arXiv:2603.07589v1 [cs.CC] 8 Mar 2026

有界个体次数稀疏多项式的因式分解

Aminadav Chuyoon*
chuyoon1@mail.tau.ac.il

Amir Shpilka*
shpilka@tauex.tau.ac.il

摘要

我们研究个体次数有界的稀疏多项式及其因子的类。具体而言，我们得到了以下算法与结构性结果：

(1) 一种确定性多项式时间算法，用于找出个体次数有界的稀疏多项式的所有稀疏因子。作为其中一部分，我们首次给出了这类多项式的非单项式不可约因子数量的上界。(2) 一种多项式$(n, s^{d \log \ell})$时间算法，可从其乘积的黑盒访问中恢复 ℓ不可约 s稀疏的个体次数有界多项式 d（该乘积不一定稀疏）。这部分解决了 [DST24]中提出的一个问题。特别地，当$\ell = O(1)$时，该算法可在多项式时间内运行。(3) 用于从黑盒访问中对个体次数有界的 s稀疏多项式乘积进行因式分解的确定性算法。在特征零或特征足够大的域上，该算法的运行时间为多项式$(n, s^{d^3}$对数$^n)$时间；在任意域上，运行时间为多项式$(n, (d^2)!,\ s^{d^5}$对数$^n)$时间。这改进了 [BSV20], 的算法；该算法的运行时间为多项式$(n, s^{d^7}$对数$^n)$时间，且仅适用于单个个体次数有界的稀疏多项式。对于输入为单个稀疏多项式的情形，我们给出了一种运行时间为多项式$(n, s^{d^2}$对数$^n)$时间的算法。(4) 给定一个乘积的黑盒访问，其中各因子是个体次数有界的稀疏多项式的因子（不一定稀疏或不可约），我们给出一种确定性多项式时间算法，用于找出其中所有不可约的稀疏多二次因子（及其重数）。这推广了 [Vol15] 和 [Vol17]的算法。我们还说明了如何判定这样的乘积是否为完全幂（当其定义在特征为零或足够大的域上时），从而扩展了 [BV25] 的算法。

我们的算法最自然地适用于特征为零或足够大的域。为了处理任意域，我们引入了针对一类多项式的 本原因子 这一概念，该概念本身也可能具有研究价值。借助这一概念，我们能够调整 [BV25]的思想，并从大多数算法中去除对特征的假设。

*特拉维夫大学计算机科学与人工智能学院。本研究由欧盟（ERC，EACTP，101142020）资助。所表达的观点和意见仅代表作者本人，不一定反映欧盟或欧洲研究委员会执行局的观点。欧盟和资助机构均不对此承担责任。

目录

1 引言 **1**

1.1 背景与动机 .11.2 我们的结果 .21.3 相关工作 .51.4 证明概述 .61.4.1 单项次数有界的稀疏多项式及其因子 . . . . . 61.4.2 寻找稀疏多项式乘积的多二次因子 . . . . . 81.4.3 寻找稀疏多项式的稀疏因子 .101.4.4 稀疏多项式的因式分解 .111.5 组织结构.11

2 预备知识 **12**

2.1 反向首一多项式与归一化 .122.2 多项式的因式分解与插值 .132.3 稀疏多项式 .142.4 结式与判别式 .15

3 生成元构造 **16**

4 整除性测试 ting与精确幂测试 **19**

4.1 任意特征结果 .194.1.1 任意特征工具 .194.1.2 任意特征算法 .244.2 零或大特征结果 . 264.2.1 零或大特征工具 .264.2.2 零或大特征算法 .27

5 有理插值定理 296 元算法 317 寻找多二次因子 338 计算稀疏多项式的所有稀疏因子 358.1 单项次数有界的稀疏多项式的因子数量 . . . . . 358.2 寻找单项次数有界的稀疏多项式的稀疏因子 . . 38.3 稀疏不可约多项式乘积的因式分解 .39

7

9 因式分解 (n, s, d)-稀疏多项式 40

9.1 稀疏多项式的因式分解 .419.2 稀疏多项式乘积的因式分解 .42**10 未来工作与开放问题 43**

参考文献 44A 第 4 节某个结果的另一种证明 46

1 引言

1.1 背景与动机

这篇论文的动机来自许多关于多元稀疏多项式的因式分解及其因子的复杂度的开放问题。这些是次数为 $\mathrm{poly}(n)$、变量数为 n 的多项式，其单项式数量也为 $\mathrm{poly}(n)$。具体而言，这是具有多项式规模 $\Sigma\Pi$ 电路的多项式类，从代数复杂性的角度来看，这是最简单的类之一。

尽管结构简单，稀疏多项式却构成了一类丰富且不平凡的多项式，受到了广泛研究。针对这一类多项式，人们开发了高效的确定性多项式恒等式检验和重构算法 [GK87，BOT88，GKS90，CDGK91，KS01]。特别是，Klivans 和 Spielman 的工作 [KS01] 提出了一种特殊的约化方法，可将多元稀疏多项式约化为多项式次数的一元多项式；该方法适用于任意域，并已在许多领域得到应用，主要用于解决多项式恒等式检验和重构方面的问题。相关综述参见 [Sax09，SY10，Sax14，DG24]。

