树和公式的多项式指纹

Mihai Prunescu

Research Center for Logic, Optimization and Security (LOS),
Faculty of Mathematics and Computer Science,
University of Bucharest, Academiei 14, 010014 Bucharest, Romania

and Simion Stoilow Institute of Mathematics of the Romanian Academy,
Research unit 5, P. O. Box 1-764, RO-014700 Bucharest, Romania.

mihai.prunescu@imar.ro, mihai.prunescu@gmail.com

为了满足（零知识）证明对（数学）证明的需求，我们描述了一种将形式句子转化为整系数多元多项式上的 2×2-矩阵的方法，使得像分离规则或替换这样的常规证明步骤可以轻松地从对应项或公式参数的矩阵中计算出来。通过在适当选择的有限域中的随机元素评估多项式变量，该证明被一个数字序列所替代。只有与公理对应的值需要从头开始计算。导出公式的值可以通过应用同态性质来从其祖先对应的值计算得出。对于这样的序列，可以应用各种零知识方法。

1 动机

在公司 PiSquared 发布的杰出白皮书 [5] 中，概述了以下想法。为了验证某些计算的正确性，例如区块链交易，可以按照以下步骤进行：

- 证明给定的算法，在给定输入数据的情况下，必须导致通过计算获得的具体输出。
- 结果得到的数学证明，以及其他一般的数学证明一样，都是冗长且繁琐的。所以它们不能作为正确性证书使用，无论是因为它们的长度还是验证所需的时间。
- 在这一点上，可以使用**零知识证明**的数学证明方法，以将它们的正确性归结为检查一个更小得多的证书。这个证书的正确性在很大程度上意味着 $1-\varepsilon$ 数学证明的正确性。Blum[1] 可能是第一个表达通过可验证零知识程序来保障数学证明安全性的作者。
- 可以将证明转化为一串数字，而最方便的数字应该是大有限域中的元素。然后可以应用一些现代的零知识证明方法，而不是 Blum[1] 的早期方法。
- 上面的线条概述了概念**证明的证明**，象征性地 $\pi(\pi)$。这也解释了缩略词 PiSquared。
- 如果这种编码对于通常的证明步骤（如分离规则或用项替换变量）来说是同态的话，那会更好。这样一来，人们只需对初始步骤（比如公理或重言式的调用）应用复杂的（耗时的）编码，而其他证明步骤的编码则可以通过同态规则从已有的编码中计算得出。
- 此外，如果像分离规则和代换这样的证明步骤变得同态性，则可以通过算术运算来模拟它们。因此，可以使用算术电路来建模各种证明段落。这样做的话，人们就可以应用零知识证明方法来认证某些算术电路所做的计算的正确性，比如 CIRCOM 方法，[2]。

Submitted to:
FROM 2025

2 原理

以下策略结晶化：

1. 每个公式 φ 对应一个域元素（或由域元素组成的向量）$V(\varphi)$。同样，每个良构项 t 对应一个域元素（或由域元素组成的向量）$V(t)$。
2. 向量 $V(\varphi)$，分别 $V(t)$，包含一个子向量 $[[\varphi]]$，分别 $[[t]]$，直接编码 φ，分别 t。在此向量中出现的其他元素对于执行替换是有用的。任何子向量都是用于将变量 x_i 替换为其他项的辅助值。因此，指纹如下所示：
$$V(\varphi) = ([[\varphi]], [[\varphi]]_{x_1}, \ldots, [[\varphi]]_{x_k}).$$
3. 存在一个算术术语 $MP(a,b)$，具有以下性质。如果公式 φ_3 是通过将规则假言推理应用于公式 φ_1 和 φ_2 而得到的结果，则 $V(\varphi_3) = MP(V(\varphi_1), V(\varphi_2))$。
4. 存在一个算术项 $Subst_y(a,b)$，具有以下性质。如果公式 φ_3 是通过将公式 φ_1 中出现的变量 y 替换为公式 φ_2 而得到的结果，那么 $V(\varphi_3) = Subst_y(V(\varphi_1), V(\varphi_2))$。另外，如果公式 φ_3 是通过将公式 φ_1 中的变量 y 替换为项 t 得到的结果，则 $V(\varphi_3) = Subst_y(V(\varphi_1), V(t))$。

