第 5 章　交互证明系统

数学中的证明是满足一定语法的符号串，在逻辑上，该符号串描述的是一个推理过程。定理的证明过程可能极富创造性，但验证证明正确性的过程是一个简单的机械过程。对我们所知的形式化系统，如皮亚诺算术和策梅洛-弗兰克尔集合论，其验证都是多项式时间可计算的。既然验证是个计算过程，那么我们可以从计算的视角审视证明-验证系统。在第 2 章，我们讨论过这类系统。任何一个 NP 问题 L 都有一个验证器 $\mathbb{M}$，当给定问题实例 x 和证书 u 时，$\mathbb{M}(x,u)$ 能算出 $x \in L$ 是否成立。这里 $x \in L$ 是“定理”，u 是“证明”，$\mathbb{M}$ 定义了验证算法。NP 问题的验证系统过于简单，为了一般化，我们从一个更加动态的角度来看一下验证过程。一位在读博士生给出的一个证明会用到一些已经证明的引理和定理，其导师在验证该证明时，不需验证被引用的那些引理和定理的证明，而只需关注博士生给出的证明本身的逻辑推导过程。为了快速验证博士生的证明的正确性，导师可以和博士生进行一次面对面交流。期间，导师向博士生提出问题，博士生必须回答所有的提问。导师最终根据博士生提供的所有信息，判定证明是否有效。在动态场景中，导师和博士生构成一个问答系统，导师持怀疑态度，博士生尽全力说服导师。导师可提多个问题，博士生必须正确地回答每个提问。问答的目的是让导师用尽可能少的时间判定证明是否为真。不能排除下述情况：博士生的证明可能含有一个他自己未意识到的错误，在问答过程中，导师所提问题并未涉及到该错误，最终导师做出了一个错误判断。

interactive proof
verifier, prover

可将上述的导师-博士生问答系统形式化为一个计算系统。交互证明系统有两个参与者，验证者和证明者。验证者的目的是通过和证明者对话获得信息并以此判定给定的命题是否为真，证明者的任务是尽可能地说服验证者相信当前的证明是真的。从实际可计算的角度，我们对验证者和证明者做如下限制：

I. 验证者的工作必须是容易的，即验证者是一台多项式时间图灵机。

II. 证明者的计算能力没有限制，但证明者对问题的回复必须是短的。

设 x 为当前要验证的命题，验证者的计算时间是 $\texttt{poly}(|x|)$，证明者的答复长度也是 $\texttt{poly}(|x|)$。如果对验证者的计算时间不做任何限制，验证者完全可以自己完成计算而无需向证明者提任何问题。一旦对验证者的计算能力做了上述限制，验证者就无法接受证明者给出的指数长的回复。设 $\mathbb{V}$ 和 $\mathbb{P}:\{0,1\}^*\to\{0,1\}^*$ 分别为验证者图灵机和证明者图灵机。$\mathbb{V}$ 和 $\mathbb{P}$ 在输入 $x\in\{0,1\}^*$ 上的 k-轮交互是一个 0-1 串序列 $a_1,\cdots,a_k\in\{0,1\}^*$，其定义如下：

$$
\begin{aligned}
a_1 &= \mathbb{V}(x)\\
a_2 &= \mathbb{P}(x,a_1)\\
&\vdots\\
a_{2i+1} &= \mathbb{V}(x,a_1,\cdots,a_{2i}),\quad \text{若 } 2i<k\\
a_{2i+2} &= \mathbb{P}(x,a_1,\cdots,a_{2i+1}),\quad \text{若 } 2i+1<k\\
&\vdots
\end{aligned}
$$