在这项工作中，我们主要关注稀疏多项式的因子。Kaltofen 的开创性工作 [Kal85a] 表明，稀疏多项式的因子具有多项式规模的代数电路。近期， [BKR+25]的一项突破性结果进一步表明，这些因子甚至具有多项式规模的有界深度电路。

然而，一个引人深思且四十多年来仍未解决的问题，关乎稀疏多项式的因子数量相对于其单项式个数的复杂度。换句话说，稀疏多项式的因子最多能有多少个单项式？von zur Gathen 和 Kaltofen 的以下例子表明，多项式的不可约因子的稀疏度可能是超多项式级别的。

例 1.1（例子出自 [VZGK85]）。 考虑

$$g(x) = \prod_{i=1}^{n}(x_i - 1) \quad \textit{and} \quad h(x) = \prod_{i=1}^{n}(1 + x_i + \cdots + x_i^{n-1}) + n.$$

不难证明， h在 $\mathbb{Q}$上不可约。令

$$f(x) = g \cdot h = \prod_{i=1}^{n}(x_i^n - 1) + n\prod_{i=1}^{n}(x_i - 1).$$

那么， f 有 $(2^{n+1} - 1)$ 个单项式，且有一个不可约因子 h，其稀疏度为 n^n，相对于 f的稀疏度而言是超多项式级别的。

这促使他们提出了以下问题。

问题 1.2 （[VZGK85]）。 因式分解问题的输出规模实际上能否超过输入稀疏度的准多项式级别？

这个问题仍然远未解决，目前仅有若干特殊情形得到了解决。在 [Vol15] Volkovich 证明了稀疏多项式的多线性因子是稀疏的。在后续工作 [Vol17]中，他证明了多二次稀疏多项式的所有因子本身都是稀疏的。这一结果由 Bhargava、Saraf 和 Volkovich 推广到了更高的个体次数 [BSV20], 他们考虑了有界个体次数的稀疏多项式 d，并证明了此类多项式的因子的稀疏度具有准多项式上界。特别地，如果 f 是一个 s-稀疏

n-元多项式，其个体次数不超过 d，则 f 的每个因子至多有 $s^{O(d^2 \log^n)}$ 个单项式（见 定理 2.14）。这是在情形 $d > 2$ 下关于因子的首个非平凡稀疏度上界。尽管 [BSV20] 给出了有界个体次数多项式的因子稀疏度的准多项式上界，但人们相信“真正”的上界应当是多项式的。在 [BV25], 中，作者通过为多个问题给出多项式时间算法，为此提供了一些证据；而这些算法的存在性可由关于因子稀疏度的猜想多项式上界推出。

在他们的工作中，von zur Gathen 和 Kaltofen 给出了一种用于稀疏多项式因式分解的随机算法，并指出了实现确定性因式分解算法的另一项障碍：

“还要注意，似乎不存在一种可行的方法来确定性地检查输出是否正确，即，是否 $f = \prod_i f_i^{e_i}$” [VZGK85]。

具体而言，以下问题尚未解决。

问题 1.3 ([DST24，问题 1])。 我们能否求出不可约因子均为稀疏多项式的多项式的因式分解？

事实上，Dutta 等人还提出了问题 1.3的以下特例。

给定一个黑盒，它计算若干有界个体次数的稀疏不可约多项式的乘积 f_i，请在确定性多项式时间内求出这些多项式的 f_i。

上述 [Vol15、Vol17、BSV20] 的结果也对问题 1.3和问题 1.4有重要影响。在 [Vol15] 中，Volkovich 给出了一个确定性多项式时间算法，用于对不可约因子均为多线性（因此也是稀疏）多项式的稀疏多项式进行因式分解。在 [Vol17]中，他给出了一个高效的确定性算法，用于对多二次稀疏多项式进行因式分解。在 [BSV20] 中，作者将他们的稀疏度上界与类似于[Vol15、Vol17] 的技术相结合，从而获得了一种确定性拟多项式时间算法，用于对有界个体次数的稀疏多项式进行因式分解。Dutta、Sinhababu 和 Thierauf [DST24] 给出了高效的确定性重构方法，用于重构稀疏多项式的常数次因子。

BSV [算法的一项显著局限20] 是，即使输入多项式存在稀疏因子，它也没有提供高效的方法来找出这些因子。这个问题与 Dutta、Sinhababu 和 Thierhauf 提出的以下问题密切相关。

问题1.5 ([DST24，问题 2])。 设计一种优于指数时间的确定性算法，输出稀疏多项式的所有不可约因子。

另一个尚未解决的重要问题涉及重构分子和分母均为稀疏多项式的有理函数。令人惊讶的是，目前已知解决这一问题的最佳确定性算法需要指数时间 [GKS94]。虽然看似无关，但这一问题的特殊情形是我们算法的核心；在提供更多信息的情况下，我们解决了其中两个特殊情形。