我们称向量 $V(\varphi)$ 为 φ 的**指纹**。
上述定义目前只是一个意向声明。这些意向必须通过以下一些其他条件来完成：

- 算法不应依赖于所选择的具体有限域 $\mathbb{F}_p$。
- 不同的公式必须映射到不同的元素上，至少在特征值 p 足够大时是这样，尽可能地。指纹识别必须是一个哈希函数。
- 对于公理 φ，计算 $V(\varphi)$ 是快速的。
- 给定 $V(\varphi_1)$ 和 $V(\varphi_2)$ 的值，计算
$$MP(V(\varphi_1), V(\varphi_2))$$
是快速的。
- 对于给定的 $V(\varphi_1)$ 和 $V(\varphi_2)$ 值，计算
$$Subst_y(V(\varphi_1), V(\varphi_2))$$
很快。

我们解决这个问题的一种方法是：考虑一个非交换环上的 2×2 矩阵，这个环是 $R = \mathbb{Z}[X_1, X_2, \ldots]$，它是具有无限多个变量的多项式环。我们首先在公式和项之间建立一种对应关系：

$$\varphi \rightsquigarrow [\varphi] \in M_{2\times 2}(R),$$

分别对应

$$t \rightsquigarrow [t] \in M_{2\times 2}(R),$$

满足唯一编码性质，也就是说某些元素 $A \in M_{2\times 2}(R)$ 最多对应一个良好形成的字符串，这个字符串可以是一个公式或一项。

对于这种编码，我们定义符号指纹

$$F(\varphi) = ([\varphi], [\varphi]_{x_1}, \ldots, [\varphi]_{x_k}),$$

以及类似术语的定义。矩阵 $[\varphi]_{x_i}$ 仅用于计算替换，但在应用假言推理步骤时也必须更新它们。请注意，我们使用符号 $F(\varphi)$ 表示多项式上的有限序列矩阵，而 $V(\varphi)$ 是有限域 $\mathbb{F}$ 上的有限序列矩阵。我们通过将多项式变量评估为随机选择的域元素来从 $F(\varphi)$ 获得 $V(\varphi)$。

算术项 $MP(a,b)$ 分别 $Subst_y(a,b)$ 已经在环 $M_{2\times 2}(R)$ 中起作用。

现在，对于一个随机选择的值 $X_1 = r_1, X_2 = r_2, \cdots \in \mathbb{F}_p$，我们评估 2×2 矩阵的条目，并得到定义在 $\mathbb{F}_p$ 上的 2×2 矩阵。总之，模条目的评估，数学证明在一个矩阵序列中的非单射编码由以下给出：

$$\varphi \rightsquigarrow [\varphi] \in M_{2\times 2}(\mathbb{Z}[X_1, X_2, \ldots]) \rightsquigarrow [[\varphi]] \in M_{2\times 2}(\mathbb{F}_p).$$

这导致了以下**原始零知识证明**过程：

- 考虑一个数学证明 P，其结论为 ψ。
- 选择值 $X_1, X_2, \cdots \in \mathbb{F}_p$。
- 使用编码规则计算 $\alpha_1 = [[\psi]]$。
- 对于出现在 P 中的所有公理 a，使用编码规则计算相应的指纹 $V(a)$。
- 对于所有出现在 P 中的公式 φ（不包括公理），分别使用同态性质 $MP(a,b)$ 和 $Subst_y(a,b)$ 计算 $V(\varphi)$，以适当选择 y, a, b。
- 最后这些计算产生了 $\alpha_2 = [[\psi]]$。观察到计算 α_2 的方法与计算 α_1 的方法不同。虽然 α_1 是通过直接编码 ψ 来计算的，但 α_2 则是基于公理并通过证明步骤的同态性质开始进行计算的。
- 检查是否 $\alpha_1 = \alpha_2$ 并接受如果是真，否则拒绝。

