)$ 是一个多项式。令 $m = n^d + 1$，并考虑生成器 G：$\{0,1\}^{mn+n^d} \to \{0,1\}^{mn+m}$，它将最后 n^d 位输入解释为某个 $C \in Circuit[n^c]$ 的描述，然后在剩余的 mn 位输入上计算 $G_{m,\,C}$。注意，G 属于 P/poly（只要针对 $Circuit[n^c]$ 的编码方案是合理的）。由于 G 不是无穷多次半比特，因此存在一个规模为次指数的非确定性电路 D，使得对于所有充分大的 n，

$$\mathbb{P}[D(y)=1] \geq 1/|D| \quad \text{and} \quad \mathbb{P}[D(G(x))=1] = 0.$$

特别地，对于每个 $C \in Circuit[n^c]$，$\mathbb{P}[D(G(\langle C\rangle, x) = 1] = 0$。也就是说，$\mathbb{P}[D(G_{m,\,C}(x)) = 1] = 0$ 对每个 $C \in Circuit[n^c]$ 都成立。因此，

$$\mathbb{P}[D(y)=1] - \mathbb{P}[D(G_{m,C}(x))=1] \geq 1/|D|.$$

因为 $m = n^d + 1$ 是关于 n 的多项式，且 $|D|$ 是次指数，因此根据 引理 A.2，$\mathcal{C}$ 可以由规模为 $2^{n^{o(1)}}$、置信度为 $1/2^{n^{o(1)}}$ 且误差不超过 $1/2 - 1/2^{n^{o(1)}}$的随机化电路族学习。 □

如果我们将假设弱化为不存在半比特，那么通过相同的证明，我们可以以如下所述的较弱意义学习 $Circuit[n^c]$：

定理 A.4。假设不存在半比特。那么对于每个 $c \in \mathbb{N}$，都存在一个无限单调序列 $\{n_i\} \subseteq \mathbb{N}$，使得对于每个 $n \in \{n_i\}$，$Circuit[n^c]$ 都可以由 $Circuit[2^{n^{o(1)}}]$ 学习（通过获取随机样本），其置信度为 $1/2^{n^{o(1)}}$，误差至多为 $1/2 - 1/2^{n^{o(1)}}$；其中学习器可以生成一个近似目标函数的非确定性或余非确定性算法。

˜由于 $NP/qpoly$-自然证明（比 NP/poly-自然证明的存在性假设更弱）的存在会排除无穷多次超比特 [Rud97]，所给出的定理 A.3，因此我们有：

推论 A.5。假设无穷多次半比特的存在意味着无穷多次超比特的存在。如果˜存在一个对 P/poly 有用的 $NP/qpoly$-自然性质，那么对于每个 $c \in \mathbb{N}$，$Circuit[n^c]$都可以由 $Circuit[2^{n^{o(1)}}]$ 学习（通过取随机样本），置信度为 $1/2^{n^{o(1)}}$，误差不超过$1/2 - 1/2^{n^{o(1)}}$；其中，学习器可以生成一个逼近目标函数的非确定性或余非确定性算法。

致谢

我们谨向 Jan Pich 致谢，他建议我们研究半比特，并对其工作提供了许多澄清说明 [Pic20]。我们感谢 Rahul Santhanam 与我们进行了非常有益的讨论，并特别提到了 定理 4.5 中出现的潜在应用。最后，我们还要感谢 Hanlin Ren 对稿件提出的宝贵意见，以及 Oliver Korten 和 Yufeng Li 与我们进行的进一步讨论。

