arXiv:2611.33126v1 [cs.CC] 24 Nov 2026

关于输入无关的无歧义层次的说明

Edward A. Hirsch*

2026年11月21日

摘要

无歧义层次的定义与多项式层次类似；不过，所有见证都必须是唯一的。不同版本的区别在于使用预言机的方式。我们考察由类**OUP**构造的输入无关无歧义层次，其中所有长度相同的正输入都有相同的唯一见证：带语言预言机的层次**OUH**，以及通过宽松方式访问承诺预言机的层次**OUH•**；后者是 Hirsch 和 Sokolov（信息处理快报，2015）提出的宽松无歧义层次的输入无关版本。我们证明，**OUH•** 中的语言可通过带有**ONP**预言机的确定性多项式时间算法判定；对于**OUH**中的语言，预言机查询可以非自适应地进行。

此外，**OUH** 中的语言可通过带有计数型**NP**预言机的确定性多项式时间算法判定；**OUH•** 中的语言可在**P**/**OptP**中判定。也就是说，它是**P**/**poly** 的一个子类，其中辅助信息由一个函数生成，该函数最大化非确定性机器在一元输入上的输出。我们证明，**P**/**OptP** 可以用**OGTD** 和**OUGTD** 生成器–测试器–解码器类来刻画；这些类近期由 Hirsch 和 Volkovich（2026）研究。特别是，**P**/**OptP** 包含于他们所谓的“终极”Karp–Lipton坍缩类**OUM**中。

我们还观察到若干与**OUP**相关的其他事实：(1) 由输入无关的无歧义点算子定义的层次会塌缩；(2) 在**OUP** 计算中，对**prOUP** 的一次宽松查询可以替换为对**OUP**的一次查询；(3) **SOUP** $=$ **OUP**$\cap$**co-OUP**，其中**SOUP**是 Hirsch、Volkovich 和 Williams（2026）最近研究的同见证类。

AI 使用声明。 本文中的所有概念和论述均由作者提出；作者还与 AI 工具 GPT6-Astra 和 Claude Opus 5.5 分享了证明思路，并与它们讨论了证明。此后，Opus 5.5 协助作者检索文献（作者承认，正因如此，他阅读的经典 文献数量前所未有），并协助起草文本。最终版本由作者在与 Opus 5.5 的长时间对话中完成；特别是，作者核验并编辑了上述工具撰写的所有部分，并对论文内容负全部责任。

1 引言

无歧义层级。 至多有一条接受路径的非确定性多项式时间计算构成类 **UP** 由[1] Valiant 定义。无歧义层级 [LR94, NR98] 的定义方式与多项式层次类似；不过，所有见证都必须是唯一的。这些层次在使用预言机的方式上存在细微差异。层次 $\mathbf{UH} = \mathbf{UP} \cup \mathbf{UP}^{\mathbf{UP}} \cup \mathbf{UP}^{\mathbf{UP}^{\mathbf{UP}}} \cup \ldots$ [NR98] 使用对语言的预言机访问，并且计算需要

*计算机科学系，Ariel University，以色列。电子邮件：edwardh@ariel.ac.il。本研究在以色列国、移民吸收部和科学家吸收中心的支持下开展。[1]此处及下文中，关于经典复杂性类的定义，请读者参阅复杂性动物园（complexityzoo.net）或 [HO02，附录 A]，，或（对于最基本的定义）参阅 [AB09]。

仅针对正确的预言机才是无歧义的。层次 $\mathcal{UH}$ [LR94，NR98] 对承诺问题 [CHV93] 使用受保护的预言机访问：禁止对承诺集之外进行查询（Groll- mann 和 Selman [GS88] 将此类归约称为智能归约）。无歧义交替层次 AUPH [LR94] 由具有有界交替次数的无歧义交替机通过单次查询预言机访问定义，或者由交替无歧义点算子定义 [NR98]（Niedermeier 和 Rossmanith 将这种等价性归功于 Hemaspaandra）。这些层次逐层满足 AUPH $\subseteq$UH $\subseteq \mathcal{UH}$，并且 $\mathcal{UH} \subseteq$SPP [LR94]；此外，UH 和 $\mathcal{UH}$ 包含于无歧义交替多项式时间类 UAP [CGRS04], ，而后者又包含于 SPP [NR98]。Hemaspaandra 和 Rothe [HR97] 将它们的结构与 UP 存在稀疏图灵完全集联系起来，而 Spakowski 和 Tripathi [ST07，ST09]研究了它们的相对化结构。

Hirsch 和 Sokolov [HS15] 放宽了预言机访问的定义，并考虑了宽松无歧义层次 prUH$_\bullet$：允许对承诺集之外的输入进行查询，但对于此类查询的所有可能答案，计算都必须是正确且无歧义的。Cai、Hemachandra 和 Vysko˘c [CHV93] 以“任意失败访问”（arbitrary failure access）之名，针对确定性机器引入了这种访问承诺问题的模式；Chakaravarthy 和 Roy [CR11] 将其用于查询 prMA 和 prAM 的确定性计算中；当 Valiant–Vazirani 定理（见 [AB09，定理 17.18]）表述为 NP $\subseteq$ RP$^{\mathrm{prUP}}$时，也指的是这种模式。类 UH$_\bullet$（即 prUH$_\bullet$中的语言类）包含 UH 和 $\mathcal{UH}$，并且按层级划分仍包含于 PH 中。Hirsch 和 Sokolov 将 Toda 定理的第一部分 PH $\subseteq$ BP$\cdot\oplus$P $\subseteq$ P$^{\mathrm{PP}}$（见 [AB09，第 17.4]节）加强为 PH $=$ BP $\cdot$ prUH$_\bullet$。他们提出 prUH$_\bullet \subseteq$ prSPP [HS15，第 4], 节的问题；该问题最近已由 Miloschewsky、Podder 和 Rudolph [以肯定的方式解决，见引理 3.6]。

输入无关计算。在输入无关计算中，见证可以依赖输入的长度，但不能依赖输入本身。Chakaravarthy 和 Roy [CR06] 定义了 $\mathrm{S}_2^{\mathrm{P}}$ 的输入无关版本 $\mathrm{O}_2^{\mathrm{P}}$；Fortnow、Santhanam 和 Williams [FSW09] 定义了 NP 和 MA 的输入无关版本 ONP 和 OMA；Goldreich 和 Meir [GM15]则将输入无关证明系统作为 P/poly 的一种一致视角进行了研究。这些类都包含于 P/poly；例如，NP $\subseteq$ P/poly 当且仅当 NP $=$ ONP [FSW09，命题 5]。Hirsch、Volkovich 和 Williams [HVW26] 最近引入了输入无关无歧义计算类 OUP，其中所有长度相同的正输入都具有相同的唯一见证；他们还引入了 OMIN 类，它等于 Korten 和 Pitassi [KP24] 所研究的 $\mathrm{L}_2^{\mathrm{P}}$ 类的输入无关版本 $\mathrm{OL}_2^{\mathrm{P}}$ [HV26a]。他们证明了 OUP $\subseteq$ OMIN，并研究了无关层次 OH $=$ ONP $\cup$ ONP$^{\mathrm{ONP}} \cup \ldots \subseteq \mathrm{O}_2^{\mathrm{P}}$。自然也可以研究输入无关无歧义层次（这正是本文所做的）：带语言谕示的层次 OUH $=$ OUP $\cup$ OUP$^{\mathrm{OUP}} \cup \ldots$，以及宽松层次 prOUH$_\bullet$；我们用 OUH$_\bullet$ 表示 prOUH$_\bullet$ 中的语言类。

多阶段计算。Hirsch 和 Volkovich [HV26b]引入了生成器–测试器–解码器计算。对于长度为 n 的输入，生成器是一台带有预言机的多项式时间转导器，它根据 1^n计算出一个辅助信息字符串；测试器在看不到输入的情况下验证该辅助信息；多项式时间解码器则使用该辅助信息判定输入。OGTD$[\mathcal{A},\mathcal{B},\mathcal{C}]$ 类由以这种方式判定的语言组成，其中生成器属于 $\mathcal{A}$，测试器属于 $\mathcal{B}$，解码器属于 $\mathcal{C}$；而在 OUGTD$[\mathcal{A},\mathcal{B},\mathcal{C}]$ 中，对于每个 n，测试器恰好接受一个辅助信息字符串（见第 2.5 节）。一个重要的特殊情形记作 OGTD$[\mathcal{X}]$：生成器属于 FP$^{\mathcal{X}}$，解码器属于 P，且不需要测试器。显然，OGTD$[\mathcal{X}] \subseteq$P$^{\mathcal{X}}$，因为一台带有来自 $\mathcal{X}$ 的预言机的多项式时间机器可以先计算辅助信息，再运行解码器。

这类两阶段计算数十年来一直受到研究。Karp 和 Lipton 定义了建议类 $\mathcal{C}/f$，允许 $\mathcal{C}$ 类的计算模型在所有长度为 n的输入上使用相同的可信（且可能不可计算的）长度为 $f(n)$的辅助信息字符串。特别地，P/poly $=$OGTD[poly $*$，P],，其中 poly 表示任意多项式有界辅助信息（一个可能不可计算的函数 $1^n \mapsto a_n$），而 $*$ 表示不设测试器。Ko 和 Sch¨oning [KS85], K¨obler 和 Thierauf [KT94], 以及 Gavald`a [Gav95] 界定了计算正确辅助信息的复杂度，也就是生成器的复杂度；K¨amper 的非一致证明系统 [K¨am91] 则界定了正确辅助信息集合的复杂度，也就是测试器接受的集合；对于通常的类 $\mathcal{B}$ 和 $\mathcal{C}$，OGTD[poly $\mathcal{B},\mathcal{C}$] 就是他的类 $\mathcal{C}$-$\mathcal{B}$-PSL。K¨obler 和 Thierauf 使用记号 $\mathcal{C}/\mathcal{F}$，要求辅助信息字符串由复杂度类 $\mathcal{F}$中的函数 $1^n \mapsto a_n$ 计算。在他们的记号中，对于语言 A，OGTD[A] $=$ P/FP^A。事实上，OGTD[NP] $=$ P/OptP（见推论 3.2，该结论也可由 Krentel 定理 [Kre88，定理 3.1]推出）；这里的 OptP 是 Krentel 的优化函数类。