关于稀疏多项式，还有许多其他未解决问题，读者可参阅[DST24]和 [FS15]。

1.2 我们的结果

在介绍我们的结果之前，先引入以下记号。

定义 1.6。 我们将次数有界为 d 的 n 元、 s 稀疏多项式类记作 $\mathcal{P}(n,s,d)$。我们称该类中的多项式为 (n,s,d) 稀疏多项式。事实证明，在许多情况下，讨论 (n,s,d) 稀疏多项式的因子类是有意义的；我们将这个多项式类记作 $\mathcal{P}_{\text{Factors}}(n,s,d)$。

我们首先对主要贡献作如下非正式总结：

(1) 我们证明了 (n,s,d)-稀疏多项式的不可约因子个数的紧确上界（定理 1.8）。(2) 利用这一上界，我们给出了首个确定性多项式时间算法，可构造所有稀疏因子：(n,s,d)-稀疏多项式的稀疏因子（定理 1.9）。(3) 我们给出了一个确定性黑盒因式分解算法，可在关于项数的准多项式时间内对稀疏 (n,s,d)-稀疏多项式的乘积进行因式分解。当项数为常数时，这解决了问题 1.4（定理 1.10）。值得注意的是，该算法的运行时间关于变量数 n为多项式。

注 1.7. 陈述结果时，我们省略了对域（除特征之外）及位复杂度的精确依赖。与通常的因式分解算法一样，还应乘上一个额外因子，以反映在该域上进行一元因式分解的复杂度。这些上界在第 2.2节中给出。

现在我们正式陈述本文的结果。我们的第一个主要结果给出了 (n,s,d)-稀疏多项式的稀疏因子数量上界。虽然其证明并不困难，但它对我们的算法至关重要，并展示了稀疏性所施加的强结构限制。

定理 1.8 (关于 (n,s,d)-稀疏多项式的因子界）。设 $f \in \mathcal{P}(n,s,d)$。 f 的非单项式不可约因子数量（按重数计）至多为 $d\log s$。特别地，一个个体次数有界为 d 的 s-稀疏多项式 f，其不能被任何单项式整除的因子数量至多为 $2^{d\log s} = s^d$。

我们的第一个主要算法结果展示了如何在多项式时间内确定性地输出这类多项式的所有稀疏因子。此前已知的只有拟多项式时间算法[BSV20]。

定理 1.9 (求出 (n,s,d)-稀疏多项式的所有稀疏因子）。存在一个确定性多项式$(n,d!\ s^d)$-时间算法，给定 $f \in \mathcal{P}(n,s,d)$，输出 s-稀疏多项式的所有稀疏因子（这些因子不能被任何单项式整除），以及整除 f 的最高次数单项式。

我们的第二个主要算法结果回答了 问题 1.4 在乘积中只有常数多个多项式时的情形；而在一般情形下，它给出一个拟多项式时间算法。

定理 1.10 (对 (n,s,d)-稀疏多项式的乘积进行因式分解）。设 $\mathbb{F}$ 是特征为零或大于 $2d$ 的域。存在一个确定性多项式$(n,d^d,s^{d\log \ell},\ell^d)$-时间算法，给定乘积 $f = \prod_{i=1}^{\ell}\phi_i$ 的黑盒访问，其中 ϕ_1、……、 $\phi_\ell \in \mathcal{P}(n,s,d)$ 均为不可约多项式（不一定互不相同），返回 ϕ_i-稀疏因子。此外，若去掉对域的假设，则存在一个算法，可在多项式$(n,(d^2)!\ s^{d\log \ell + d^3},\ \ell^d)$-时间内解决此问题。

本定理将作为下一个更一般结果的推论而得出。

定理 1.11 （稀疏多项式因子的乘积的因式分解）。 设 $\mathbb{F}$ 是一个域，其特征$(\mathbb{F}) = 0$或特征$(\mathbb{F}) > 2d$。存在一个确定性的多项式$(n, d^d, s^{d \log \ell}, \ell^d)$时间算法：给定对域 $\mathbb{F}$上 $\mathcal{P}_{Factors}(n, s, d)$ 中 ℓ 个多项式的乘积 f 的黑盒访问，该算法返回 f 的所有有界个体次数为 d 、且不被任何单项式整除的 s-稀疏因子，以及每个 x_i 作为 f的因子的重数 k_i 。此外，若 f 实际上是 $\mathcal{P}(n, s, d)$中多项式的乘积，则存在一个适用于任意域的此类算法，其运行时间为多项式$(n, (d^2)!, s^{d \log \ell + d^3}, \ell^d)$。