参考文献

[ABK+06] Eric Allender、Harry Buhrman、Michal Kouck´y、Dieter van Melkebeek 和 Detlef Ronneburger。随机字符串的力量。《SIAM计算杂志》，35(6):1467–1493，2006。2.1.2，4.1，4.1[ABRW04]Michael Alekhnovich、Eli Ben-Sasson、Alexander A. Razborov 和 Avi Wigder- son。命题证明复杂性中的伪随机生成器。《SIAM计算杂志》，34(1):67–88，2004。（初步版本发表于第41届计算机科学基础年度研讨会论文集（加利福尼亚州雷东多海滩，2000年）。）2.1.3[ACGS88] W. Alexi、B. Chor、O. Goldreich 和 C. P. Schnorr。RSA和Rabin函数：某些部分与整体一样难。《SIAM计算杂志》，17(2):194–209，1988。1.1[BEHW87] A. Blumer、A. Ehrenfeucht、D. Haussler 和 M. K. Warmuth。奥卡姆剃刀。《信息处理快报》，24(6):377–380，1987年4月。3[BFKL94] A. Blum、M. Furst、M. Kearns 和 R. J. Lipton。基于困难学习问题的密码学原语。载于 Douglas R. Stinson（编），《密码学进展——CRYPTO'93》，第278–291页，柏林、海德堡，1994。Springer Berlin Heidelberg。2.4，A.2[BM84] M. Blum 和 S. Micali。如何生成密码学安全的伪随机比特序列。《SIAM计算杂志》，13(4):850–864，1984年11月。1.2，2.2[BOV07] Boaz Barak、Shien Jin Ong 和 Salil P. Vadhan。密码学中的去随机化。《SIAM计算杂志》，37(2):380–400，2007。1.1

[DVV16] Akshay Degwekar、Vinod Vaikuntanathan 和 Prashant Nalini Vasudevan。细粒度密码学。密码学电子预印本档案，论文 2016/580，2016。https://eprint.iacr.org/2016/580。3[GGM86] O. Goldreich、S. Goldwasser 和 S. Micali。如何构造随机函数。《ACM学报》，33(4):792–807，1986年8月。1.2[GL89] O. Goldreich 和 L. A. Levin。适用于所有单向函数的硬核谓词。载于第21届ACM计算理论年度研讨会，STOC '89，第25–32页，美国纽约州纽约市，1989年。计算机协会。（文档），2.3[GM84] S. Goldwasser 和 S. Micali。概率加密。《计算机与系统科学杂志》，28(2):270–299，1984。1.2，2.1.4[Gol01] Oded Goldreich。《密码学基础 I：基本工具》。剑桥：剑桥大学出版社，2001。3.4[Gol08] O. Goldreich。《计算复杂性：概念视角》。剑桥大学出版社，2008。5.1，6.1，6.2[HILL99] J. H˚astad、R. Impagliazzo、L. A. Levin 和 M. Luby。由任意单向函数构造伪随机生成器。《SIAM计算杂志》，28(4):1364–1396，1999。（文档），2.3[IN89] R. Impagliazzo 和 M. Naor。可证明具有子集和同等安全性的高效密码方案。载于第30届计算机科学基础年度研讨会，第236–241页，1989。1.1[Kal05] B. S. Kaliski。椭圆曲线与密码学：伪随机比特生成器及其他工具。麻省理工学院博士论文，2005。1.1[Kra04] Jan Krajíˇcek。对偶弱鸽巢原理、伪满射函数与电路下界的可证明性。《符号逻辑杂志》，69(1):265–286，2004。1.1，2.1.3[Kra10] Jan Krajíˇcek。随机变量与证明复杂性中的强迫法，伦敦数学会讲义系列第382卷。剑桥出版社，2010。2.1.3[KV94] M. J. Kearns 和 U. V. Vazirani。《计算学习理论导论》。麻省理工学院出版社，美国马萨诸塞州剑桥，1994。4[KvM02] Adam R. Klivans 和 Dieter van Melkebeek。除非多项式时间层次坍塌，否则图非同构具有亚指数规模的证明。《SIAM计算杂志》，31(5):1501–1526，2002。1.1[LP20] Yanyi Liu 和 Rafael Pass。论单向函数与柯尔莫哥洛夫复杂度。载于 Sandy Irani （编），第 61 届电气电子工程师学会计算机科学基础年度研讨会，FOCS 2020，美国达勒姆，2020年11月16–19日，第1243–1254页。电气电子工程师学会，2020。2.3