Aaronson [Aar07]提出了语言类 YP，即带有不可信建议且可在多项式时间内判定的语言类；Aaronson 和 Drucker [AD14] 提出了它的变体 YP*，其中建议可在多项式时间内进行测试，且无需访问输入；YP* 是 OGTD[poly、P、P],，也就是说，是 K¨amper 的类 P-P-PSL [K¨am91]。输入无关证明系统 [FSW09、GM15]可视为将 P/poly 限制为可验证建议；在 Goldreich 和 Meir [GM15], 的类 OIP 中，交互在 1^n上进行，其结果随后会被用于实际输入。Buhrman、Fortnow、Hitchcock 和 Loff [BFHL13, 定理 4] 证明，如果 NP $\subseteq$ P/poly 且 PH $\subseteq$ P$^{\mathbf{NP}}$，那么对于某个其图属于 co-NP 的辅助信息函数 α，有 PH $\subseteq$ P/α；也就是说，存在由 co-NP 测试器认证的唯一建议。 [HV26b]的主要类是 OM $=$OGTD[FP$^{\mathbf{sOprMA}}$、co-ONP、P],，其中 sOprMA 是 prOMA 的受限版本（仍包含 ONP）。他们还证明，Karp–Lipton 坍缩一直成立到其唯一建议子类 OUM（即 NP $\subseteq$ P/poly 蕴含 PH $=$ OUM）；据目前所知，该子类包含于三个两两不可比较的 Karp–Lipton 坍缩类中，即 OUM $\subseteq$ OM $\subseteq$ OMIN $\cap$P$^{\mathbf{sOprMA}}$ $\cap$ ZPP$^{\mathbf{ONP}}$。

1.1 我们的结果

输入无关的无歧义层次。已知**OUP** $\subseteq$**OMIN**，并且**OMIN** 在多项式时间图灵归约下封闭 [HVW26]；因此，我们可以问**OUH** $\subseteq$ **OMIN** 是否成立，以及**OUH**$_\bullet$的上界是什么。我们对前一个问题给出了肯定回答；更进一步，我们证明这两个层次中的语言都具有多项式规模电路，且这些电路可以通过带有 1^n 预言机的**NP**构造。对于**OUH**，对一个 TALLYNP 语言进行非自适应查询就足够了：

定理（定理4.1）。**OUH** $\subseteq$ **P**$^{\mathbf{TALLY}\cap\mathbf{NP}}$。

特别地，**OUH**中的每种语言都可以在多项式时间内通过对某个**ONP**中的语言进行非自适应查询来判定，即**OUH** $\subseteq$**P**$_{\parallel}^{\mathbf{ONP}}$（推论4.2）。对于对承诺预言机的宽松访问，我们构造了一个自适应生成器：

定理 （定理 5.1）。**OUH**$_\bullet$ $\subseteq$**OGTD**[**NP**]。

对于带有 **NP** 预言机的生成器，可以将辅助信息规范化，从而得到一个由 **co-ONP**测试器认证的唯一建议字符串：

引理 （引理3.1 和推论 3.2）。**P**/**OptP** $=$ **OGTD**[**NP**] $=$ **OGTD**[**ONP**] $=$**OUGTD**[**FP**$^{\mathbf{ONP}}$, **co-ONP**，**P**]。

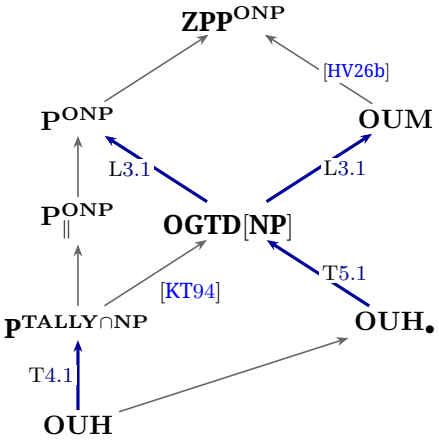


图1： 本文讨论的主要类之间的包含关系；箭头指向更大的类。粗箭头所示的包含关系由本文证明；其他箭头所示的关系是已知的或显然的。所有这些类都包含于 $\mathbf{P}$/**多项式**。此外，根据引理 $\mathbf{OGTD}[\mathbf{NP}] = \mathbf{OUGTD}[\mathbf{FP}^{\mathbf{ONP}}、\mathbf{co\text{-}ONP}、\mathbf{P}] = \mathbf{P}/\mathbf{OptP}$ *3.1* 以及推论 *3.2*。

它将该类（以及 $\mathbf{OUH}_\bullet$）同时归入 $\mathbf{P}^{\mathbf{ONP}}$ 和$\mathbf{OUM}$。因此，$\mathbf{OUH}_\bullet$ 中的每种语言都可以通过对$\mathbf{ONP}$中的某种语言进行自适应查询，在多项式时间内判定： $\mathbf{OUH}_\bullet \subseteq \mathbf{P}^{\mathbf{ONP}}$ （推论 5.3）。

参见图 1中的包含关系图。

点算子与单次查询。我们可以尝试用点算子而非预言机来定义这些层次。点算子由 Wagner [Wag86] 引入，并在 [ZF87、Zac88、Sch89、BGM06] 中得到进一步研究。多项式层次可以用交替存在与全称点算子来定义（例如，$\mathrm{NP}^{\mathbf{NP}} = \exists \cdot$co-NP），OH 也可以如此定义，因为 $\mathrm{ONP}^{\mathbf{ONP}} = \exists_{\mathbf{o}} \cdot$co-ONP 的证明 [HVW26，定理 4.4] 显然可以相对化；AUPH层次也可以等价地用交替唯一点算子来定义 [LR94、NR98]。相比之下，由交替的输入无关无歧义点算子定义的层次会塌缩为 OUP ∪co-OUP（定理 6.2 和推论 6.3）。我们还注意到，对 prOUP 中的一个承诺问题进行一次宽松查询，可以替换为对 OUP 中某个语言的一次查询，即 $\mathrm{OUP}^{\mathbf{prOUP}[1]} = \mathrm{OUP}^{\mathbf{OUP}[1]}$；$\mathrm{UP}^{\mathbf{prUP}[1]} = \mathrm{UP}^{\mathbf{UP}[1]}$ 也有类似结论（定理 6.5）。这与 $\mathrm{RP}^{\mathbf{prRP}[1]} =$ BPP 的情形形成对照（依据 [BF99，定理 1] 的证明），而 P =RP ≠ BPP 相对于某个预言机 [BF99，定理 3]，；$\mathrm{ONP}^{\mathbf{prONP}[1]} = \mathrm{O}_2^{\mathrm{P}}$ [HVW26，定理 4.7]，；但 $\mathrm{ONP}^{\mathbf{ONP}[1]} = \mathrm{ONP}^{\mathbf{ONP}}$ 是否等于 $\mathrm{O}_2^{\mathrm{P}}$ 尚不明确。最后， [HVW26] 中的同见证类 SOUP 等于 OUP∩co-OUP；因此 $\mathrm{OUP}^{\mathbf{SOUP}} =$OUP 且 $\mathrm{SOUP}^{\mathbf{SOUP}} =$ SOUP（命题 6.6）。

1.2 我们的方法

规范建议。为了更清楚地定位我们将层次结构置于哪个类中，$\mathbf{OGTD}[\mathbf{NP}]$, 我们在第3节讨论这一问题。我们的引理3.1 基于 Krentel 的观察 [Kre88，定理3：1]NP**预言机的正确答案序列，是所有肯定回答均有证书的序列中字典序最大的序列；稍加推导，也可以从他的结果中得出该引理。Buhrman、Fortnow、Hitchcock 和 Loff** [BFHL13，定理 4] 的证明以类似方式获得了一个具有 co-NP**图的唯一建议**：将肯定回答对应的字典序最小见证附加到$\mathbf{FP}^{\mathbf{NP}}$ 生成器的运行记录中。我们将生成器的一次运行编码为

在 1^n 中，编码由预言机答案序列以及肯定回答的证书组成，并将字典序最大的可行编码 a_n 作为辅助信息。这样， a_n 它就是找到它的前缀搜索中所有查询的公共见证，因此可在$\mathbf{FP}^{\mathbf{ONP}}$中找到；它也能反驳其他所有可行编码，因此它是**co-ONP**测试器唯一接受的字符串。类似地，在 [HVW26、HV26b]中，线性序的最小元被用作公共反驳。