将此交互记为 $\langle\mathbb{V},\mathbb{P}\rangle(x)$。称 $\mathbb{V}(x,a_1,\cdots,a_k)\in\{0,1\}$ 为交互的结果，记为 $\mathsf{out}_{\mathbb{V}}\langle\mathbb{V},\mathbb{P}\rangle(x)$。因验证者必须在多项式时间给出判定结果，可以假定 k 是多项式。值得注意的是，我们称验证者的一个提问为一轮，证明者的一个回复也为一轮，称一个验证者和证明者的问答为一个回合。交互协议可规定验证者先问，也可以让证明者先答。在 NP 问题的一轮交互证明中，证明者先给答案，验证者随即做出判断。

可以用交互证明系统定义问题类。设 L 为一语言，$\mathbb{V}$ 为验证者，k 为多项式。若对任意输入实例 x，$\mathbb{V}$ 可以和任意证明者 $\mathbb{P}$ 进行 $k(|x|)$ 轮交互，并且交互的结果满足下述的完备性和可靠性，称 L 具有 k-轮交互证明系统。

$$
\begin{aligned}
&\text{完备性：}\quad x\in L\Rightarrow\exists\mathbb{P}:\{0,1\}^*\to\{0,1\}^*.\mathsf{out}_{\mathbb{V}}(\mathbb{V},\mathbb{P})=1\\
&\text{可靠性：}\quad x\notin L\Rightarrow\forall\mathbb{P}:\{0,1\}^*\to\{0,1\}^*.\mathsf{out}_{\mathbb{V}}(\mathbb{V},\mathbb{P})=0
\end{aligned}
\tag{5.0.1}
$$

完备性和可靠性一起，确保验证者的判定结果永远是对的。读者必然会问，什么样的语言具有交互证明系统？请看下面的推导：

$$
\begin{aligned}
x \in L \quad & \text{当仅当} \quad \exists \mathbb{P} : \{0,1\}^* \to \{0,1\}^*.\mathsf{out}_{\mathbb{V}}(\mathbb{V},\mathbb{P}) = 1 \\
& \text{当仅当} \quad \exists a_1, a_2, \cdots, a_k.\mathbb{V}(x) = a_1 \wedge \mathbb{V}(x, a_1, a_2) = a_3 \\
& \qquad \wedge \cdots \wedge \mathbb{V}(x, a_1, \cdots, a_k) = 1
\end{aligned} \tag{5.0.2}
$$

串 $a_1, a_2, \cdots, a_k$ 是多项式长的，$\mathbb{V}(x), \mathbb{V}(x, a_1, a_2), \cdots, \mathbb{V}(x, a_1, \cdots, a_k)$ 是多项式时间可计算的，故 $\mathbb{V}(x) = a_1 \wedge \mathbb{V}(x, a_1, a_2) = a_3 \wedge \cdots \wedge \mathbb{V}(x, a_1, \cdots, a_k) = 1$ 是多项式时间可判定的。因此，上述等价关系告诉我们，$L \in \mathbf{NP}$。换言之，一个多项式轮的交互证明系统和一个 1-轮交互证明系统是等价的！认真思考上述的等价性推导，读者就会恍然大悟：验证者其实无需计算 $a_1, a_3, \cdots$，证明者可以模拟 $\mathbb{V}$ 计算出这些问题，然后一次性将 $a_1, a_2, a_3, \cdots, a_k$ 发送给验证者。一台确定图灵机计算出的问题都是可预测的，而可预测的计算对证明者而言不构成任何挑战。要使交互证明系统定义出有意思的复杂性类，验证者必须问更 “聪明” 的问题。何谓 “聪明” 的问题？从计算的角度，不可预测的问题就是聪明的问题。下面是个常用的例子，很好地解释了为什么 “不可预测的问题 = 聪明的问题”：甲患色盲症，乙希望知道甲是否能区别红色和绿色。做法是乙左手拿一个红球，右手拿一个同样大小的绿球，乙让甲闭上眼睛，投硬币，若面朝上，乙将红球和绿球换手，若面朝下，则什么也不做，然后乙让甲睁开眼，问甲两个球是否换手了。如果甲每次都能答对，乙就有很大的把握判断甲能区分红和绿。如果甲答对的概率和投硬币差不多，则有很大的概率甲无法区别红色和绿色。这个例子告诉我们，如果我们让验证者随机地提问，我们能得到强得多的交互证明系统。因此，我们将前述的对验证者的限制更新如下：