请注意，对于小特征的域，这一结果略弱。事实上，在第 4 节中我们将看到，我们的一些方法仅适用于零特征域，或特征足够大的域。

除了主要结果之外，我们还对先前的工作进行了若干改进和推广。我们首先陈述有关将有界个体次数的稀疏多项式分解为不可约因子的结果。第一个结果改进了目前已知的此类多项式的最佳因式分解算法 [BSV20]。

定理 1.12. 存在一个确定性的多项式$(n, s^{d^2 \text{对数} n})$-时间算法：给定一个 s-稀疏、个体次数有界的多项式 d，它定义在 n 个变量上，该算法返回其不可约因式分解。

第二个定理将 [BSV20] 的结果推广到一个显著更大的多项式类，同时与其算法相比，实现了更短的运行时间。

定理 1.13. 设 $\mathbb{F}$ 是一个特征为$\text{char}(\mathbb{F}) = 0$ 或$\text{char}(\mathbb{F}) > 2d$的域。存在一个确定性的多项式$(n, s^{d^3 \text{对数} n}, (d\ell)^d)$-时间算法：给定对 $\mathbb{F}$ 上 $\mathcal{P}_{Factors}(n, s, d)$ 中 ℓ 个多项式的乘积 f 的黑盒访问，该算法返回其不可约因式分解。此外，还存在一个适用于任意域的算法，其运行时间为多项式$(n, (d^2)!, s^{d^5 \text{对数} n}, \ell^d)$。

注意，这最后一个结果考虑的是单项次数有界的稀疏多项式的因子这一类。事实上，我们将看到，在许多情形下考虑这一类更为自然，而且我们的若干结果实际上对这一类成立，而不仅仅是对稀疏多项式这一类成立。

最后，我们推广了 [Vol15、Vol17、BV25] 的结果，其方式类似于我们对 [BSV20]的结果所做的推广。

我们的下一个算法找出来自 $\mathcal{P}_{\text{Factors}}(n, s, d)$的多项式乘积的所有多二次不可约因子。此前，仅知道不可约因子全部为多二次且 s-稀疏的多项式的因式分解，见 [Vol17，定理 47]。

定理 1.14. 设 $\mathbb{F}$ 是特征零或大于 2d的域。存在一个确定性$\text{poly}(n, d!, s^d, \ell)$时间算法：给定对乘积 f 的黑盒访问，该乘积由 ℓ 个属于 $\mathcal{P}_{Factors}(n, s, d)$、定义在 $\mathbb{F}$上的元素构成，并返回所有 s-稀疏多二次不可约因子，即 f的因子。特别地，该算法返回 f 的所有多线性因子（因为它们是 s-稀疏的，由引理 *2.12*可知）。此外，存在适用于任意域的此类算法，其运行时间为$\text{poly}(n, (d^2)!, s^{d^2}, \ell)$。

我们对 [BV25] 的完全幂测试也给出了类似的改进。

定理 1.15。 设 $\mathbb{F}$ 是一个特征为零或大于 2d的域。存在一个确定性多项式$(n, d!, s^d, \ell)$-时间算法，给定多项式 f 的黑盒访问权限（该多项式是 $\mathcal{P}_{Factors}(n, s, d)$ 上 $\mathbb{F}$ 定义的 ℓ 个多项式的乘积）以及一个正整数 e，判断 f 是否为一个完全 e 次幂。

最后，我们扩展了 [BV25] 的多项式恒等式检验结果。

定理 1.16。 设 $\mathbb{F}$ 是一个特征为零或大于 $2d$ 的域。存在一个确定性多项式$(n, d!, s^d, \ell)$-时间算法，给定两个多项式（在 $\mathbb{F}$ 上） f 和 g 的黑盒访问权限（它们分别是 $\mathcal{P}_{Factors}(n, s, d)$ 中 ℓ 个多项式的乘积），判断 $f|g$ 是否成立。此外，如果 f 和 g 实际上是 $\mathcal{P}(n, s, d)$ 中 ℓ 个多项式的乘积，则存在一个可在任意域上运行的算法，其运行时间为多项式$(n, (d^2)!, s^{d^3}, \ell)$

如第 1.1 节中所述，我们需要解决一些有理插值问题。有些出人意料的是，目前尚不知道如何在多项式时间内，从其商 s-稀疏多项式 f 和 g 的黑盒访问中，确定性地重构两个互素的 g/f 。我们将看到，利用关于 f的一些额外信息，实际上可以做到这一点。在没有这些额外信息的情况下，目前已知解决该问题的只有指数时间确定性算法 [GKS94]。值得注意的是，已知解决该问题的高效随机化算法，例如参见 [CL11、vdHL21]。