语言预言机。对于 OUH（定理 4.1），我们证明 $\mathrm{P}^{\mathbf{TALLY}\cap\mathbf{NP}}$ 在 OUP 预言机计算下封闭。我们使用 Mahaney 的普查技术（见 [HO02，第 1.2] 节），该技术也曾用于从 1^n 计算辅助信息（见 [KT94] ），以及处理带稀疏预言机的无歧义计算（见 [HR97]）：只要知道一元 NP 预言机在相关长度以内的字符串数量，NP 机器就能验证其所有回答。借助这一方法，K¨obler 和 Thierauf [在2] KT94，定理3.2] 中证明了 $\mathrm{NP}^{\mathbf{SPARSE}\cap\mathbf{NP}} \subseteq \mathrm{NP/OptP}[O(\log n)]$, ，其中辅助信息就是字符串的数量。对于 $\mathrm{NP}^{\mathbf{TALLY}\cap\mathbf{NP}}$中的计数语言，可以向计数预言机本身查询这个数量，而剩下的问题则是对另一个 NP 中计数语言的一次查询；因此，每个 $\mathrm{NP}^{\mathbf{TALLY}\cap\mathbf{NP}}$ 中的计数语言都属于 $\mathrm{P}^{\mathbf{TALLY}\cap\mathbf{NP}}$ （引理 2.8；稍加推导，也可从 [KT94]的结果中得出）。我们的新要点是外层 OUP 计算的见证具有无关性：由于长度为 n 的所有肯定输入都共享同一个唯一见证 w_n，其各位就是与输入无关的预言机 NP 问题的答案，而这些问题可以由一个 $\mathrm{NP}^{\mathbf{TALLY}\cap\mathbf{NP}}$中的计数语言回答。

宽松访问。 对于承诺预言机，普查技术并不起作用，因为无法明确归属承诺外查询，而且搜索接受见证时可能会找到一个在承诺之外的输入上被接受的见证。我们为**OUH•**（定理5.1）构造的生成器遵循 Hartmanis 和 Hemachandra [HH90 的证明策略，定理 2.1] 指出，所谓“稳健地单值”的机器（对于每个预言机至多有一条接受路径的非确定性机器）相对于每个预言机 A 都接受可在多项式时间内、通过查询 A 和一个**NP** 预言机判定的语言；该结果在 [ST07 定理 5.1]中被扩展到“稳健地无歧义”的交替机（交替次数有界的机器，其中对于每个预言机，每个存在配置至多有一个接受后继，每个全称配置至多有一个拒绝后继）。它们的模拟过程反复使用一个**NP**预言机，寻找一条与迄今为止 A获知的答案一致的接受路径，并查询A该路径上的所有查询；当路径正确或不存在这样的路径时停止。由于不同预言机对应的两条接受路径必定在某个查询上发生冲突，因此每条新路径都会在某个查询上与每条先前路径发生冲突，而先前路径在该查询上的答案是错误的；因此，轮次数被一条路径上的查询数量所界定。Spakowski 和 Tripathi [ST07] 逐层应用这一论证：通过对层级数量进行 k 归纳法，他们在多项式时间内模拟较低层级，并查询 A 和一个 $\mathbf{\Sigma^P_{k-1}}$ 预言机，然后将 [HH90]的论证应用于顶层，从而得到使用一个 $\mathbf{\Sigma^P_k}$ 预言机的相同结论。

我们的设定有两点不同。首先，我们的机器并不鲁棒：只有当查询属于承诺集，且所用预言机与下一层的承诺问题一致时，它们才是无歧义的。其次，我们的生成器无法访问承诺预言机（它只能访问一个**NP**预言机），因此无法利用承诺查询的答案来检查路径。因此，我们用每个层级、每种查询长度对应的候选无知见证集合来替代学得的答案。相对于这些集合，预言机的两种回答都有经认证的依据：肯定回答由接受见证认证，否定回答则由所有候选见证均被拒绝来认证。生成器会寻找一个输入，使得两次经认证的运行产生不同的输出。根据宽松访问，这种不一致意味着对承诺集中的某个查询给出了错误回答；对于这类查询，见证的唯一性和无关性取代了接受路径之间的冲突，使我们能够沿预言机塔逐层追溯，直至发现缺失的正确见证，而这

得以补上。由于见证是无知的，轮次数量受（层级，查询长度）对的数量限制，而集合的大小则可通过按层级递降的归纳法加以限制。因此，一个**NP** 预言机便能同时服务所有层级，无须像 [ST07], 那样逐层归纳；并且，对于给定长度的所有输入，生成器会生成一个建议字符串，而不是判定某个给定输入。

点算子与单次查询。 推论中的坍缩6.3 运用了这一思路，并在 [HVW26] 中加以应用，**以证明 SOUP** 在多项式时间图灵归约下封闭：在固定输入上检验候选无知见证。由于外层见证是唯一的，内层**co-OUP** 谓词在两个固定对中的一个上为假，因此无需知道实际输入，就能确定反驳该谓词的无知见证，从而得到 $\exists_{\mathbf{ou}}\cdot\mathbf{co\text{-}OUP} = \mathbf{OUP}\cup\mathbf{co\text{-}OUP}$ （定理6.2）。相比之下，基于非无知点算子的无歧义层次 **AUPH** 相对于预言机 [ST07]是无限的。

为证明对承诺问题的单次查询可以转化为语言查询，我们将“输入标记”论证（在输入 q 上提出的查询 x 会被替换为一对（x, q），以便预言机能够检查是否确实在 q 上提出了 x）与 Cai、Hemachandra 和 Vysko˘c 的观察结合起来 [CHV93，命题 2。6], 他们证明，对于确定性机器，受保护的非自适应访问**prUP** 并不比访问**UP**更强；同时注意到，对于无歧义计算，只要单个宽松查询的答案会影响结果，该查询就必然受到保护：否则，以两种方式回答该查询，就会违反正确性，或者使计算结果依赖于预言机的回答。

论文结构。第2节包含定义、记号、约定和已知事实。第3 节介绍关于带有**NP** 预言机的生成器的准备工作。接下来的几节给出论文的主要结果：第4 节和第 5节分别讨论**OUH** 和 $\mathbf{OUH}_\bullet$，第6节则考察层次的前几层（由点算子、单次查询以及$\mathbf{OUP}\cap\mathbf{co\text{-}OUP}$类定义的层次）。

2 预备知识

F关于复杂性理论标准概念的严格定义，请参阅 [AB09]。

2.1 约定、单射配对、承诺问题与预言机访问 s

除非另有说明，所有机器均在多项式时间内运行； n表示输入的长度。在基于多项式时间谓词的定义中，对于长度为 n 的输入，所有见证、随机串和建议字符串的长度都由关于 n的多项式给定，量词的取值范围为这些长度的字符串。对和元组采用长度规则编码：若各分量的长度由 n确定，则编码的长度可以确定 n及各分量的长度。若语言 T 满足 $T\subseteq\{1\}^*$，则称其为一元语言；**TALLY** 是所有一元语言构成的类。我们固定一个常数个自然数的单射配对 π ，使得 π 关于其参数是多项式有界的，并且 π 及其逆映射均可在关于一元长度的多项式时间内计算；我们用它将数的元组编码为一元串。用$<_{\ell ex}$ 表示等长字符串上的字典序。

承诺问题（由 Even、Selman 和 Yacobi 提出；参见 [AB09，第 23.4] 节）是由两个不相交的字符串集合组成的有序对 $\Pi=(\Pi_{\mathrm{YES}},\Pi_{\mathrm{NO}})$；语言 L 可表示为 (L,L)。我们将$\Pi(x)=1$记作 $x\in\Pi_{\mathrm{YES}}$ ，将 $\Pi(x)=0$记作 $x\in\Pi_{\mathrm{NO}}$。若语言 O 满足 $\Pi_{\mathrm{YES}}\subseteq O$且 $O\cap\Pi_{\mathrm{NO}}=\emptyset$，则称其与 Π 一致。对于由机器定义的语言类 $\mathcal{C}$ ，pr$\mathcal{C}$ 是相应的承诺问题类：正确性要求以及语义要求

（例如见证的唯一性）仅针对承诺集 $\Pi_{\text{YES}} \cup \Pi_{\text{NO}}$ [HS15]施加。对于承诺问题类 $\mathcal{D}$，其中的语言类 $\mathcal{D}$ 由满足以下条件的语言 L 组成：$(L,\ L) \in \mathcal{D}$。

我们针对承诺问题默认采用的预言机访问模式是宽松访问：允许对承诺集之外的查询进行提问，并可任意回答；对于所有此类回答，计算必须满足其所属类别的全部要求。这些回答不必一致：同一查询在不同计算路径上可能得到不同的回答。我们假设一条计算路径不会重复查询。于是，对于确定性预言机图灵机，宽松访问等价于以下要求：对于所有与预言机一致的语言，计算都必须正确 [CHV93，命题 5。9]对于无歧义预言机图灵机，宽松访问等价于 [HS15的定义 4]：对于每个一致语言，计算都必须正确；并且其接受见证（如果存在）必须对所有一致语言都相同。（事实上，单条计算路径上收到的回答由某个一致语言给出；如果两个一致语言产生了不同的接受见证，那么将它们在相应路径上的回答结合起来，就会产生两条接受路径。）下文将采用这一表述。

因此，$\mathbf{P}^{\Pi}$ 是由确定性多项式时间预言机图灵机 M 判定的语言类，其中 M^O 对每个与 Π 一致的语言 O 都给出正确答案；$\mathbf{FP}^{\Pi}$ 计算可能有多个合法输出（对应不同的一致语言 O），这些输出都必须正确，但不必是同一个字符串。下标 $\|$（如$\mathbf{P}^{\Pi}_{\|}$或$\mathbf{FP}^{\Pi}_{\|}$中的下标）表示所有查询都在收到任何答案之前计算完成。对于由承诺问题组成的类 $\mathcal{D}$，$\mathbf{P}^{\mathcal{D}} = \bigcup_{\Pi\in\mathcal{D}} \mathbf{P}^{\Pi}$，其他预言机复杂性类也类似；$\mathcal{C}^{\mathcal{D}[1]}$ 表示每条计算路径上最多进行一次查询。

2.2 无歧义且输入无关的类

We 在其一般承诺相对化版本中定义我们感兴趣的主要复杂性类 .