III. 验证者 $\mathbb{V}$ 是一台多项式时间概率图灵机，即 $\mathbb{V} \in \mathbf{BPP}$。

限制 III 的一个明显的推论是，验证者做出的最终判断一般不满足（5.0.1）中所陈述的完备性，我们得允许验证者犯小概率错误。在上述例子中，如果乙将交互协议重复 10 遍，其犯错概率可降至 2^{-10}。

综上所述，我们感兴趣的交互证明系统有三个核心概念，即交互（问与答）、提问的随机性、允许小概率错误，三者缺一不可。利用随机性，我们似乎可以为如下形式的量化布尔公式

$$\forall x_1 \exists x_2 \cdots Q_n x_x.\varphi(x_1, \cdots, x_n)$$

设计一个简单的交互系统：验证者随机地取 x_1 的一个赋值，证明者给出 x_2 的

一个赋值，以此类推，最后验证者对 $\varphi(x_1,\cdots,x_n)$ 进行求值并给出结果。这个系统的问题在于很难对验证算法的出错概率进行分析。我们将证明，**PSPACE** 中的每个问题都有交互证明系统，我们的任务是为验证者找出我们能提供正确性证明的算法。

交互证明系统在密码学、近似算法、复杂性理论中有很多重要应用。本章将讨论满足限制 II 和限制 III 的交互证明系统的性质和表达能力。

5.1　私币交互证明

戈德瓦塞尔、米卡利和拉科夫在二十世纪八十年代中叶提出并研究了私币交互证明系统 [91]。在这类交互证明系统中，验证者在交互开始前要生成一个多项式长的随机串 $r\in_{\mathrm{R}}\{0,1\}^l$，然后用此随机串计算多项式个随机的问题，即

Goldwasser, Micali, Rackoff private-coin

$$a_1=\mathbb{V}(x,r),\ a_3=\mathbb{V}(x,r,a_1,a_2),\ \cdots$$

证明者无法看到随机串 r，所以依旧有

$$a_2=\mathbb{P}(x,a_1),\ a_4=\mathbb{P}(x,a_1,a_2,a_3),\ \cdots$$

在此框架下，验证者的判定结果 $\mathsf{out}_{\mathbb{V}}\langle\mathbb{V},\mathbb{P}\rangle(x)$ 可视为一个随机变量。

设 k 为多项式。类型 $\mathbf{IP}[k(n)]$ 的定义如下：$L\in\mathbf{IP}[k(n)]$ 当仅当存在多项式时间概率图灵机（验证者）$\mathbb{V}$，当输入为 x 时，可和任意图灵机（证明者）$\mathbb{P}$ 进行 $k(|x|)$-轮交互，然后验证者做出判定，其判定结果满足如下定义的完备性和可靠性：

$$\text{完备性：}\quad x\in L\Rightarrow\exists\mathbb{P}:\{0,1\}^*\to\{0,1\}^*.\Pr[\mathsf{out}_{\mathbb{V}}(\mathbb{V},\mathbb{P})=1]\geqslant 2/3 \tag{5.1.1}$$

$$\text{可靠性：}\quad x\notin L\Rightarrow\forall\mathbb{P}:\{0,1\}^*\to\{0,1\}^*.\Pr[\mathsf{out}_{\mathbb{V}}(\mathbb{V},\mathbb{P})=1]\leqslant 1/3 \tag{5.1.2}$$

复杂性类 **IP** 定义为 $\bigcup_{c\geqslant 1}\mathbf{IP}[cn^c]$。完备性参数 2/3 和可靠性参数 1/3 只是无数可供选择的参数对中的一个，这对参数甚至可以依赖于输入长度。在证明此事实之前，我们先看一下交互证明系统的若干性质。