此**零知识程序**归功于 Schwartz[4] 和 Zippel[6] 定理：

定理 2.1. 令 $\mathbb{F}$ 是一个有限域，令 $f \in \mathbb{F}[x_1, \ldots, x_n]$ 是一个次数为 $d \geq 0$ 的非零多项式。如果 $r_1, r_2, \ldots, r_n$ 是随机选择的，并且在 $\mathbb{F}$ 中的选择是独立的，则：

$$Pr[f(r_1, r_2, \ldots, r_n) = 0] \leq \frac{d}{|\mathbb{F}|}.$$

Schwartz 和 Zippel 定理指出，对于一个非恒等零的多项式，在大有限域中随机评估为零的概率是相对较小的。令 φ 为数学证明的结论。设 $F_1 \in M_{2\times 2}(\mathbb{Z}[X_1, X_2, \ldots])$ 是从 φ 通过直接编码获得的多项式矩阵。令 $F_2 \in M_{2\times 2}(\mathbb{Z}[X_1, X_2, \ldots])$ 是由公理的编码以及使用证明步骤及其同态性质得到的多项式矩阵。我们希望以高度置信度证明 $F_1 = F_2$ 作为多元多项式。为了这个目的，我们只需计算它们的评估值 α_1 和 α_2，其中 α_2 是通过公理的评估和使用同态性质来计算的。如果域足够大，等式 $\alpha_1 = \alpha_2$ 意味着，以高概率而言，$F_1 = F_2$，因此证明得到的公式确实与所声称的结论相同，后者已经被直接编码。

这种原始的零知识程序有一个缺点，即其长度等于证明的长度。该程序必须与算术电路的零知识证明方法、折叠方法等相结合，以生成恒定长度的 ZK 证书。此外，整个过程必须通过应用 Fiat-Shamir 启发法，[3]，转化为非交互式的。

3 多元多项式的矩阵

最初导致这一小节的观察结果是，由不同变量组成的矩阵：

$$A(k) = \begin{pmatrix} x_{4k+1} & x_{4k+2} \\ x_{4k+3} & x_{4k+4} \end{pmatrix}$$

是非交换性的，以至于如果两个乘积相等：

$$A(i_1)\dots A(i_n) = A(j_1)\dots A(j_m)$$

那么 $n = m$ 和 $i_1 = j_1$，…，$i_n = j_n$。这意味着这样的单项式（基本矩阵的乘积）包含了关于因子数量、顺序及其身份的信息。这三个信息元素对于编码树中的路径是必不可少的。

然而，继续使用这样的矩阵会很昂贵，因为需要选择四个域元素来评估每一个基本矩阵，并且为了保持作为指纹的一部分，每个矩阵都需要保留四个元素。相反，我们提出了一种基本矩阵系统，它实现了相同的目标，但只需要评估一个域元素，并且作为指纹一部分的每个矩阵只需三个域元素。

定义 3.1. 令 x_1 为一个变量，即某个在 $\mathbb{Q}$ 上超越的元素。设：

$$A(x_1) = \begin{pmatrix} x_1 & 1 \\ 0 & 1 \end{pmatrix}$$

我们考虑 $A(x_1) \in M_{2\times 2}(\mathbb{Z}[x_1, x_2, \dots])$。

我们现在展示这些初等矩阵满足相同的性质：一个单项式包含了足够的信息以唯一确定初等矩阵的数量、它们的顺序及其身份。

引理 3.1. 考虑一组不同的变量 $V = \{x_1, x_2, \dots, x_k\}$。假设 $0 \le i_1, \dots, i_n, j_1, \dots, j_m \le k$。如果：

$$A(x_{i_1})A(x_{i_2})\dots A(x_{i_n}) = A(x_{j_1})A(x_{j_2})\dots A(x_{j_m}),$$

那么以下等式成立：$n = m, i_1 = j_1, \dots, i_n = j_n$。

Demonstraţie. 如果在这个恒等式中，我们设 $x_1 = \dots = x_k = 1$，我们可以观察到：

$$\begin{pmatrix} 1 & n-1 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} 1 & n \\ 0 & 1 \end{pmatrix},$$