定义 2.1 (**UP**，**OUP** [HVW26], **ONP** [FSW09])。设 Π 是一个承诺问题。承诺问题 Γ 属于$\mathbf{prUP}^{\Pi}$，当且仅当存在一台确定性多项式时间预言机机器 V，使得对于每个 $x \in \Gamma_{\text{YES}} \cup \Gamma_{\text{NO}}$，集合

$$W(x) = \{w : V^O(x, w) = 1\}$$

对于所有与 Π 一致的语言 O，其定义相同：$|W(x)| = 1$ 对于 $x \in \Gamma_{\text{YES}}$，以及 $W(x) = \emptyset$ 对于 $x \in \Gamma_{\text{NO}}$。此外，若存在字符串 w_n，使得对于每个 $x \in \Gamma_{\text{YES}}$，$W(x) = \{w_{|x|}\}$，那么 $\Gamma \in \mathit{prOUP}^{\Pi}$ 和 w_n 称为无视无歧义见证。若存在一个确定性多项式时间预言机机器 V 和字符串 w_n，使得对于每个与 Π 一致的语言 O，$V^O(x, w_{|x|}) = 1$ 对于每个 $x \in \Gamma_{\text{YES}}$，且 $V^O(x, w) = 0$ 对于每个 $x \in \Gamma_{\text{NO}}$ 和每个 w，则问题 Γ 属于 prONP^{Π}。类 UP^{Π}、OUP^{Π} 和 ONP^{Π} 由这些类中的语言组成；对于语言预言机 Π（特别是没有预言机时），我们得到通常的定义。

2.3 输入无关的无歧义层次

定义 2.2（输入无关的无歧义层次）。定义 $\mathbf{OU\Sigma}^{\mathrm{p}}_0 = \mathbf{P}$、$\mathbf{prOU\Sigma}_{\bullet 0} = \mathbf{prP}$，以及对于 $k \geq 0$，定义 $\mathbf{OU\Sigma}^{\mathrm{p}}_{\mathrm{k}+1} = \mathbf{OUP}^{\mathbf{OU\Sigma}^{\mathrm{p}}_{\mathrm{k}}}$ 和 $\mathbf{prOU\Sigma}_{\bullet k+1} = \mathbf{prOUP}^{\mathbf{prOU\Sigma}_{\bullet k}}$（回顾一下，对承诺预言机的访问是宽松的）。令 $\mathbf{OUH} = \bigcup_k \mathbf{OU\Sigma}^{\mathrm{p}}_{\mathrm{k}}$ 以及 $\mathbf{prOUH}_{\bullet} = \bigcup_k \mathbf{prOU\Sigma}_{\bullet k}$。类 $\mathbf{OU\Sigma}_{\bullet k}$ 和 $\mathbf{OUH}_{\bullet}$ 由相应承诺问题类中的语言组成。

注意，$\mathbf{P}^{\mathbf{prOUH}_\bullet} = \mathbf{OUH}_\bullet$：一台以宽松方式访问 $\Pi \in \mathbf{prOU\Sigma}_{\bullet k}$ 的确定性多项式时间机，是一个见证为空的$\mathbf{OUP}^\Pi$ 验证器，因此其语言属于$\mathbf{OU\Sigma}_{\bullet k+1}$；反向包含需要一次查询。

2.4 点算子

定义 2.3 ([HVW26，定义 2.8])。设 $\mathcal{C}$ 是一个承诺问题类。若语言L 属于 $\exists_{\mathbf{ou}}\cdot\mathcal{C}$，则存在 $\Pi \in \mathcal{C}$、一个多项式 p以及字符串 $w_n \in \{0,1\}^{p(n)}$，使得对于每个 n 以及每个 $x \in \{0,1\}^n$，

$$\begin{aligned} x \in L &\Rightarrow (x, w_n) \in \Pi_{\text{YES}} \text{ and } (x, w) \in \Pi_{\text{NO}} \text{ for all } w \neq w_n, \\ x \notin L &\Rightarrow (x, w) \in \Pi_{\text{NO}} \text{ for all } w. \end{aligned}$$

如果见证 w_n 可以依赖于 x，我们得到唯一存在算子 $\exists_{\mathbf{u}}\cdot$（在 *[LR94、NR98、ST09]* 中记作 $\exists!$）。相反，如果去掉关于字符串 $w \neq w_n$的条件，我们得到无视存在算子 $\exists_{\mathbf{o}}\cdot$ *[HVW26]*,；此外，如果见证还可以依赖于 x，则得到存在算子 $\exists\cdot$。对偶算子是 $\forall_{\mathbf{ou}}\cdot\mathcal{C} = co\text{-}\exists_{\mathbf{ou}}\cdot co\text{-}\mathcal{C}$，其中 *co-* 交换$\Pi_{\text{YES}}$ 和 Π_{NO}。我们允许见证长度恒等为零（将 (x, ε) 与 x 视为相同）；因此，$\mathcal{C}$ 中的语言属于 $\exists_{\mathbf{ou}}\cdot\mathcal{C}$。

对于 $\exists_{\mathbf{ou}}\cdot$ 和 $\exists_{\mathbf{u}}\cdot$，所有满足 $|w| = p(|x|)$ 的对 (x, w) 都属于 Π 的承诺集。注意，$\mathrm{OUP} = \exists_{\mathbf{ou}}\cdot\mathrm{P}$，并且对于每个预言机 A，$\mathrm{OUP}^A = \exists_{\mathbf{ou}}\cdot\mathrm{P}^A$（参见 [HVW26，推论 4.2]）。对于对承诺问题 Π 的宽松访问，同样有：$\mathrm{OUP}^\Pi = \exists_{\mathbf{ou}}\cdot\mathrm{P}^\Pi$。事实上，根据定义 2.1，$\mathrm{OUP}^\Pi$ 验证器在输入 x 上接受的见证集合，对于所有与 Π 一致的语言都是相同的，因此被接受的对 (x, w) 构成一个属于 P^Π 的语言；反之则显然成立。

2.5 生成器–测试器–解码器计算

定义 2.4 (**OGTD**，**OUGTD** [HV26b])。设 $\mathcal{X}$ 是一个承诺问题类。属于 FP 的生成器 $^{\mathcal{X}}$ 是一个确定性的多项式时间预言机转换器 G，其预言机为 $\Pi \in \mathcal{X}$，在输入 1^n上输出长度为 p的字符串n，其中 p是一个多项式。其 合法输出 集合为

$$\mathrm{Out}_G(n) = \{G^O(1^n) : O \text{ is a language consistent with } \Pi\}.$$

$\mathcal{B}$和 $\mathcal{C}$是语言类。若存在一个生成器 $G \in \mathbf{FP}^{\mathcal{X}}$、一个测试器 $T \in \mathcal{B}$（一个由对 $(1^n, a)$构成的语言）以及一个解码器 $D \in \mathcal{C}$（一个由对 (x, a)构成的语言），使得对于每个 n，语言 L属于 $\mathbf{OGTD}[\mathbf{FP}^{\mathcal{X}}, \mathcal{B}, \mathcal{C}]$，

$$\mathrm{Out}_G(n) \subseteq \mathrm{Acc}_T(n) \subseteq \mathrm{Good}_D(n),$$

其中 $Acc_T(n) = \{a : (1^n, a) \in T\}$，而 $Good_D(n) = \{a \in \{0,1\}^{p(n)} : \forall x \in \{0,1\}^n\ D(x, a) = L(x)\}$。若此外对于每个 n都有 $|Acc_T(n)| = 1$，则语言 L属于 $OUGTD[FP^{\mathcal{X}}, \mathcal{B}, \mathcal{C}]$。用 $*$ 代替 $\mathcal{B}$ 表示不需要测试器，即只需 $Out_G(n) \subseteq Good_D(n)$。最后，$OGTD[\mathcal{X}] = OGTD[FP^{\mathcal{X}}, *, P]$。

命题 2.5 ([HV26b，命题 2.11])。设 Π 是一个承诺问题。那么 $\mathbf{OGTD}[\Pi] \subseteq \mathbf{P}^\Pi \cap \mathbf{P/poly}$。

我们也使用 [KT94] 的记号来表示具有有界建议函数的类：对于一个类$\mathcal{C}$（语言类）和一个类 $\mathcal{F}$（建议函数类），$\mathcal{C}/\mathcal{F}$ 是语言类 L，使得$x \in L \Leftrightarrow (x, h(1^{|x|})) \in B$对某个 $B \in \mathcal{C}$和 $h \in \mathcal{F}$成立。

定义 *2.6*（*sOprMA*，*OUM* [*HV26b*]）。承诺问题 Π 属于 *sOprMA*，如果存在一个多项式时间谓词 $V(x,y,r)$和字符串 y_n，使得对每个 $x \in \{0,1\}^n$：若 $x \in \Pi_{\text{YES}}$，则对所有 r有 $V(x,y_n,r) = 1$；若 $x \in \Pi_{\text{NO}}$，则对所有 y有 $Pr_r[V(x,y,r) = 1] < 1/4$；并且，对所有 y和 r，$V(x,y,r) = 1$蕴含 $V(x,y_n,r) = 1$。定义 $OUM = OUGTD[FP^{\mathbf{sOprMA}}$、*co-ONP*、$P]$。

每个 **ONP** 中的语言都属于 **sOprMA**：取 $V(x,y,r) = V'(x,y)$，其中 **ONP**验证器 V' [HV26b]。因此 $\mathbf{FP}^{\mathbf{ONP}}$ 生成器也是 $\mathbf{FP}^{\mathbf{sOprMA}}$ 生成器。

2.6 关于计数 NP 预言机计算的有用事实

对一元预言机的查询可以并行化，因为所有查询都可以在准备阶段计算出来，因为所有可能的一元查询的（多项式）范围都是已知的（例如，参见 [KT94]）。由于计数 **NP** 语言属于 **ONP**（参见 [FSW09，GM15]），可以表述如下有用事实。