我们首先刻画证明者的计算能力，这点对我们后续的讨论极其重要。对特定的验证者 $\mathbb{V}$，我们可以设计一个最优的证明者 $\mathbb{P}_{opt}$。设 $\mathbb{V}$ 使用的随机串长度为 l，证明者的答案长度为 d。假定已经进行了 j-轮交互，问题和答案串为

$a_1, \cdots, a_j$。在第 $(j+1)$-轮，$\mathbb{P}_{opt}$ 必须给出一个在概率意义上最优的答案。证明者用暴力法选出最优回答：对每个可能的答案 a，证明者依次考察所有长度为 l 的随机串，用暴力法算出让验证者接受输入的最大条件概率 p_a。因为交互的轮数是多项式的，所以在多项式空间里 $\mathbb{P}_{\text{opt}}$ 可以算出 p_a。因此，$\mathbb{P}_{\text{opt}}$ 在多项式空间里可以算出最佳答案 $a = \max_{a\in\{0,1\}^d} p_a$。我们可将交互证明系统的限制 II 更新如下：

IV. 证明者 $\mathbb{P}$ 是一台多项式空间图灵机，即 $\mathbb{P} \in$ **PSPACE**。

限制 IV 极大地简化了可靠性证明，我们不需要说明不等式 (5.1.2) 对所有的证明者都成立，而只需说明它对那个最优的证明者成立即可。而且，最优的证明者可以处理所有的输入。因此，我们可以给出下述简化了的定义。

定义 5.1 $L \in$ **IP**$[k(n)]$ 当仅当存在验证者 $\mathbb{V} \in$ **BPP** 和证明者 $\mathbb{P} \in$ **PSPACE**，当输入为 x 时，$\mathbb{V}$ 和 $\mathbb{P}$ 进行 $k(|x|)$-轮交互，交互的结果满足如下条件：

$$\text{完备性：}\quad x \in L \Rightarrow \Pr[\text{out}_{\mathbb{V}}(\mathbb{V},\mathbb{P}) = 1] \geqslant 2/3 \tag{5.1.3}$$

$$\text{可靠性：}\quad x \notin L \Rightarrow \Pr[\text{out}_{\mathbb{V}}(\mathbb{V},\mathbb{P}) = 1] \leqslant 1/3 \tag{5.1.4}$$

利用此定义，容易推出 **IP** 的一些重要性质。

命题 25 **IP** $\subseteq$ **PSPACE**。

证明 因为 **BPP** $\subseteq$ **PSPACE**，所以一台多项式空间图灵机既可以模拟证明者，也可以模拟验证者，因而可以把所有的交互过程模拟一遍并计算出成功概率。当成功概率大于等于 2/3 时回答“是”，小于等于 1/3 时回答“否”。 □

下述引理的证明是另一个利用限制 IV 的例子。

引理 5.1 在复杂性类 **IP** 的定义中，可将完备性参数 2/3 换成 $1-2^{-n^s}$，可靠性参数 1/3 换成 2^{-n^s}。

证明 将交互协议重复 $O(n^s)$ 遍，按多数原则给出结果。由切诺夫不等式即可推得引理的结果。 □

在上述证明中，我们无需担心交互协议是顺序地还是并行地重复了 $O(n^s)$ 遍。只要我们每遍用的都是那个最优的证明者，以任何方式重复都不会影响验证结果。引理5.1告诉我们可以将可靠性参数指数地降低，但一般地，我们无法

降得更低。如果一个交互证明系统的可靠性参数为 0，(5.0.2) 中给出的等价关系就会成立，系统所判定的就是一个 NP 问题。所以一般的，我们必须放弃完美可靠性。换言之，在一个表达能力强于 NP 问题验证器的交互证明系统中，错误在所难免。另一方面，交互证明系统的可靠性参数可以为 1，称此类系统具有完美完备性。可能意想不到的是，完美完备性不是一个实质性限制。与引理5.1一样，下述命题反映了 **IP** 定义的稳健性。　robust