从而得到 $n = m$。让我们用 $S(n)$ 表示以下陈述：如果

$$A(x_{i_1})A(x_{i_2})\dots A(x_{i_n}) = A(x_{j_1})A(x_{j_2})\dots A(x_{j_n})$$

那么 $i_1 = j_1$，…，$i_n = j_n$。我们可以观察到通过识别项 $S(1)$ 是显而易见的。假设我们已经证明了 $S(n)$。我们考察假设 $S(n+1)$：

$$A(x_{i_1})A(x_{i_2})\dots A(x_{i_{n+1}}) = A(x_{j_1})A(x_{j_2})\dots A(x_{j_{n+1}}).$$

我们通过归纳观察到：

$$A(x_{i_1})A(x_{i_2})\dots A(x_{i_n}) = \begin{pmatrix} x_{i_1}x_{i_2}\dots x_{i_n} & P(x_{i_1},x_{i_2},\dots,x_{i_{n-1}}) \\ 0 & 1 \end{pmatrix},$$

其中 $P(z_1,\dots,z_{n-1}) \in \mathbb{Z}[z_1,\dots,z_{n-1}]$ 是一个固定的多项式，具有如下性质：没有变量 z_i 能整除 P。我们将归纳假设写成形式：

$$\begin{pmatrix} x_{i_1} & 1 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} x_{i_2}x_{i_3}\dots x_{i_{n+1}} & P(x_{i_2},x_{i_3},\dots,x_{i_n}) \\ 0 & 1 \end{pmatrix} =$$

$$= \begin{pmatrix} x_{j_1} & 1 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} x_{j_2}x_{j_3}\dots x_{j_{n+1}} & P(x_{j_2},x_{j_3},\dots,x_{j_n}) \\ 0 & 1 \end{pmatrix},$$

因此：

$$\begin{pmatrix} x_{i_1}x_{i_2}x_{i_3}\dots x_{i_{n+1}} & 1+x_{i_1}P(x_{i_2},x_{i_3},\dots,x_{i_n}) \\ 0 & 1 \end{pmatrix} =$$

$$= \begin{pmatrix} x_{j_1}x_{j_2}x_{j_3}\dots x_{j_{n+1}} & 1+x_{j_1}P(x_{j_2},x_{j_3},\dots,x_{j_n}) \\ 0 & 1 \end{pmatrix}.$$

我们识别出相应的项：

$$\begin{aligned} x_{i_1}x_{i_2}x_{i_3}\dots x_{i_{n+1}} &= x_{j_1}x_{j_2}x_{j_3}\dots x_{j_{n+1}}, \\ 1+x_{i_1}P(x_{i_2},x_{i_3},\dots,x_{i_n}) &= 1+x_{j_1}P(x_{j_2},x_{j_3},\dots,x_{j_n}). \end{aligned}$$

我们应用没有变量 z_i 整除 P 的性质。通过变量识别我们得到 $x_{i_1} = x_{j_1}$。我们将假设从左侧与 $A(x_{i_1})^{-1}$ 相乘。我们得到了一个 $S(n)$ 的实例，并应用归纳假设。当然，$A(x_{i_1})^{-1}$ 不属于多项式上的矩阵，而是属于有理函数上的矩阵。重要的是可以简化为 $A(x_i)$。 □

4 非交换树中的边变量

本节的目标是展示如何利用之前的构造方法进一步将矩阵与一棵树表示的公式关联起来。

如下，**边变量**表示一个形状为：

$$X_i = \begin{pmatrix} x_i & 1 \\ 0 & 1 \end{pmatrix}.$$

的初等矩阵。我们将给树的每条边和每个顶点关联这样一个矩阵。

为了用树表示公式，逻辑运算和构建项的操作都表示为树的顶点。对于每个特定符号 c，其元数为 $d = d(c)$，都会关联到 $d+1$ 个不同的边变量 $C, C_1, \dots, C_d$。