1.3 相关工作

近年来，本文所研究的问题得到了广泛研究。我们围绕三个对本文至关重要的主题，组织介绍已有结果：稀疏度界、因式分解与重构算法，以及整除性与恒等性测试。

稀疏多项式的因子稀疏度。Volkovich [Vol15] 证明了，s-稀疏多项式的多线性因子仍然是 s-稀疏的。在 [Vol17] 中，他证明了 s-稀疏多项式的因子同样是 s-稀疏的，其中该多项式为多线性二次多项式。对于个体次数有界的一般多项式d，Bhargava、Saraf 和 Volkovich [BSV20], 首次对具有有界个体次数 d 的 s-稀疏多项式的因子稀疏度获得了非平凡上界，即 $s^{O(d^2 \log n)}$。Bisht 和 Saxena [BS25] 将这一上界改进为 $s^{\text{poly}(d)}$ ，适用于个体次数有界的对称 s-稀疏多项式类 d。Bisht 和 Volkovich [BV25] 获得了一个 $s^{d \log s}$ 上界，该上界针对一个具有有界个体次数 d 的 s-稀疏多项式与一个多线性因子之商的稀疏度。最后，Forbes [For15] 的工作以及 [BV25]中的结果表明，$f \in \mathcal{P}(n, s, d)$ 与一个次数为 r 的多项式之商的稀疏度有上界 $(ns)^{O(dr)}$。

确定性因式分解与重构。 因式分解方面的研究进展，大体上与稀疏度界的发展同步。Volkovich 在 [Vol15] 中展示了如何高效地对因子均为多线性多项式的稀疏多项式进行因式分解。在 [Vol17] 中，他展示了如何在给定黑盒访问的情况下，高效计算多项式的不可约因子，前提是这些因子均为多二次且稀疏的。在定理 1.14 中，我们进一步改进了这一结果，放宽了对输入多项式的条件——允许它是某个个体次数有界的稀疏多项式的若干因子的乘积，而不只是稀疏多二次多项式的乘积。

利用其稀疏度界，Bhargava、Saraf 和 Volkovich [BSV20] 提出了一种用于对个体次数有界的稀疏多项式进行因式分解的确定性拟多项式算法。然而，他们的方法无法得到一个多项式时间算法来寻找输入多项式的稀疏因子，因此也无法推出定理 1.9。此外，他们的算法无法推广到对稀疏有界次数多项式的乘积进行因式分解，更不用说对这类多项式的因子的乘积进行因式分解了。

Kumar、Ramanathan 和 Saptharishi [KRS24] 以及 Dutta、Sinhababu 和 Thierauf [DST24],提出了一种算法，能够在准多项式时间内输出稀疏多项式的所有常数次数不可约因子。由于他们的结果适用于个体次数不受限制的一般稀疏多项式，因此不能由本文的方法推出。

在 [BV25], 中，给出了一种算法，用于判断有界个体次数的稀疏多项式是否为完全幂；在 第 4.2 节 中，当输入多项式是 $\mathcal{P}_{\text{Factors}}(n, s, d)$ 中多项式的乘积，且我们可以对其进行黑盒访问时，我们运用更强的方法得到了这一结果。然而，当底层域具有小特征时，这种推广并不成立。

整除性与恒等性测试。在 [Vol17] 中还展示了如何高效检验两个 s个体次数有界的稀疏多项式 d 是否互相整除；这一结果在定理 1.16中得到改进：我们针对两个由个体次数有界的稀疏多项式的因子构成的乘积，在可对其进行黑盒访问的条件下，给出了整除性测试结果。在同一项工作中，Volkovich 还给出了一个确定性算法，用于测试个体次数为常数的稀疏多项式能否分解为不可约稀疏因子。也就是说，检验 $f = \prod g_i$ 是否成立，其中 f 的个体次数为常数，且 g_i均为不可约稀疏多项式。我们注意到，根据推论 8.6，我们可以高效地找出 g_i，并验证 $f = \prod g_i$。

Forbes [For15] 给出了一个拟多项式时间算法，用于检验常数次数多项式是否整除稀疏多项式。请注意，虽然我们的结果允许作为除数的多项式复杂得多，但 Forbes 的结果对给定稀疏多项式的个体次数没有任何限制。因此，我们的方法不足以证明他的结果，也无法证明任何其他关于个体次数不受限制的稀疏多项式的结果。

在 [BV25], 中，作者为类 $\Sigma^{[2]}\Pi\Sigma\Pi^{[\text{ind-deg } d]}$设计了一种高效的黑盒多项式恒等式测试算法；在此过程中，他们还得到了关于类 $\mathcal{P}_{\text{Factors}}(n, s, d)$的一些技术结果。在第 4.1 节中，我们将说明，他们的方法蕴含了一个比该多项式恒等式测试更强的结果。为这一类多项式进行恒等式测试，实质上等价于给出一种算法：在仅能黑盒访问的条件下，判断两个 f、 g 由 (n, s, d)-稀疏多项式构成的乘积是否相等。我们将证明， [BV25] 的方法可用于判断 f 是否整除 g。在第 4.2 节 中，我们将进一步推广并加强 [BV25], 的技术结果，但仅适用于特征为零或足够大的域。