命题 26　*具有完美完备性的交互证明系统所接受的语言类也是* **IP**。

证明　根据命题25，每个 **IP** 中的问题都可卡普归约到 PSPACE-完全问题 QBF。我们将证明，QBF 有一个具有完美完备性的交互证明系统，见定理5.6。因此，我们可以为 **IP** 中的任意问题设计一个具有完美完备性的交互证明系统。　□

让我们看几个著名的交互证明系统。第一个例子取自 [89]。图同构 GI 是一个著名的未被准确定位的问题。我们已知，若 GI 是 NP-完全的，则多项式谱系塌陷到第二层，但我们不知 GI 是否在 **P** 中。另一方面，我们不知道图不同构问题 NGI是否在 **NP** 里，但我们可以为 NGI 设计一个两轮的交互证明系统。给定具有 n 个结点的两个图 G_0 和 G_1，用 $1,\cdots,n$ 表示图的结点。图 G_0 和 G_1 之间的一个同构映射定义了 $[n]$ 上的一个置换。反之，$[n]$ 上的一个置换可作为 G_0 和 G_1 同构的证据。基于这一现象，可设计图5.1中定义的交互证明系统，其中 V 表示验证者，P 表示证明者。

V：选$i\in_{\mathrm{R}}\{0,1\}$；对G_i的结点做一随机置换得到图H；将H发给P。

P：判定H是由G_0还是G_1生成，若为前者，将0发给V，否则将1发给V。

若从P接收的和i一样，V接受，否则V拒绝。

图 5.1　图不同构协议

命题 27　*图不同构问题在* **IP** *中。*

证明　用图5.1中定义的协议，可知：若 G_0 和 G_1 不同构，证明者能准确地判定 H 是由哪个图生成的。若 G_0 和 G_1 同构，证明者只能猜，猜对的概率是 1/2。因此 NGI 有一个具有完美完备性的两轮交互证明系统。　□

quadratic residue

第二个例子源自 [91]。设 a 和 p 为自然数，若存在满足 $a \equiv b^2 \pmod{p}$ 的自然数 b，称 a 是模 p 的二次剩余。非二次剩余问题 QNR 包含满足下述条件的所有的自然数对 (a, p)：

1. $p > 1$ 为素数，$a \neq 0$。
2. a 不是模 p 的二次剩余。

易见，$\overline{\text{QNR}} \in \mathbf{NP}$。

尽管素数判定问题有多项式算法 [7]，目前我们尚不知道 QNR 是否在 **NP** 中。与前例一样，我们可为 QNR 设计一个两轮交互协议，见图5.2。

> V：选$r \in_{\mathrm{R}} \mathbf{F}_p$ 和$i \in_{\mathrm{R}}\{0, 1\}$；若$i = 0$，将$r^2 \bmod p$发给P，否则将$ar^2 \bmod p$发给P。
>
> P：判定是否$i = 0$，若是，将0发给V，若否，将1发给V。
>
> 若从P接收的和i一样，V接受，否则V拒绝。

图 5.2 二次剩余协议

命题 28 非二次剩余问题在 **IP** 中。

证明 图5.2中定义的协议基于有限域 $\mathbf{F}_p$ 中的运算性质。若 a 不是二次剩余，则 ar^2 也不是二次剩余。这是因为在有限域 $\mathbf{F}_p$ 中：若 $ar^2 = c^2$，则 $a = (c/r)^2$，即 a 为模 p 的二次剩余，矛盾。这种情况下，证明者一定能将二次剩余 r^2 和非二次剩余 ar^2 区分，让验证者接受 a。若 $a \equiv b^2 \pmod{p}$，即 a 为模 p 的二次剩余，则 r^2 和 ar^2 均为二次剩余。由于对任意 $b \in \mathbf{F}_p$，函数 $f(x) = bx$ 是有限域 $\mathbf{F}_p \setminus \{0\}$ 上的双射，所以 br 和 r 都是 $\mathbf{F}_p$ 上的一致分布。在这种情况下，证明者只能猜测。结论：QNR 有具有完美完备性的两轮交互证明系统。 □