假设树 T 的根为 c，与 c 相连的子树是 $T_1,\dots,T_d$。假设已经将矩阵

$$[T_1],\dots,[T_d] \in M_{2\times 2}(\mathbb{Z}[X_1,X_2,\dots])$$

与这些子树关联起来。然后我们将以下配对与 T 关联：

$$[T] = A(C) + A(C_1)[T_1] + \dots + A(C_d)[T_d].$$

定义 4.1. 如果 φ 是一个公式或项，则令 $[\varphi]$ 表示与其树关联的多项式矩阵。

定理 4.1. 一个矩阵表示最多一个公式。

证明: 我们通过一个简化的示例来说明这一点。考虑以下归纳定义：

1. 公式 x,y,z 是原子命题公式。
2. 如果 φ 和 ψ 是公式，那么：

$$\neg\varphi,\ \varphi\to\psi,$$

是公式。

字母表是 $A=\{x,y,z,\neg,\to\}$。

变量 x,y,z 是元数为 0 的符号，并且始终是终端节点。我们将其与矩阵关联：

$$[x]=A(X)=\begin{pmatrix}X & 1\\ 0 & 1\end{pmatrix},\ [y]=A(Y)=\begin{pmatrix}Y & 1\\ 0 & 1\end{pmatrix},\ [z]=A(Z)=\begin{pmatrix}Z & 1\\ 0 & 1\end{pmatrix}.$$

具有正元数的符号是 $\{\neg,\to\}$。我们与 $\neg$ 关联矩阵：

$$A(N)=\begin{pmatrix}N & 1\\ 0 & 1\end{pmatrix},\ A(N_1)=\begin{pmatrix}N_1 & 1\\ 0 & 1\end{pmatrix}.$$

我们与 $\to$ 关联矩阵：

$$A(I)=\begin{pmatrix}I & 1\\ 0 & 1\end{pmatrix},\ A(I_1)=\begin{pmatrix}I_1 & 1\\ 0 & 1\end{pmatrix},\ A(I_2)=\begin{pmatrix}I_2 & 1\\ 0 & 1\end{pmatrix}.$$

变量 $7X,Y,Z,N,N_1,I,I_1,I_2$ 两两不同。

归纳步骤由以下给出：

$$[\neg\alpha]=A(N)+A(N_1)[\alpha],$$

$$[\alpha\to\beta]=A(I)+A(I_1)[\alpha]+A(I_2)[\beta],$$

定理的陈述是通过对公式构建规则进行归纳证明的。我们真正证明的是等价语句：每个公式仅由一个多项式矩阵编码。如果 φ 是一个原子命题符号，那么 $[\varphi]$ 是 $[x]$，$[y]$ 或 $[z]$，因此从 $[\varphi]=[\varphi']$ 可以立即得出 $\varphi=\varphi'$。假设 $\varphi=\neg\alpha$ 成立。那么：

$$[\varphi]=A(N)+A(N_1)[\alpha].$$

我们观察到 $A(N)$ 是这里唯一一个一次单项式。因此可以得出结论，我们在读取一个否定表达。所有其他单项式都以 $A(N_1)$ 开头，因为正如引理 3.1 所示，所有这些非交换单项式只能以 $A(N_1)$ 开头。现在，由归纳假设可知公式 α 唯一地被 $[\alpha]$ 编码，并且可以推断出 φ 唯一地被 $[\varphi]$ 编码。

现在我们考虑情况 $\varphi=\alpha\to\beta$。我们已经看到：

$$[\varphi]=A(I)+A(I_1)[\alpha]+A(I_2)[\beta].$$

再次，$A(I)$ 是唯一的单变量一次项，并且它的存在表明我们在读取一个蕴含关系。所有其他单项式具有形如 $A(I_1)B$ 或 $A(I_2)B$ 的形式。由初等矩阵乘积的唯一性（引理 3.1），这些单项式只能以 $A(I_1)$ 或 $A(I_2)$ 开头。通过提取公因数，我们得到表达式 $A(I_1)[\alpha]+A(I_2)[\beta]$。根据归纳假设，公式 α 和 β 分别由多项式矩阵 $[\alpha]$，即 $[\beta]$ 唯一表示，因此 φ 由多项式矩阵 $[\varphi]$ 唯一表示。 □