1.4 证明概述

我们首先介绍一些关于单项次数有界的稀疏多项式类的技术结果，这些结果是我们的证明所需要的。随后，我们概述关于稀疏多项式因式分解的研究结果。首先讨论我们关于恢复由 (n, s, d)-稀疏多项式构成的乘积的所有不可约稀疏多二次因子的结果，因为其证明包含了我们主要结果的证明中用到的一些重要思路。接着，我们将介绍如何恢复单项次数有界的稀疏多项式的所有稀疏因子，以及我们对问题 1.4的部分解答。最后，我们简要说明如何利用我们的方法对一般的 (n, s, d)-稀疏多项式及其乘积进行因式分解，并用到 [BSV20]的界。

1.4.1 有界个别次数稀疏多项式及其因子

我们用于获得这些结果的主要工具是一个生成器 $G : \mathbb{F}^6 \to \mathbb{F}^n$ ，它能保留 $\mathcal{P}_{\text{Factors}}(n, s, d)$ 中多项式的重要性质。粗略地说，我们希望该映射 G 满足以下两个性质：

(1) 它的像 $\text{Im}(G)$ 是 $\mathcal{P}(n, s, d)$ 中多项式的一个插值集。

(2) 若 ϕ 和 ψ 是非相伴的[1] $\mathcal{P}_{\text{Factors}}(n, s, d)$中的不可约元素，则 $\phi(G_i)$ 和 $\psi(G_i)$作为 x_i中的多项式互素，其中 G_i 应理解为“恢复第 G个变量”（即保持 i不变，并根据 x_i 对其余变量进行代换）。 G

[1] 如果一个多项式是另一个多项式的标量倍数，则称这两个多项式相伴。

第一项性质可由 Klivans-Speilman 生成器（KS 生成器）的基本性质轻易得到，该生成器由 [KS01] 引入。我们证明，第二项性质可由第一项性质以及如下事实推出：有界个体次数的稀疏多项式的结式仍然是稀疏的。然而，只有当底层域 $\mathbb{F}$的特征为零，或其特征大于个体次数上界的两倍时，我们才能做到这一点。若不满足该条件，我们只能证明一个稍弱的结论——将不可约多项式这一概念替换为我们在本文中首次定义的新概念“本原因子”，则第二项性质成立（下文将介绍这一概念）。

在概述证明 G 的第一条性质蕴含第二条性质之前，我们注意到，证明该结论将自动为稀疏多项式带来一些算法。例如，很容易看出，通过对输入多项式与 G_i的复合进行因式分解，可以从 G 的一般第二条性质得到以下两个结果。

(1) 给定对 $\mathcal{P}_{\text{Factors}}(n,s,d)$中两个元素的乘积的黑盒访问，我们可以在确定性多项式时间内判断其中一个是否整除另一个（定理 1.16）。(2) 给定对 $\mathcal{P}_{\text{Factors}}(n,s,d)$中元素乘积的黑盒访问，我们可以在确定性多项式时间内判断它是否为完全幂（定理 1.15）。

这已经推广了 [Vol17] 和 [BV25]中的结果。由于我们只能对某些域得到 G 的一般第二条性质，这些结果并不适用于任意域。然而事实证明，前述 G 的较弱版本足以在任意域上得到 定理 1.16 ，只需将 $\mathcal{P}_{\text{Factors}}(n,s,d)$ 替换为 $\mathcal{P}(n,s,d)$ （见定理 4.14）。

在本小节余下部分，我们将说明第二个性质的证明，既包括任意域的情形，也包括特征为零或足够大的域的情形。

任意域。在 [BV25 的引理 4.5、4.6] 中证明了，只要每对不可约多项式 ϕ 和 ψ 满足某个非退化条件， G 的第二个性质就成立：存在 (n,s,d)-稀疏多项式 f、g，使得 $\phi|f$、$\psi|g$，且矩阵 $\begin{pmatrix} \nu_\phi(f) & \nu_\phi(g) \\ \nu_\psi(f) & \nu_\psi(g) \end{pmatrix}$ 可逆；其中 $\nu_\phi(f)$ 是 f 作为 ϕ 的因子的重数，其余各项也依此类推。也就是说，如果 ϕ 作为 $\mathcal{P}(n,s,d)$ 中某个元素 f 的因子的重数决定了 ψ 作为 f 的因子的重数，那么多项式对 ϕ、$\psi \in \mathcal{P}_{\text{Factors}}(n,s,d)$ 就不满足该条件。从这个意义上说，从 $\mathcal{P}(n,s,d)$ 中多项式的角度来看， ϕ 和 ψ 是“不可分”的。

对于每个不可约 $\phi \in \mathcal{P}_{\text{Factors}}(n,s,d)$，考虑所有与其不可分的 ψs，即 ϕ,ψ 不满足上述非退化条件。对于每个 $f \in \mathcal{P}(n,s,d)$，考虑这些 ψs 作为 f - $(\nu_\psi(f))_\psi$的因子时的重数向量。根据退化假设，这些向量都彼此成比例。考虑这些向量张成的 $\mathbb{Z}$-空间，并令 $(a_\psi)_\psi$为这个一维格的一个生成器。多项式 $\prod_\psi \psi^{a_\psi}$ 称为类 $\mathcal{P}(n,s,d)$的本原因子。请注意，根据构造，由于“不可分”这一性质在 $\mathcal{P}_{\text{Factors}}(n,s,d)$中的不可约多项式类上构成等价关系，因此任意两个本原因子必定互素。