cofactor

在第三个例子里，我们为积和式计算设计一个交互证明系统，这是隆德、福特纳、卡洛夫和尼桑在文 [150] 中给出的。设 $\boldsymbol{A} = (a_{j,k})_{1 \leqslant j,k \leqslant n}$ 为定义在有限域 $\mathbf{F}_p$ 上的 $(n \times n)$-方阵。矩阵 $\boldsymbol{A}$ 中元素 $a_{i,j}$ 的余子式 $\boldsymbol{A}_{i,j}$ 是删去 $\boldsymbol{A}$ 中第 i 行和第 j 列后得到的 $((n-1) \times (n-1))$-方阵。矩阵 $\boldsymbol{A}$ 的积和式 $\mathtt{perm}(\boldsymbol{A})$ 可由下列递归等式计算。

$$\mathtt{perm}(\boldsymbol{A}) = \sum_{i=1}^{n} a_{1i} \mathtt{perm}(\boldsymbol{A}_{1,i}) \tag{5.1.5}$$

式 (5.1.5) 给出了积和式的自归约算法。所谓自归约指的是将一个问题的求解分解成几个比该问题规模小的同类问题的求解。当给定自然数 k 和方阵 $\boldsymbol{A}$ 后，我们的交互证明系统能大概率地验证 $k=\mathtt{perm}(\boldsymbol{A})$ 是否成立。利用自归约，可以设计如下的协议：

self-reduction

1. 证明者将 n 个自然数 $k_1,\cdots,k_n$ 发给验证者。
2. 验证者测试 $a_{11}k_1+\cdots+a_{1n}k_n$ 是否等于 k。若等式成立，则对每个$i\in[n]$，将 $(k_i,\boldsymbol{A}_{1,i})$ 作为协议的输入递归地验证。

此协议的问题是它的交互轮数是 $n!$。读者会马上想到，验证者无需测试所有需要测试的，它可以随机地选 $i\in[n]$，然后将 $(k_i,\boldsymbol{A}_{1,i})$ 作为协议的输入递归地验证。这一思路本质上是对的，但其误差分析较复杂。我们有一个实现这一思路的代数方案。首先将余子式 $\boldsymbol{A}_{1,1},\cdots,\boldsymbol{A}_{1,n}$ 用一个含变量 x 的矩阵 $\boldsymbol{A}(x)$ 表示，使得对所有 $i\in[n]$，有 $\boldsymbol{A}(i)=\boldsymbol{A}_{1,i}$。这里 $\boldsymbol{A}(x)$ 是 $((n-1)\times(n-1))$-方阵，它的每个元素是 x 的 $n-1$ 次多项式。矩阵 $\boldsymbol{A}(x)$ 的积和式 $\mathtt{perm}(\boldsymbol{A}(x))$ 是 x 的 $(n-1)^2$ 次多项式。必须强调的是，$\mathtt{perm}(\boldsymbol{A}(x))$ 无法在多项式时间算出，故需证明者提供。设

$$\boldsymbol{A}(x)_{j,k}=b_{n-1}x^{n-1}+b_{n-2}x^{n-2}+\cdots+b_1x+b_0$$

为求系数 $b_{n-1},b_{n-2},\cdots,b_1,b_0$，分别令 $x=1,\cdots,n$，得 n 元线性方程组

$$\begin{pmatrix}1 & 1 & \cdots & 1 & 1\\ \vdots & \vdots & \vdots & \vdots & \vdots\\ 1 & k & \cdots & k^{n-2} & k^{n-1}\\ 1 & k+1 & \cdots & (k+1)^{n-2} & (k+1)^{n-1}\\ \vdots & \vdots & \vdots & \vdots & \vdots\\ 1 & n & \cdots & n^{n-2} & n^{n-1}\end{pmatrix}\begin{pmatrix}b_0\\ \vdots\\ b_k\\ b_{k+1}\\ \vdots\\ b_{n-1}\end{pmatrix}=\begin{pmatrix}a_{(j+1)(k+1)}\\ \vdots\\ a_{(j+1)(k+1)}\\ a_{(j+1)k}\\ \vdots\\ a_{(j+1)k}\end{pmatrix}$$