命题 2.7. $\mathbf{P}^{\mathbf{TALLY}\cap\mathbf{NP}} = \mathbf{P}_{\|}^{\mathbf{TALLY}\cap\mathbf{NP}} \subseteq \mathbf{P}_{\|}^{\mathbf{ONP}}$。

引理 2.8。对于集合 $C \in \mathbf{TALLY} \cap \mathbf{NP}$，任意属于 $\mathbf{NP}^C$ 的一元集合都属于 $\mathbf{P}^{\mathbf{TALLY}\cap\mathbf{NP}}$。

证明。设 $U \in \mathbf{NP}^C$ 是一个计数语言，由带有预言机 C 的非确定性多项式时间预言机机器 N 接受。如果 U 属于 **NP**，我们的确定性机器只需查询 U。然而，机器 N 使用预言机 C。我们将通过把该预言机替换为 **NP** 机器的一个额外输入来消除它（这样我们的确定性机器将查询另一个计数语言 D）。利用预言机 C，我们可以确定性地计算 C 的真值表中相关部分。然而，我们不能将这张真值表作为额外输入，因为它太大，无法嵌入计数语言中。相反，这个额外输入将是一个短得多的“校验和”：C 中相关长度的字符串数量（我们同样可以利用预言机 C 计算该数量）。

设 $h(m)$ 是 N 在输入 1^m 上的运行时间界；它将界定 N 在该输入上的任何查询的长度。令 C_m 包含 C 中长度至多为 $h(m)$ 的所有字符串，并令 $c_m = |C_m|$。知道 c_m 后，一台 **NP** 机器可以猜测 C_m（即，c_m 个数 $k \leq h(m)$）以及 C_m 中各元素的 **NP** 见证；随后，它可以确定性地验证这些见证，以确保猜出的字符串确实属于 C。此后，该机器便可以自行回答对 C 的查询。

形式化地，考虑计数语言 D，它由所有字符串 $1^{\pi(m,c)}$ 组成（回忆一下，π 表示单射配对）；对于这些字符串，存在一个大小为 c 的数字集合 S（$\leq h(m)$），其中每个 $k \in S$ 都有 $1^k \in C$ 的正确 NP 见证，并且存在一个 N 的接受计算，其输入为 1^m，其中对 $k \in S$ 的查询 1^k 均得到肯定回答，而所有其他查询均得到否定回答。显然，$D \in$ NP。对于 $c = c_m$，唯一这样的集合 S 是 $\{k \in \mathbb{N} : 1^k \in C_m\}$：$S$ 中的所有元素都必须有正确见证，而校验和确保列出了 C_m 的所有元素。因此，给 N 的回答与 C 的回答相同，所以 $1^{\pi(m,c_m)} \in D$ 当且仅当 $1^m \in U$。因此，要判定 $1^m \in U$ 是否成立，确定性多项式时间机针对所有 $k \leq h(m)$ 向预言机 C 查询 1^k，从而得到 c_m，然后再向预言机 D 查询 $1^{\pi(m,c_m)}$。计数预言机 C 和 D 可利用 π 合并为一个属于 TALLY $\cap$ NP 的预言机。 □

3 带有 NP 预言机的生成器

我们特别关注**OGTD**[**NP**]，，正如我们下面所证明的（推论3.2），它是类**P**/**OptP**，由 K¨obler 和 Thierauf [KT94]很早以前研究过。但首先，我们证明在这个类中

生成器计算出的辅助信息可以变得“规范的”（即无歧义），并且可以在**co-ONP**中进行测试。

我们遵循 Krentel 的证明 [Kre88，定理 3。1]：一个 **NP**预言机的正确回答序列，是所有肯定回答均有证书的字典序最大序列。我们需要将肯定回答的见证加入序列，使候选序列能够在多项式时间内验证；并利用这一点，在输入 1^n 上，使最大的可行字符串 a_n 仅取决于 n，从而使其成为生成器所有查询的公共见证，以及所有其他字符串的公共反驳；由此得到 **ONP** 预言机和 **co-ONP** 测试器。

引理 3.1. OGTD[NP] = OGTD[ONP] = OUGTD[$\mathbf{FP}^{\mathbf{ONP}}$, co-ONP, P].

证明。包含关系 **OUGTD**[$\mathbf{FP}^{\mathbf{ONP}}$, **co-ONP**, **P**] $\subseteq$ **OGTD**[**ONP**] $\subseteq$ **OGTD**[**NP**]是显然的。我们将生成器的 **NP** 预言机替换为 **ONP** 预言机，并通过使用额外的**co-ONP**测试器使建议字符串唯一。

设 $L \in \mathbf{OGTD}[A]$ 配有一个生成器 $G \in \mathbf{FP}^A$ 和一个解码器 $D \in \mathbf{P}$，其中 $A \in \mathbf{NP}$ 有一个多项式时间验证器 U。我们的新生成器将输出一段能够重构旧辅助信息的内容：$G(1^n)$ 的查询序列 a 对应的预言机回答，后面接上肯定回答对应的见证列表（用零填充至足以涵盖所有可能情况的固定长度）。通过模拟 G 的运行并使用 U 验证见证，即可在多项式时间内验证这段内容。唯一无法验证的是否定回答。因此，我们将要求这段字符串 a_n在字典序上最大：若找到它，那么所有回答必然都是正确的；此外，最大字符串保证唯一。可以在 **co-ONP** 中测试一个形式上正确的候选字符串是否最大：正确答案 a_n 就是无知见证。

设 $R(1^n, a)$ 是一个多项式时间确定性过程，用于检查 a 是否与 $G(1^n)$ 使用来自 a 的回答进行的运行一致，并检查所有见证是否正确。我们将 **co-ONP** 测试器定义为

$$T = \{(1^n, a) : R(1^n, a) \wedge \forall b >_{lex} a \ \neg R(1^n, b)\},$$

因此它仅接受 $a = a_n$。我们的新生成器将使用预言机 $Q = \{(1^n,\ c) :\ \exists a \geq_{lex} c\ R(1^n,\ a)\} \in \mathbf{ONP}$（其中 a_n 再次扮演无知见证的角色）。它将使用该预言机通过前缀搜索找到 a_n。也就是说，在找到 a_n的前 $j-1$ 位 $b_1 \dots b_{j-1}$后，它会询问字符串 $c = b_1 \dots b_{j-1}10 \dots 0$（其长度与 a_n相同）是否满足 $(1^n,\ c) \in Q$。由于 a_n 是满足 $R(1^n,\ a)$ 的最大字符串 a，因此 $(1^n,\ c) \in Q$ 当且仅当 $a_n \geq_{lex} c$，也就是说，当且仅当 a_n 的第 j位是 1。

新的解码器将从 a_n 重建旧的辅助信息，并运行旧的解码器 . □

推论 3.2. OGTD[NP] =P/OptP。

证明。 这是证明引理3.1而非其表述的推论。我们生成a_n ，使其等于所有字符串中的最大值 a ，其中 $R(1^n, a)$，由此得到一个 **OptP** 函数，计算a_n ，输入为 1^n。反过来，每个 **OptP** 中的函数都可以通过二分搜索$\mathbf{FP}^{\mathbf{NP}}$ 来计算 [Kre88]。

□

4 无关的无歧义层次

我们证明，基于语言的无关无歧义层次包含于无关层次的第二层；此外，其中的语言可通过对 TALLY **NP** 预言机进行非自适应查询来计算。

定理 4.1. OUH $\subseteq$ $\mathbf{P}^{\mathbf{TALLY} \cap \mathbf{NP}}$。

proof。只需证明 $\mathbf{OUP}^{\mathbf{P}^{\mathbf{TALLY}\cap\mathbf{NP}}} \subseteq \mathbf{P}^{\mathbf{TALLY}\cap\mathbf{NP}}$，并反复使用它.

设 $L \in \mathbf{OUP}^A$ 满足 $A \in \mathbf{P}^T$，其中 $T \in \mathbf{TALLY} \cap \mathbf{NP}$，并设 V是针对 L 的多项式时间验证器，具有长度为 $p(n)$ 的无视无歧义见证 w_n。设 V' 是 V 的一个版本，它使用自身的$\mathbf{P}^T$ 算法来模拟预言机 A（因此 V' 将 T 用作预言机）。

由于 w_n 是唯一的，我们可以通过查询以下计数语言，逐位计算它（且不依赖于输入）

$$W = \{1^{\pi(n,i)} : 1 \le i \le p(n) \text{ and } V'^T(x,w) = 1 \text{ for some } x \in \{0,1\}^n \text{ and } w \text{ with } w_i = 1\}.$$

如果 $L \cap \{0,1\}^n \neq \emptyset$，那么 $w_n = W(1^{\pi(n,1)}) \dots W(1^{\pi(n,p(n))})$。如果 $L \cap \{0,1\}^n = \emptyset$，那么这个字符串就是 $0^{p(n)}$，在这种情况下，它与其他任何字符串同样合适，因此不失一般性，我们可以假设 $w_n = 0^{p(n)}$。从 W 的定义很容易看出，$W \in \mathbf{NP}^T$。根据引理 2.8，W 可以在某个TALLY $\cap$NP预言机的帮助下，在确定性多项式时间内计算。

现在，L 可以先计算 w_n，然后利用这个见证在实际输入上模拟 V'^T 来判定。由此得到的确定性算法使用计数**NP** 预言机 T 和 O，根据 π，可以将它们合并为一个预言机。