5 同态性质

在本小节中，我们定义了公式指纹的概念，并展示了这一概念具有证明中使用的操作的同态性质：分离规则和代换。我们展示如何通过分离规则操作的参数的指纹来计算由分离规则生成的公式的指纹。此外，我们还展示了如何通过被代换变量所在公式的指纹以及代换了该变量的公式（或项）的指纹来计算通过代换获得的公式的指纹。

假设证明中的三行不同内容为：

$$\varphi$$
$$\varphi \to \psi$$
$$---$$
$$\psi$$

使得公式 ψ 由公式 φ 和 $\varphi \to \psi$ 通过分离规则得出。假设我们为蕴含符号 $\to$ 配备了三个矩阵 $A(I)$、$A(I_1)$ 和 $A(I_2)$，使得：

$$[\varphi \to \psi] = A(I) + A(I_1)[\varphi] + A(I_2)[\psi].$$

然后可以按如下方式计算结论：

$$[\psi] = A(I_2)^{-1}\left([\varphi \to \psi] - A(I) - A(I_1)[\varphi]\right).$$

替换也享有同态性质。假设有一个公式 $\varphi(x)$ 并用一棵树 $[\psi]$ 替换 x，这棵树对应于一个公式或一个项。我们观察到：

$$[\varphi(x)] = \sum_{\text{nodes } c} A(X_{i_1})\ldots A(X_{i_n})\cdot A(X_c).$$

这里对于每一个节点 c，单项式 $A(X_{i_1})\ldots A(X_{i_n})$ 由从根到节点 c 的路径上的边变量组成。如果两个这样的节点被标记为 x 并且要进行替换，则有：

$$[\varphi(x/\psi)] = [\varphi(x)] - A(X_{i_1})\ldots A(X_{i_n})[x] - A(X_{j_1})\ldots A(X_{j_m})[x] +$$

$$+A(X_{i_1})\ldots A(X_{i_n})[\psi] + A(X_{j_1})\ldots A(X_{j_m})[\psi].$$

一般来说，令 $x \in A$ 是一个变量，并且令 X 与这个元数为 0 的符号相关联的多项式变量。令 φ 是一个公式或项。我们表示为：

$$\sum_{c=x} A(X_{i_1})\ldots A(X_{i_n})\cdot A(X_c) := [\varphi]_x \cdot A(X_c).$$

由此可知，对于每一个公式或项 ψ，

$$[\varphi(x/\psi)] = [\varphi] - [\varphi]_x \cdot A(X) + [\varphi]_x \cdot [\psi].$$

注意矩阵 $[\varphi]_x$ 在之前的公式中已被隐式定义。

定义 5.1. 令 φ 是在 A 上的一个良好形成的表达式，即一个项或一个公式。假设 $x_1,\ldots,x_k$ 是 φ 中的自由变量。我们将 φ 的指纹称为元组：

$$([\varphi],[\varphi]_{x_1},\ldots,[\varphi]_{x_k}).$$

我们用 $F(\varphi)$ 表示 φ 的指纹。

注意，正如我们在第 2 节中宣布的那样，我们将处理两种类型的指纹。对于一个可能是公式或项的字符串 φ，我们已经定义了一个由多项式环 $\mathbb{Z}[X_1,X_2,\ldots]$ 上的矩阵向量组成的指纹。一旦固定了变量的有限域元素，比如说 $X_1 = r_1,\ldots,X_k = r_k$，这个指纹就可以在这些值上进行计算，并且变成一个在有限域上的矩阵向量。由于下面展示的同态性质，多项式指纹之间的代数关系与已计算的指纹之间的代数关系是相同的。

在接下来的两个定理中，我们展示了可以通过输入的指纹计算出分离规则和替换结果的指纹。证明只是简单的计算，在这里省略了它们。

定理 5.1. 假设公式 φ 和 $\varphi \to \psi$ 的指纹为：

$$F(\varphi) = ([\varphi],[\varphi]_{x_1},\ldots,[\varphi]_{x_k}),$$

$$F(\varphi \to \psi) = ([\varphi \to \psi],[\varphi \to \psi]_{x_1},\ldots,[\varphi \to \psi]_{x_k}).$$