上式中的矩阵是著名的范德蒙矩阵。根据范德蒙矩阵的非奇异性，上述方程有唯一解。解线性方程可在 $O(n^3)$ 时间完成，故验证者可算出 $\boldsymbol{A}(x)$。基于这些代数性质，我们给出图5.3中定义的积和式协议。设 PERM 为如下语言 $\{\langle\boldsymbol{A},p,k\rangle\mid p>n^4,\ k=\mathtt{perm}(\boldsymbol{A}),\ \boldsymbol{A}$为有限域$\mathbf{F}_p$上的$(n\times n)$-方阵$\}$。

Vandermonde

双方知道k和矩阵$\boldsymbol{A}$。证明者的目的是说服验证者$k=\mathrm{perm}(\boldsymbol{A})$成立。
若$k\neq\mathrm{perm}(\boldsymbol{A})$，验证者应该以较大的概率拒绝。
P：发送给V一个$(n-1)^2$-次一元多项式$f(x)$，该多项式应为$\mathrm{perm}(\boldsymbol{A}(x))$。
V：若$k\neq\sum_{i=1}^{n}a_{1i}f(i)$，拒绝，否则随机选$b\in_{\mathrm{R}}\mathbf{F}_p$，并要求P给出等式$f(b)=\mathrm{perm}(\boldsymbol{A}(b))$的证明。

图 5.3 积和式协议

命题 29 $\mathtt{PERM}\in\mathbf{IP}$。

证明 若$n\leqslant 3$，用暴力法进行验证，否则用积和式协议。若$k=\mathtt{perm}(\boldsymbol{A})$，证明者一定能让验证者接受输入。若$k\neq\mathtt{perm}(\boldsymbol{A})$，验证者的出错概率可估算如下：假定证明者给出的$f(x)$是假的，即$f(x)\neq\mathtt{perm}(\boldsymbol{A}(x))$但$k=\sum_{i=1}^{n}a_{1i}f(i)$。在这种情况下，方程$f(x)-\mathtt{perm}(\boldsymbol{A}(x))=0$最多有$(n-1)^2$个根。随机挑选的$b$为根之一的概率不超过$(n-1)^2/p$。若$b$的确为根，证明者被要求证明一个正确的等式，在这种情况下，证明者的欺骗行为将无法被发现，验证者必然做出错误的判定。若b不是根，证明者被要求证明一个错误的等式。如果证明者一直蒙混过关，他的欺骗行为一定会在倒数第三轮被发现。因此，出错概率不超过

$$\frac{(n-1)^2}{p}+\frac{(n-2)^2}{p}+\cdots+\frac{4^2}{p}<\frac{n^3}{p}<\frac{1}{n}<\frac{1}{3}$$

命题得证。□

隆德、福特纳、卡洛夫和尼桑的证明用到了立普顿发现的定义在有限域上的方阵的随机自归约性质 [147]，立普顿的证明基于文 [29] 中的构造。

根据户田定理（定理 4.15），多项式谱系中的问题可以库克归约到 ♯**P**-完全问题，而积和式问题是 ♯**P**-完全的，所以多项式谱系中的问题都有交互证明系统，即 **HP** $\subseteq$ **IP**。