□

推论 4.2。$\mathbf{OUH} \subseteq \mathbf{P}_{\|}^{\mathbf{ONP}}$。

证明。由定理4.1 和命题 2.7。 □

5 松弛的输入无关无歧义层次

定理 4.1 的证明利用了预言机是一个语言这一事实：对于承诺预言机，寻找接受见证时可能会碰到一个见证，它在承诺之外的输入上也被接受。现在，我们将通过转向可能更大的类 OGTD[NP], 来克服这一困难；该类的生成器会进行自适应 NP（实质上是 ONP）查询（若采用非自适应查询，则会得到 $\mathbf{P}^{\mathbf{TALLY}\cap\mathbf{NP}}$：它在 1^n 上的查询q_n,**1**, q_n,2, .. . 不依赖预言机回答，因此其预言机 A ∈NP 可以替换为计数语言 $\{1^{\pi(n,j)} : q_n,_j \in A\}$，而该语言属于 NP），相应地，我们使用 ONP 预言机进行的确定性模拟也会变为自适应的（比较推论 4.2 和 5.3）。

定理 5.1. $\mathbf{OUH}_\bullet$ ⊆OGTD[NP]。

证明。回顾一下，$\mathbf{OUH}_\bullet = \mathbf{P}^{\mathbf{prOUH}_\bullet}$。设 $L \in \mathbf{P}^{\Pi_d}$，其中 $\Pi_d \in \mathbf{prOUP}^{\Pi_{d-1}}$，并设 M 为相应的多项式时间确定性预言机机器。问题 Π_d 通过前面层级上的承诺问题归纳定义，也就是说，存在 $\Pi_0 \in$**prP**，并且对于$1 \le i \le d$，有 $\Pi_i \in \mathbf{prOUP}^{\Pi_{i-1}}$，以及预言机多项式时间确定性机器（验证器）$V_i$，用于 Π_i，其无视无歧义见证为$w_i,\ m$。所有机器都必须能够针对预言机查询在承诺集之外的任意回答正确运行，尤其是针对任何与 Π_i一致的语言 A_i 所给出的回答。下面定义这类特定语言。固定一个与 Π_0一致的多项式时间语言 A_0，并定义

$$A_i = \{q : \exists w\ V_i^{A_{i-1}}(q,w) = 1\} \qquad (1 \le i \le d).$$

对 i进行归纳可知，语言 A_i 与 Π_i一致；特别地，M^{A_d} 必须判定 L。注意，在$\mathbf{OUH}_\bullet$的定义中，我们没有用通过一致语言进行访问来替代宽松访问；这些语言仅用于分析。

我们将用具有相同效果的不同语言 A_i 替换这些语言 B_i。然后，对 B_d 的查询将在**OGTD**[**NP**]中处理。

建议字符串。固定输入长度 n，令 $h = h(n)$为一个多项式，用于界定所有查询和见证的长度，以及在模拟中 M 和所有 V_i 的运行时间

在长度为 n 的输入上，对于所有见证和所有预言机回答， M 的运行时间。我们的 $\mathbf{FP}^{\mathbf{NP}}$生成器将计算的建议信息将包含：在长度为 n 的输入上运行 M 期间，我们在所有可达到的长度上可能使用的全部见证。它将是一个逐步构造的族$S = (S_{i,\ m})_{1\leq i\leq d,\ 0\leq m\leq h}$，由见证集组成，其中 $S_{i,\ m}$ 包含针对长度为 m 的查询上 V_i 的候选见证；对于 $m > h$，我们置入$S_{i,\ m} = \emptyset$。

限制 V_i 仅使用候选见证的一个子集；经认证的运行。预言机机器的一次运行是对其行为的协议，其中包括它收到的针对预言机查询的回答（包括那些不属于预期承诺集的查询）。如果我们能够按以下方式为其所有预言机查询的回答提供证书，则称 $M(x)$ 或某个 $V_i(q, u)$ 的运行是经认证的（相对于特定的 S）。我们为层级-i查询（即对Π_i的查询）的回答定义证书。在第 0 层，如果对 q 的回答恰好等于 $A_0(q)$，则该回答是经认证的（回顾 $A_0 \in$P）。对于$i \geq 1$，对 q 的肯定回答的证书由一个见证 w 、$V_i(q, w)$ 的一次接受运行，以及该运行中所有预言机回答的证书组成；否定回答的证书则对于每个 $s \in S_{i,|q|}$，由 $V_i(q, s)$ 的一次拒绝运行及这些运行中所有预言机回答的证书组成（如果 $S_{i,|q|} = \emptyset$，则空证书足以证明否定回答）。注意，对于常数层级 d，证书的大小关于 n 以及 S中的见证总数是多项式的，并且给定 1^n 和 S后，可以在多项式时间内验证这些证书。

新语言 B_i 和解码器。 如果使用语言 A_i 来给出预言机回答，那么相对于每个 S，它们都是经认证的：对 i进行归纳，对于 $A_i(q) = 1$ 使用一个接受见证，而对于 $A_i(q) = 0$ 使用 $V_i^{A_{i-1}}(q, s)$ 在 $s \in S_{i,|q|}$时的拒绝运行。定义 $B_0 = A_0$ 并且

$$B_i = \{q : \exists s \in S_{i,|q|}\ V_i^{B_{i-1}}(q, s) = 1\} \qquad (1 \leq i \leq d);$$

B_i 的回答也可类似地得到认证。语言 B_i 不必与 Π_i 一致（其定义仅限于 S中的见证），但给定 S后， $B_d(q)$ 可在关于 n和 S中见证总数的多项式时间内计算。因此，我们将解码器定义为 $D(x, S) = M^{B_d}(x)$，并且可以在确定性多项式时间内实现。

生成器。 从 $S = \emptyset$ 开始，并重复以下轮次。

询问 **NP** 预言机：对于某个 $x \in \{0, 1\}^n$， M 是否存在两个相对于 S 具有不同输出的经认证运行。如果不存在，过程结束，生成器输出 S，这已经足够好：由于运行 $M^{A_d}(x)$ 和 $M^{B_d}(x)$ 都是经认证的，对于每个 $x \in \{0, 1\}^n$，我们有 $D(x, S) = M^{B_d}(x) = M^{A_d}(x) = L(x)$。

否则，找到此输入 x 以及这两次运行，并通过反复使用 **NP** 预言机查询它们的前缀，找出它们的所有证书，同时找出问题所在（这应该是缺少一个正确见证——而且正如我们下面所示，它确实会被找到）。具体而言，考察这两次运行并递归地检查它们，检查所有回答的证书。对于每个针对层级-i 查询 q 且带有 $i \geq 1$的否定回答，询问 **NP** 预言机， q 是否存在一个能证明肯定回答的证书（相对于同一个 S）；如果存在，就找出一个，并安排将其见证w 添加到$S_{i,\ |q|}$，然后也对其进行检查。此递归过程结束后，将所有已安排的见证添加到 S，并重复这一轮。

我们需要检查所需的轮次数量。要证明多项式数量的轮次足够，只需证明每一轮都会向某个集合 $S_{i,\ m}$中加入一些正确见证 $w_{i,\ m}$ 即可——由于我们处理的是无歧义无关计算，正确见证最多有 $d(h + 1)$ 个。也可能加入错误见证：对层级-i查询 q 的肯定回答的证书，如果 q 不在 Π_i 的承诺集内（此时 V_i 可能接受多个见证），或者如果证书包含对较低层级的承诺查询的错误回答（如下方命题的第三种情况），则其中可能包含错误见证；只有当 S中缺少某个较低层级的正确见证时，才可能出现后一种情况。这类见证会留在 S中，但它们不会

妨碍我们的计算（我们会验证所有内容），也不会耗费太多时间（每轮在层级 i新增的见证数量受 h 的多项式以及高于层级 i的各层级集合大小的多项式所界定：每个在该轮检查的层级-j 查询，至多会导致检查$h(\max_m |S_{j,\ m}|+1)$个层级-$(j-1)$查询）；因此，根据归纳法，在所有轮次中各集合的大小始终保持为多项式。特别是，我们需要将 S 填充到的长度有多项式上界，且生成器的运行时间也是多项式的。

剩余论断：每轮至少加入一个正确见证。如果某个答案是对相应 Π_i 的承诺集中的某个查询给出的错误答案，则称该答案为错误的。我们声称，每一轮都会向某个 $S_{i,m}$ 加入一个此前不存在的见证 $w_{i,m}$ 。 M的两次运行中，有一次输出错误；因此它包含一个错误答案，否则它就会是某个与 Π_d 一致的语言 O 对应的 $M^O(x)$ 运行。类似地，若 V_i 的一次运行对所有承诺查询的回答都正确，那么它就是某个与 Π_{i-1} 一致的语言 O 对应的 V_i^O 运行。考虑本轮中发现的一个针对层级-i 查询 q 的错误答案（$i \geq 1$，因为在层级 0，所有经证实的答案始终正确）：

- 如果答案为肯定，则 $q \in \Pi_{i,\text{NO}}$，且证书包含 $V_i(q,w)$ 的一次接受运行，其中必定包含一个层级 $i-1$ 上的错误答案。
- 如果答案为否定且 $w_{i,|q|} \in S_{i,|q|}$，那么 $q \in \Pi_{i,\text{YES}}$，且证书包含 $V_i(q,w_{i,|q|})$ 的一次拒绝运行，其中必定包含一个层级 $i-1$ 上的错误答案。
- 如果回答是否定的，并且 $w_{i,\ |q|} \notin S_{i,\ |q|}$，那么肯定回答的证书就存在（如 $A_i(q)=1$），且生成器会利用某个见证 w 找到一个证书。如果 $w = w_{i,\ |q|}$，则加入所声称的内容。否则，根据 $q \in \Pi_{i,\ \text{YES}}$ 的见证的唯一性，该证书中 $V_i(q,\ w)$ 的接受运行在层级 $i-1$ 处包含一个错误回答。

□

注释 5.2。生成器遵循 *Hartmanis* 和 *Hemachandra* 对稳健范畴机器进行模拟的策略 *[HH90, 定理 2.1]*：集合 $S_{i,\ m}$ 是在那里获知的预言机回答的更复杂的等价形式。