那么 ψ 的指纹是：

$$F(\psi) = ([\psi],[\psi]_{x_1},\ldots,[\psi]_{x_k}),$$

其中：

$$[\psi] = A(I_2)^{-1}\left([\varphi \to \psi] - A(I) - A(I_1)[\varphi]\right),$$

$$[\psi]_{x_i} = A(I_2)^{-1}\left([\varphi \to \psi]_{x_i} - A(I_1)[\varphi]_{x_i}\right).$$

定理 5.2. 令 φ 和 ψ 为公式或项。假设他们的指纹是：

$$F(\varphi) = ([\varphi],[\varphi]_{x_1},\ldots,[\varphi]_{x_k}),$$

$$F(\psi) = ([\psi],[\psi]_{x_1},\ldots,[\psi]_{x_k}).$$

让 $\varphi(x_i/\psi)$ 是用 x_i 替换为 ψ 的结果，并且让 X_i 是一个多项式变量，使得 $A(X_i)$ 与 x_i-节点相关联。然后表达式为：

$$F(\varphi(x_i/\psi)) = ([\varphi(x_i/\psi)],[\varphi(x_i/\psi)]_{x_1},\ldots,[\varphi(x_i/\psi)]_{x_k})$$

其中

$$[\varphi(x_i/\psi)] = [\varphi] - [\varphi]_{x_i}\cdot A(X_i) + [\varphi]_{x_i}\cdot[\psi],$$

并且，如果 $j \neq i$，则：

$$[\varphi(x_i/\psi)]_{x_j} = [\varphi]_{x_j} + [\varphi]_{x_i}[\psi]_{x_j}$$

而如果 $j = i$，则：

$$[\varphi(x_i/\psi)]_{x_i} = [\varphi]_{x_i}[\psi]_{x_i}$$

示例指纹: 考虑该公式

$$\varphi = (x \to y) \to (x \to z).$$

根据我们的定义，有：

$$[\varphi] = [(x \to y) \to (x \to z)] = A(I) + A(I_1)[x \to y] + A(I_2)[x \to z] =$$

$$= A(I) + A(I_1)(A(I) + A(I_1)A(X) + A(I_2)A(Y)) +$$

$$+ A(I_2)(A(I) + A(I_1)A(X) + A(I_2)A(Z)) =$$

$$= A(I) + A(I_1)A(I) + A(I_1)^2A(X) +$$

$$+ A(I_1)A(I_2)A(Y) + A(I_2)A(I) + A(I_2)A(I_1)A(X) + A(I_2)^2A(Z).$$

此外，还有：

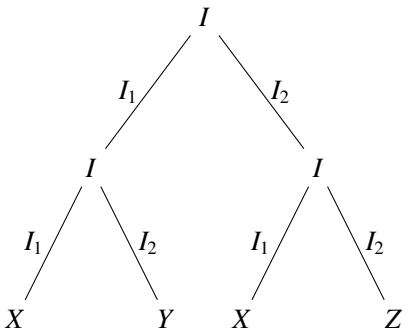
$$[\varphi]_x = A(I_1)^2 + A(I_2)A(I_1),$$

$$[\varphi]_y = A(I_1)A(I_2),$$

$$[\varphi]_z = A(I_2)^2.$$

最后，φ 的指纹是：

$$F(\varphi) = ([\varphi], [\varphi]_x, [\varphi]_y, [\varphi]_z).$$



6 形式化证明的示例

在本节中，我们将介绍结晶一个包含指纹的块模型证明的努力。我们考虑以下命题逻辑框架。有三个公理如下：

$$K(\alpha, \beta): \ \alpha \to (\beta \to \alpha),$$

$$S(\alpha, \beta, \gamma): \ (\alpha \to (\beta \to \gamma)) \to ((\alpha \to \beta) \to (\alpha \to \gamma)),$$

$$N(\alpha, \beta): \ (\neg\alpha \to \neg\beta) \to (\beta \to \alpha).$$

我们考虑以下定理：

定理 6.1.