本原因子的概念是一个一般性的概念，适用于一般的多项式类 $\mathcal{P}$，因此它本身也可能值得独立研究；其形式定义见定义 4.1。多项式类 $\mathcal{P}$ 的本原因子对于该类而言 $\mathcal{P}$ ，就如同不可约多项式对于所有多项式构成的类而言一样，因为该类 $\mathcal{P}$ 中的每个元素都能以唯一的方式写成 $\mathcal{P}$ 的本原因子的乘积；这一点不难从

上一段给出的构造看出，并且在形式上可由引理 4.6证明。因此，讨论本原因子 $\nu_\phi(f)$ 的重数 ϕ作为 $f \in \mathcal{P}$的一个因子是有意义的。

有了这个定义，我们可以证明，任意两个非相伴本原因子 ϕ和 ψ属于某一类的 $\mathcal{P}$ 因子都满足 [BV25] 的非退化条件。因此，我们可以应用 [BV25] 上述结果的证明，从而得到 G 第二个性质的较弱版本，即对于任意两个非相伴本原因子 ϕ， ψ 对于 $\mathcal{P}(n,s,d)$，都有最大公因式 $_{x_i}(\phi(G_i),\psi(G_i)) = 1$。

我们进一步指出，对于任意两个乘积 f， g 其中各元素取自 $\mathcal{P}(n,s,d)$，我们有 $f|g$ 当且仅当 $\nu_\phi(f) \leqslant \nu_\phi(g)$ 对于每个本原因子 ϕ，对于 $\mathcal{P}(n,s,d)$均成立；这只是我们的唯一分解定理的一个简单推论。因此， G较弱的第二个性质足以用于对 f 和 g进行整除性测试：只需检查 $f(G_i)|g(G_i)$ 作为 x_i 中的多项式 i对每个 。这一点在 定理 4.14 中形式化地给出。

零特征或大特征。 在 [BV25，引理 4 的证明中，5], Bisht 和 Volkovich 考虑了西尔维斯特矩阵 $M_{x_i}(f^\alpha, g^\beta)$ ，该矩阵由经过适当选择的多项式的小次幂构成f， g并将它们视为关于 x_i的多项式。随后，他们利用 G 的第一条性质证明，与生成器 G_i复合后，该矩阵的秩不会改变，即秩 $(M_{x_i}(f^\alpha, g^\beta))$ =秩$(M_{x_i}(f(G_i)^\alpha, g(G_i)^\beta))$。由此，他们推断 $\phi(G_i)$ 和 $\psi(G_i)$不可能有公因子，因为如果存在这样的因子，矩阵的秩就会降低。

为了得到我们的结果，我们不考虑西尔维斯特矩阵 $M_{x_i}(f^\alpha, g^\beta)$，而是考虑定义多项式 fg 的判别式 $\Delta(fg)$ 的矩阵。也就是说，我们考虑由 fg 及其关于 x_i 的形式导数构成的西尔维斯特矩阵 $M_{x_i}(fg,(fg)')$ ， $(fg)'$。下面我们解释其中的思路。

设 $\phi|f$， $\psi|g$ 是 $\mathcal{P}_{\text{Factors}}(n,s,d)$中的两个不可约元素。将所有多项式视为关于某个变量 x_i的多项式，并考虑西尔维斯特矩阵 $M_{x_i}(fg,(fg)')$。不难证明，该矩阵的秩实质上统计了 fg 中不同不可约因子的数量（需适当计数；形式化表述见 断言 4.16）。由于$M_{x_i}(fg,(fg)')$ 的所有子式的行列式都是低次稀疏多项式，因此我们能够利用 G的第一条性质证明，将该矩阵与 G_i复合后，其秩不会改变。特别地，这意味着 $\phi(G_i)$， $\psi(G_i)$ 不可能有公因子；否则， $fg(G_i)$ 中不同因子的数量将少于 fg中的数量，这与秩保持不变这一事实矛盾。由此，我们通过考察输入与生成器 G_i复合后的因式分解，得到 定理 1.16 和 定理 1.15 。