推论 5.3。$\mathbf{OUH_\bullet} \subseteq \mathbf{OUM}$ 和 $\mathbf{OUH_\bullet} \subseteq \mathbf{P}^{\mathbf{ONP}}$。

证明。定理5.1将 $\mathbf{OUH_\bullet}$ 置于$\mathbf{OGTD}[\mathbf{NP}]$ 中。引理3.1 证明了该类等于$\mathbf{OGTD}[\mathbf{ONP}]$（从而将其置于 $\mathbf{P}^{\mathbf{ONP}}$ 中，参见命题 2.5），并将其加强为类$\mathbf{OUGTD}[\mathbf{FP}^{\mathbf{ONP}}$、$\mathbf{co\text{-}ONP}$、$\mathbf{P}]$（从而将其置于 $\mathbf{OUM}$）。 □

6 层级的前几层

本节将梳理更小类中的情况。我们考察对层次的构造施加更多限制后会发生什么。我们可以问：如果尝试用点算子定义无偏单值层级，会发生什么？我们将证明，它出乎意料地坍缩了。我们还可以问：将我们的点算子应用于承诺问题会怎样？它不会改变该类。此外，点算子与单次预言机查询有些相似——我们将证明，对于这类查询，在我们的语境下可以用语言替代承诺预言机（这与我们对承诺 $\mathbf{RP}$ 和承诺 $\mathbf{ONP}$ 查询所知或未知的情况不同）。最后，我们指出，在更小的类 $\mathbf{OUP} \cap \mathbf{co\text{-}OUP}$ 中，实际上可以让两部分使用相同的见证，从而使该类等于$\mathbf{SOUP}$这一类；该类最近由[HVW26]定义。

6.1 通过点算子定义的层次坍缩

我们先介绍算子 $\exists_{\mathbf{ou}}\cdot$（定义 2.3）的两个简单性质，然后直接进入我们的第一个坍缩结果。

命题 6.1。

1. $\exists_{\mathbf{ou}}\cdot\mathbf{pr}\mathcal{C} = \exists_{\mathbf{ou}}\cdot\mathcal{C}$ 对于 $\mathcal{C} \in \{\mathbf{P},\ \mathbf{OUP}、\mathbf{co\text{-}OUP}\}$。*2.* $\exists_{\mathbf{ou}}\cdot\mathbf{OUP} = \mathbf{OUP}$。

证明。 1. 将 Π 的机器限制在满足 $|w| = p(|x|)$ 的点对 $(x,\ w)$ 上；这些点对全部属于承诺集，因此得到一个属于 $\mathcal{C}$ 的语言，可以用来替代 Π。

2. 将两个见证合并为一个（注意可能存在的长度冲突）。 □

定理 6.2。 $\exists_{\mathbf{ou}}\cdot\mathbf{co\text{-}OUP} = \mathbf{OUP}\cup\mathbf{co\text{-}OUP}$。

证明。见证长度为零意味着 co-OUP $\subseteq \exists_{\mathbf{ou}}\cdot$co-OUP，而 OUP $= \exists_{\mathbf{ou}}\cdot\mathrm{P} \subseteq \exists_{\mathbf{ou}}\cdot$co-OUP。反过来，设 $L \in \exists_{\mathbf{ou}}\cdot R$，其中 $R \in$ co-OUP，并设 V 是用于 R 、见证为 v_N 的无知觉无歧义验证器。若 $p \equiv 0$，则 $L \in$ co-OUP。否则，对于除有限多个 n 以外的所有情况， $p(n) \geq 1$ ；又由于有限多个输入长度可以单独处理，我们可以假设对于所有 n，$p(n) \geq 1$ 。至多有一个 w 满足 $(0^n, w) \in R$；因此，$(0^n, 0^{p(n)})$ 和 $(0^n, 1^{p(n)})$ 这两对中至少有一对属于 R，并且 v_N（其中 N 是这些对的长度）是唯一满足以下条件的字符串 v

$$V((0^n, 0^{p(n)}), v) = 1 \ \vee\ V((0^n, 1^{p(n)}), v) = 1$$

（在固定输入上测试候选无知见证）。输入 $x \in \{0,1\}^n$ 时，当且仅当此条件成立且 $V((x,\ w)\ v) = 0$ 时，接受一对 $(w,\ v)$。那么，对于 $x \in L$，$(w_n,\ v_N)$ 是唯一被接受的一对，而对于 $x \notin L$，没有任何一对会被接受。因此 $L \in \mathbf{OUP}$。 □

推论 6.3. 由$\mathbf{P}$ 通过交替应用 $\exists_{\mathbf{ou}}\cdot$ 和 $\forall_{\mathbf{ou}}\cdot$得到的层次会塌缩到第二层 $\exists_{\mathbf{ou}}\cdot\mathbf{co\text{-}OUP} = \forall_{\mathbf{ou}}\cdot\mathbf{OUP} = \mathbf{OUP}\cup\mathbf{co\text{-}OUP}$：该类在 $\exists_{\mathbf{ou}}\cdot$和 $\forall_{\mathbf{ou}}\cdot$下封闭。

证明。 显然， $\exists_{\mathbf{ou}}\cdot(\mathcal{C}\cup\mathcal{D}) = \exists_{\mathbf{ou}}\cdot\mathcal{C}\cup\exists_{\mathbf{ou}}\cdot\mathcal{D}$ 对所有类 $\mathcal{C}$ 和 $\mathcal{D}$ 均成立。应用命题6.1(2)和定理6.2并取补集。 □

这与多项式层次以及无关层次形成对比：它们通过预言机和点算子定义时得到相同的类（且已知不会塌缩）；此外，我们塌缩后的无歧义点算子层次的非无知觉对应物$\mathbf{AUPH}$ 相对于某个预言机是无限的 [ST07]。我们的塌缩实质上同时利用了无关性和无歧义性：由于外层见证是唯一的，内层co-OUP 谓词在固定对之一上为假： $(0^n,\ 0^{p(n)})$ 和 $(0^n,\ 1^{p(n)})$ ；又由于反驳该谓词的见证是无关的，因此可以在该对上进行检验。

6.2 单次查询情形

对于确定性机器，Cai、Hemachandra 和 Vysko˘c [CHV93，命题 2。6]指出，对 $\mathbf{prUP}$进行受保护的非自适应访问，并不比访问 $\mathbf{UP}$更强：给每个查询都附上输入作为标记，就能得到一个属于 $\mathbf{UP}$ 的语言。我们证明，在无歧义计算中，只要单个宽松查询的答案会影响结果，该查询就受到保护，因此标记论证可以适用。

引理 6.4。设 Π 是一个承诺问题，V 是一个多项式时间预言机验证器，它对每个输入 (x,w) 至多进行一次查询 $q(x,w)$，且见证 $L\in UP^{\Pi}$。设 T 为满足以下条件的点对 (x,w) 的集合：$V(x,w)$ 的输出取决于对 $q(x,w)$ 的回答。那么 $T\in P$，并且对每个 (x,w)，$q(x,w)\in\Pi_{\mathrm{YES}}\cup\Pi_{\mathrm{NO}}$。

证明。要判定 T，分别用两种答案运行 $V(x,w)$。令 $(x,w)\in T$，并假设 $q=q(x,w)$ 不在承诺范围内。取与 Π 一致、且仅在 q 上不同的语言 O_0 和 O_1。那么，对于恰好一个 b，$V^{O_b}(x,w)=1$。若 $x\notin L$，这与 V^{O_b} 的正确性矛盾。若 $x\in L$，则 w 是 V^{O_b} 在 x 上的接受见证；由于接受见证不依赖于一致语言（定义 2.1），$V^{O_{1-b}}(x,w)=1$，矛盾。 □

定理 6.5. $\mathbf{UP}^{\mathbf{prUP}[1]}=\mathbf{UP}^{\mathbf{UP}[1]}$ 和 $\mathbf{OUP}^{\mathbf{prOUP}[1]}=\mathbf{OUP}^{\mathbf{OUP}[1]}$。

证明。“$\supseteq$”中的包含关系显然。设 V 和 Π 如引理 6.4 所述，其中 V 见证 $L\in\mathrm{OUP}^{\Pi}$ 和 $\Pi\in\mathrm{prOUP}$，其验证器为 U（UP 的情形类似且更简单）。令 $\langle x、w\rangle$ 表示经过填充的数对 (x,w)，使其长度能够确定 $|x|$ 和 $|q(x,w)|$，并令

$$A=\{\langle x,w\rangle:(x,w)\in T\text{ and }q(x,w)\in\Pi_{\mathrm{YES}}\}.$$

接下来，$A\in\mathrm{OUP}$：对于输入 $\langle x,w\rangle$，若 $(x,w)\notin T$，则拒绝；否则在 $q(x,w)$ 上运行 U，该项属于承诺集；同一长度的 A 中所有成员都有等长的查询，因此也有相同的见证。新的验证器在 $(x,w)\notin T$ 时无需预言机即可计算 $V(x,w)$；否则通过查询 $\langle x,w\rangle\in A$ 来回答查询 $q(x,w)$。对于每个输入 (x,w)，它给出的结果都与某个固定语言 O 对应的 $V^{O}(x,w)$ 相同，该语言与 Π 一致；因此，它以相同的见证证明了 $L\in\mathrm{OUP}^{A}$。 □