$$A \to A.$$

该定理将通过三种版本进行证明。第一种是经典风格。第二种，我们将证明转化为指纹计算。第三种，我们把这些计算表示为块。

经典证明: 考虑公式 $B := A \to A$。通过相应的代换，我们写出：

$$S(A,B,A):\ (A \to (B \to A)) \to ((A \to B) \to (A \to A)),$$

$$K(A,B):\ A \to (B \to A),$$

$$K(A,A):\ A \to (A \to A),$$

此时我们观察到 $K(A,A)$ 实际上是：

$$K(A,A):\ A \to B,$$

$$MP(K(A,B),S(A,B,A)) = C:\ (A \to B) \to (A \to A),$$

$$MP(K(A,A),C):\ A \to A.$$

我们也注意到结论与 B 相同。 □

指纹翻译: 我们考虑数值变量 a,i,i_1,i_2。在计算中，它们将被域元素的固定选择所替代。相应的矩阵为：

$$A = \begin{pmatrix} a & 1 \\ 0 & 1 \end{pmatrix},\ I = \begin{pmatrix} i & 1 \\ 0 & 1 \end{pmatrix},\ I_1 = \begin{pmatrix} i_1 & 1 \\ 0 & 1 \end{pmatrix},\ I_2 = \begin{pmatrix} i_2 & 1 \\ 0 & 1 \end{pmatrix},$$

$$I_2^{-1} = \begin{pmatrix} i_2^{-1} & -i_2^{-1} \\ 0 & 1 \end{pmatrix},$$

进一步定义：

$$[A] = A,$$

$$[B] = I + I_1[A] + I_2[A],$$

$$[B \to A] = I + I_1[B] + I_2[A],$$

$$[K(A,B)] = I + I_1[A] + I_2[B \to A],$$

$$[K(A,A)] = I + I_1[A] + I_2[B],$$

$$[C] = I + I_1[K(A,A)] + I_2[B],$$

$$[S(A,B,A)] = I + I_1[K(A,B)] + I_2[C],$$

$$I_2^{-1}([S(A,B,A)] - I - I_1[K(A,B)]) = [C],$$

$$I_2^{-1}([C] - I - I_1[K(A,A)]) = [B].$$

□

致谢

作者与 Grigore Rou、Xiaohong Chen、Brandon Moore、Miruna Roca 和 Traian erbnu 就这一主题进行了许多讨论。本研究得到了罗马尼亚布加勒斯特逻辑与数据科学研究所的 PiSquared 部分支持。

Bibliografie

[1] **Manuel Blum** *How to Prove a Theorem So No One Else Can Claim It*, Proceedings of the International Congress of Mathematicians, Berkeley, California, USA, 1986.

[2] **Marta Bellés-Muñoz, Miguel Isabel, Jose Luis Muñoz-Tapia, Albert Rubio, Jordi Baylina** *Circom: A Circuit Description Language for Building Zero-Knowledge Applications*, IEEE Transactions on Dependable and Secure Computing, Volume 20, Issue: 6, Nov.-Dec., 2022. DOI: 10.1109/TDSC.2022.3232813

[3] **Amos Fiat, Adi Shamir** *How to Prove Yourself: Practical Solutions to Identification and Signature Problems* Advances in Cryptology — CRYPTO' 86. Lecture Notes in Computer Science. Vol. 263. Springer Berlin Heidelberg. pp. 186 – 194, 1987 DOI:10.1007/3-540-47721-7_12 ISBN 978-3-540-18047-0

[4] **Jacob T. Schwartz** *Fast probabilistic algorithms for verification of polynomial identities*, Journal of the ACM. 27 (4): 701 – 717. 1980 DOI: 10.1145/322217.322225

[5] **PiSquared** *The White Paper*, https://pi2.network/papers/pi2-whitepaper, 2025.

[6] **Richard Zippel** *Probabilistic algorithms for sparse polynomials*, In Ng, Edward W. (ed.) Symbolic and Algebraic Computation, EUROSAM '79, An International Symposiumon Symbolic and Algebraic Computation, Marseille, France, June 1979, Lecture Notes in Computer Science. Vol. 72. Springer. pp. 216 – 226. DOI:10.1007/3-540-09519-5_73 ISBN 978-3-540-09519-4.