5.2 公币交互证明

帕帕季米特里乌在 [167] 中写道：“可将不确定条件下的判定问题形式化为一类新的博弈——‘与大自然的对弈’。博弈一方随机地移动棋子，对博弈过程毫无兴趣；博弈的另一方试图遵守一个能极大化其赢率的策略。”为了刻画一类决策优化问题，帕帕季米特里乌对交替图灵机进行了修改，将所有 ∀-状态

换成随机状态，并用此类图灵机定义了多项式空间类 **PPSPACE**，证明了该类和 **PSPACE** 相同。可将这类图灵机的计算视为一博弈过程，大自然只管掷骰子，对手是一台多项式空间图灵机。显而易见，这类博弈等同于一类特殊的交互证明系统，在交互过程中，验证者每轮产生一随机串，并将随机串作为问题发给证明者；换个角度看，验证者在一开始产生一个随机串，然后在随后的交互中依次将随机串的一段告诉证明者。我们称这类系统为公币交互证明系统。 public-coin
在公币交互证明系统中，验证者除了掷骰子外无需做任何计算，因为证明者有能力做这些计算。

巴柏称公币交互证明系统为亚瑟-梅林博弈 [22]。亚瑟是验证者，梅林是证明者。若亚瑟先掷骰子，有 $k(n)$-轮的亚瑟-梅林博弈所接受的问题类用 $\mathbf{AM}[k(n)]$ 表示；若梅林先给出一个答案，有 $k(n)$-轮的亚瑟-梅林博弈所接受的问题类用 $\mathbf{MA}[k(n)]$ 表示。为了行文方便，我们称一个亚瑟先下棋的亚瑟-梅林博弈为亚瑟博弈，一个梅林先下棋的亚瑟-梅林博弈为梅林博弈。显然 Arthur-Merlin

$$\mathbf{AM}[k(n)], \mathbf{MA}[k(n)] \subseteq \mathbf{IP}[k(n)] \tag{5.2.1}$$

易见，$\mathbf{MA}[1] = \mathbf{NP}$ 和 $\mathbf{AM}[1] = \mathbf{BPP}$。由此两等式推出 $\mathbf{NP} \cup \mathbf{BPP} \subseteq \mathbf{MA}[2]$。

具有常数轮的亚瑟-梅林博弈常用 **A** 和 **M** 的一个交替串表示，如

$$\mathbf{MA},\ \mathbf{AM},\ \mathbf{AMA},\ \mathbf{MAM},\ \mathbf{MAMAMA}$$

等。这里 **MA** 表示可由两轮梅林博弈接受的语言类，**AMA** 表示可被三轮亚瑟博弈判定的语言。巴柏注意到一个重要的事实，即有限轮的亚瑟-梅林博弈没有看上去的那么复杂。首先看一个在 [22] 中称为交换引理的重要结果。 Switching Lemma

引理 5.2（交换引理）　$\mathbf{MA} \subseteq \mathbf{AM}$。

证明　设 $L \in \mathbf{MA}$。在接受 L 的两轮梅林博弈中，梅林先给出一个答案 a，然后亚瑟产生随机串 r，最后亚瑟做出判断。如果我们简单地交换梅林和亚瑟的动作次序，让亚瑟先掷骰子，然后梅林给出一个回答，最后亚瑟做出判断，那么从下面的推导可看出：交换后得到的亚瑟博弈的完备性参数不会降低，完美完备性会保留。

$$x \in \mathbf{L} \Rightarrow \exists a.\Pr_r[\mathbb{A}(x,a,r)=1] \geqslant 1-\epsilon \quad \Rightarrow \quad \Pr_r[\exists a.\mathbb{A}(x,a,r)=1] \geqslant 1-\epsilon$$

但可靠性会受到影响，这是因为

$$x \notin \mathbf{L} \Rightarrow \forall a.\Pr_r[\mathbb{A}(x,a,r)=1] \leqslant \epsilon \quad \Rightarrow \quad \Pr_r[\exists a.\mathbb{A}(x,a,r)=1] \leqslant 2^{|a|}\epsilon$$