作为比较，对 prRP 进行一次宽松查询可得 $\mathrm{RP}^{\mathbf{prRP}[1]}=\mathrm{BPP}$（根据 [BF99 的证明，定理 1]），而相对于预言机 [BF99，定理 3]，$\mathrm{P}=\mathrm{RP}\neq\mathrm{BPP}$。同样，对 prONP 进行一次宽松查询可得 $\mathrm{ONP}^{\mathbf{prONP}[1]}=\exists_{\mathbf{o}}\cdot\text{pr co-ONP}=\mathrm{O}_2^{\mathrm{p}}$ [HVW26，定理 4.7]；而 $\mathrm{ONP}^{\mathbf{ONP}[1]}=\mathrm{ONP}^{\mathbf{ONP}}=\exists_{\mathbf{o}}\cdot\text{co-ONP}$ [HVW26，定理 4.4]，，目前尚不清楚这两个类是否不同。

6.3 关于 OUP ∩co-OUP 的说明

事实证明，$\mathbf{OUP}\cap\mathbf{co\text{-}OUP}$ 与类 $\mathbf{SOUP}$ [HVW26，定义 2。7] 中的语言 L 满足以下条件：L 以及$\overline{L}$ 都具有使用相同无视无歧义见证的无知觉无歧义验证器。因此，它继承了$\mathbf{SOUP}$的多项式时间图灵闭包性质；由此还可得出，它是自低的。

命题 6.6. $\mathbf{SOUP}=\mathbf{OUP}\cap\mathbf{co\text{-}OUP}$。因此，$\mathbf{OUP}^{\mathbf{SOUP}}=\mathbf{OUP}$ 且$\mathbf{SOUP}^{\mathbf{SOUP}}=\mathbf{SOUP}$。

证明。包含关系$\mathbf{SOUP}\subseteq\mathbf{OUP}\cap\mathbf{co\text{-}OUP}$ 显然成立。反过来，设 Y和Z 分别是 L和 $\overline{L}$的无知觉无歧义验证器，且（不失一般性）见证长度相同。对于输入 $\overline{x}\in\{0,1\}^n$，猜测一位比特 b 和一个字符串 w，并检查

$$(b=1\wedge Y(0^n,w)=1)\ \vee\ (b=0\wedge Z(0^n,w)=1).$$

由于 0^n 要么属于 L，要么属于 $\overline{L}$，因此恰有一对通过此项检查。如果 $b=1$，其第二个分量就是 Y的无知见证，因此 $Y(x,w)=L(x)$。如果 $b=0$，同理 $1-Z(x,w)=L(x)$。无论哪种情况，这一对 (b,w) 都能证明任一答案。

令 $A\in\mathbf{SOUP}$。由于 $\mathbf{P}^{A}\subseteq\mathbf{SOUP}$（因为 $\mathbf{SOUP}$ 在多项式时间图灵归约下封闭 [HVW26]），命题 6.1(2) 给出 $\mathbf{OUP}^{A}=\exists_{\mathbf{ou}}\cdot\mathbf{P}^{A}\subseteq\exists_{\mathbf{ou}}\cdot\mathbf{OUP}=\mathbf{OUP}$；因此$\mathbf{co\text{-}OUP}^{A}\subseteq\mathbf{co\text{-}OUP}$ 且 $\mathbf{SOUP}^{A}\subseteq\mathbf{OUP}^{A}\cap\mathbf{co\text{-}OUP}^{A}\subseteq\mathbf{SOUP}$。 □

类 $\mathbf{SOUP}$ 是 $\mathbf{YP}^*$ [HVW26], 和 $\mathbf{OUP} \cap \mathbf{co\text{-}OUP}$ 的无歧义版本，而 $\mathbf{YP} = \mathbf{ONP} \cap \mathbf{co\text{-}ONP}$ [AD14]；命题 6.6 的证明依赖于唯一性，且无法给出 $\mathbf{YP} = \mathbf{YP}^*$。

致谢

作者谨向 Ilya Volkovich 和 Ryan Williams 致谢。他与他们共同开启了对无歧义无知类的研究 [HVW26，HV26b]。

参考文献

[Aar07] S. Aaronson. 量子态的可学习性。英国皇家学会学报 A辑，463(2088):3089–3114，2007。[AB09] S. Arora 和 B. Barak。《计算复杂性：现代方法》。剑桥大学出版社，2009。[AD14] S. Aaronson 和 A. Drucker。量子建议的完整刻画。SIAM 计算学报，43(3):1131–1183，2014。[BF99] H. Buhrman 和 L. Fortnow。概率计算中的单侧错误与双侧错误。载于 STACS 1999，LNCS 1563，第 100–109 页，1999。[BFHL13] H. Buhrman 、 L. Fortnow 、 J. M. Hitchcock 和 B. Loff。到稀疏集的学习归约。载于 MFCS 2013，LNCS 8087，第 243–253 页，2013。[BGM06] E. B¨ohler、C. Glaßer 和 D. Meister。MA 与 AM 之间的有误差界的概率计算。计算机与系统科学杂志，72(6):1043–1076，2006。[CGRS04] M. Crˆasmaru、C. Glaßer、K. W. Regan 和 S. Sengupta。串行化唯一策略的协议。载于 MFCS 2004，LNCS 3153，第 660–672 页，2004。[CHV93] J.-Y. Cai、L. A. Hemachandra 和 J. Vysko˘c。承诺与容错数据库访问。载于 K. Ambos-Spies、S. Homer 和 U. Sch¨oning 编，《复杂性理论：当前研究》，第 101–146 页。剑桥大学出版社，1993。初步版本见技术报告 TR 447，罗切斯特大学计算机科学系，1993，https://hdl.handle.net/1802/599。[CR06] V. T. Chakaravarthy 和 S. Roy。无知对称交替。载于 STACS 2006，LNCS 3884，第 230–241 页，2006。[CR11] V. T. Chakaravarthy 和 S. Roy。作为预言机的 Arthur 与 Merlin。计算复杂性，20(3):505–558，2011。[FSW09] L. Fortnow、R. Santhanam 和 R. Williams。固定多项式规模电路界。载于 CCC 2009，第 19–26 页，2009。[Gav95] R. Gavald`a。建议函数复杂性的界。计算机与系统科学杂志，50(3):468–475，1995。[GM15] O. Goldreich 和 O. Meir。输入无关证明系统与关于 P/poly 的一致复杂性视角。ACM 计算理论汇刊，7(4):16:1–16:13，2015。

[GS88] J. Grollmann 和 A. L. Selman。公钥密码系统的复杂性度量。SIAM 计算学报，17(2):309–335，1988。[HH90] J. Hartmanis 和 L. A. Hemachandra。鲁棒机器接受易集。理论计算机科学，74(2):217–225，1990。[HO02] L. A. Hemaspaandra 和 M. Ogihara。《复杂性理论指南》。理论计算机科学丛书。EATCS 丛书。施普林格，2002。[HR97] L. A. Hemaspaandra 和 J. Rothe。无歧义计算：布尔层级与稀疏图灵完全集。SIAM 计算学报，26(3):634–653，1997。[HS15] E. A. Hirsch 和 D. Sokolov。关于宽松无歧义层次的概率闭包。信息处理快报，115(9):725–730，2015。[HV26a] E. A. Hirsch 和 I. Volkovich。线性排序原理的上下界。载于 STACS 2026，LIPIcs 364，第 52:1–52:19 页，2026。完整版：ECCC TR25-142，修订版 2，2026，https://eccc.weizmann.ac.il/report/2025/142/revision/2/download。[HV26b] E. A. Hirsch 和 I. Volkovich。最终的 Karp–Lipton 坍缩。arXiv:2611.16096，2026。[HVW26] E. A. Hirsch、I. Volkovich 和 R. Williams。多项式层次第二层中的输入无关复杂性类。ECCC TR26-250，修订版 2，2026，https://eccc.weizmann.ac.il/report/2026/250/revision/2/download。[K¨am91] J. K¨amper。非一致证明系统：描述非一致与概率复杂性类的新框架。理论计算机科学，85(2):305–331，1991。[KP24] O. Korten 和 T. Pitassi。强值域回避与弱值域回避，以及线性排序原理。载于 FOCS 2024，第 1388–1407 页，2024。完整版：ECCC TR24-076，2024，https://eccc.weizmann.ac.il/report/2024/076/download。[Kre88] M. W. Krentel。优化问题的复杂性。计算机与系统科学杂志，36(3):490–509，1988。[KS85] K.-I. Ko 和 U. Sch¨oning。关于 NP 中的电路规模复杂性与低层级。SIAM 计算学报，14(1):41–51，1985。[KT94] J. K¨obler 和 T. Thierauf。复杂性受限建议函数。SIAM 计算学报，23(2):261–275，1994。[LR94] K.-J. Lange 和 P. Rossmanith。无歧义多项式层级与指数规模。载于《第九届复杂性理论结构年会论文集》，第 106–115 页。IEEE，1994。[MPR26] D. Miloschewsky、S. Podder 和 D. Rudolph。应认真对待承诺：关于承诺问题的相对化。arXiv:2609.07945，2026。[NR98] R. Niedermeier 和 P. Rossmanith。无歧义计算与局部可定义接受类型。理论计算机科学，194(1–2):137–161，1998。[Sch89] U. Sch¨oning。概率复杂性类与低性。计算机与系统科学杂志，39(1):84–100，1989。

[ST07] H. Spakowski 和 R. Tripathi. 交替机中无歧义性的力量。*Theory Comput. Syst.*, 41(2):291–326, 2007.[ST09] H. Spakowski 和 R. Tripathi. 层次无歧义性。*SIAM* 计算学报, 38(5):2079–2112, 2009.[Wag86] K. W. Wagner. 简洁输入表示下组合问题的复杂性。*Acta Inform.*, 23(3):325–356, 1986.[Zac88] S. Zachos. 概率量词与博弈。计算机与系统科学杂志, 36(3):433–451, 1988.[ZF87] S. Zachos 和 M. F¨urer. 概率量词与不可信对手。载于*FSTTCS 1987*, LNCS 287, 第 443–455 页, 1987.