[Lub96] M. Luby.《伪随机性与密码学应用》，第1卷。普林斯顿大学出版社，1996年。4[NW94] N. Nisan 和 A. Wigderson。困难性与随机性。《计算机与系统科学杂志》，49(2):149–167，1994年。3.4，3.4，3[OS17] I. C. Oliveira 和 R. Santhanam。学习算法、电路下界与伪随机性的关联。载于第32届计算复杂性会议论文集，CCC '17，德国达格施图尔，2017年。Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik。2.4[Pic20] J. Pich。基于电路下界的学习算法。CoRR，abs/2012.14095，2020年。2.4，A，A.2，A.2[PS19] Jan Pich 和 Rahul Santhanam。为什么证明复杂性下界难以证明？载于第60届IEEE计算机科学基础年会（FOCS 2019），2019年11月9日至12日，美国马里兰州巴尔的摩，2019年。1.1[Raz15] Alexander A. Razborov。对 k-DNF 归结和多项式演算归结具有困难性的伪随机生成器。《数学年刊》，181:415–472，2015年。2.1.3[RR97] A. A. Razborov 和 S. Rudich。自然证明。《计算机与系统科学杂志》，55(1):24–35，1997年。（文档），1.1，1.2，3.4，3.4[RS21] N. Rajgopal 和 R. Santhanam。P/Poly以外可学习性的结构。载于 Mary Wootters 和 Laura Sanità（编），《近似、随机化与组合优化：算法与技术》（APPROX/RANDOM 2021），莱布尼茨国际信息学论文集（LIPIcs）第207卷，第46:1–46:23页，德国达格施图尔，2021年。Schloss Dagstuhl – Leibniz-Zentrum fuer Informatik。2.4，3[Rud97] S. Rudich。超比特、半比特与NP/qpoly-自然证明。《计算机与系统科学杂志》，55:204–213，1997年。（文档），1.1，1.1，1.2，1.3，1.4，1.5，1.2，2，2.1，2.1.1，3.5，3.5，3.10，3.5，4，6.2，7，A.2[San20] Rahul Santhanam。伪随机性与最小电路规模问题。载于 Thomas Vidick（编），第11届理论计算机科学创新会议（ITCS 2020），2020年1月12日至14日，美国华盛顿州西雅图，第151卷 LIPIcs，第68:1–68:26页。Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik，2020年。（文档），1.1，2.1，2.1.2，4.1，4.3[San22] Rahul Santhanam。私人通信，2022年。2.1.2[SU05] Ronen Shaltiel 和 Christopher Umans。适用于所有最小熵的简单提取器及一种新的伪随机生成器。《ACM学报》，52(2):172–216，2005年。1.1[SvM23] Nicollas Sdroievski 和 Dieter van Melkebeek。Arthur–Merlin协议的实例级困难性与随机性权衡。《计算复杂性电子研讨会》（ECCC），2023年。1.1[Val84] L. G. Valiant。可学习对象理论。《ACM通讯》，27(11):1134–1142，1984年11月。2.4

[Val85] L. G. Valiant。析取的合取学习。载于第九届国际人工智能联合会议论文集——第1卷，IJCAI’85，第560–566页，美国加利福尼亚州旧金山，1985年。摩根·考夫曼出版社。 2.4[VS84] L. G. Valiant 和 J. C. Shepherdson。演绎学习 [与讨论]。《伦敦皇家学会哲学汇刊：A辑，数学与物理科学》，312(1522):441–446，1984年。2.4[Yao82] A. C. Yao。陷门函数理论与应用。载于第二十三届计算机科学基础年度研讨会，FOCS ’82，第80–91页，1982年。(文档)，1.2，2.2，4