1.4.2 寻找稀疏多项式乘积的多二次因子

接下来，我们将展示如何利用第 1.4.1 节中的结果来证明定理 1.14。为便于说明，我们在此假设$(\mathbb{F}) = 0$。

我们首先考虑如下一般情形。设 f 是$\mathcal{P}_{\text{Factors}}(n,s,d)$中多项式的乘积；也就是说， f 是个体次数有界为 d的 s-稀疏多项式的因子的乘积。考虑某个不可约的 $\phi \in \mathcal{P}_{\text{Factors}}(n,s,d)$。由 G 的第二条性质可知，对每一个使得 $x_i \in$ 变量集(ϕ)的 i， 整除 $f(G_i)$ 的 $\phi(G_i)$ 的最高次幂恰好是 ϕ 作为 f的因子的重数。特别地，每当 ϕ 是个体次数有界为 d的不可约 s-稀疏多项式时，这一点都成立。基于这一观察，我们现在可以描述用于寻找多线性因子的算法。假设 $f = \prod_{i=1}^{\ell} \phi_i^{e_i} \in \mathbb{F}[x_0, \mathbf{x}]$， 其中 ϕ_i 是 $(n+1,s,d)$-稀疏多项式的因子。进一步假设我们拥有对

f，并且我们希望恢复其全部不可约多线性因子。 [2] 我们的算法遵循以下步骤：

步骤 1 - 归一化与递归。在此步骤中，我们希望用另一个多项式 f 替换输入多项式 f，它仍然保留关于 f 的因子的所有信息，但经过了归一化（即其作为 x_0 的多项式时常数项为 1）。为此，我们采用一种标准方法，例如 BV25、BSV 中所用的方法。令 $f = \sum_{i=0}^{\ell d} f_i x_0^i$，其中 $f_i \in \mathbb{F}[\mathbf{x}]$，并令 $\tilde{f}(y, x_1, \ldots, x_n) = f(f_0 y, x_1, \ldots, x_n)/f_0$。 [3] 注意，由于我们可以黑盒访问 f，因此也可以黑盒访问 f_0，进而访问 $\tilde{f}$。不难看出，f 的每个多线性因子 $f_i(\mathbf{x}) = a_i(\mathbf{x})x_0 + b_i(\mathbf{x})$ 现在都对应于 $\tilde{f}$ 的一个多线性因子 $\frac{a_i f_0}{b_i} y + 1$。因此可知：

(1) 若 $a_i \neq 0$，则关于 a_i 和 b_i 的全部信息都包含在 $\tilde{f}$ 中。

(2) 若 $a_i = 0$，则因子 b_i 无法从 $\tilde{f}$ 加以研究。然而请注意，在这种情况下 $b_i | f_0$，因此可以从 f_0 的因式分解来研究它。

注意，f_0 与 f 具有相同的结构——它本身是个体次数为 d 的 s-稀疏多项式的因子的乘积。因此，通过一次递归调用，我们可以求出 f_0 的所有不可约多线性因子。于是，为了理解带有 $a_i = 0$ 的因子，我们只需理解：对于 f_0 的任意多线性不可约因子 ϕ，它整除 f 多少次。我们将在下一步给出一种做法。目前，这一步的主要收益是归一化 $\tilde{f}$，以及对 f_0 的多线性因子 ϕ_i 的计算。

步骤 2 - 获取因式分解。正如我们在第一步中看到的，因子有两类——一类含有 $a_i = 0$，另一类含有 $a_i \neq 0$。因此，我们将算法分成两部分：

(1) 求出满足 $a_i = 0$ 的因子。在这种情况下，有 $b_i | f_0$。因此，正如我们已经看到的，只需确定 f_0 的哪些不可约因子整除 f（以及整除的次数）。为此，请注意，对于 f_0 的一个给定稀疏不可约因子 ϕ，根据 G 的第二个性质，以及 f 是 $\mathcal{P}_{\text{Factors}}(n+1, s, d)$ 中元素的乘积这一事实，要计算 ϕ 作为 f 的因子的重数，只需检查 $\phi(G_j)$ 整除 $f(G_j)$ 的次数，其中 $x_j \in$ 是变量 (ϕ)。这很容易做到，例如，可以对两个多项式进行因式分解；这是可行的，因为它们的变量数和多项式次数都是常数。

(2) 恢复满足 $a_i \neq 0$ 的因子。在这种情况下，利用第一步，我们可以通过对 $\tilde{f}(y, G)$ 进行因式分解来获得 $\frac{a_i(G) f_0(G)}{b_i(G)}$。从这样的表达式中恢复 a_i，就是我们先前提到的有理插值问题。这里我们还知道，b_i 是 f_0 的一个多线性因子。因此，要找到 b_i，只需弄清楚，对于 f_0 的每个多线性不可约因子 ϕ，能整除 b_i 的 ϕ 的最高次幂是多少。对于任意这样的因子 ϕ，利用 a_i 与 b_i 互素这一事实（否则 $a_i x_0 + b_i$ 将不可约），可知 $\phi^k | b_i$ 当且仅当 $\phi^{t-k+1} \nmid \frac{a_i f_0}{b_i}$，其中 t 是 ϕ 作为 f_0 的因子的重数。由此可知，我们只需弄清楚 ϕ 作为 $\frac{a_i f_0}{b_i}$ 的因子的重数。由于 a_i 是稀疏的，且 f_0 是 $\mathcal{P}_{\text{Factors}}(n, s, d)$ 中多项式的乘积，因此 $\frac{a_i f_0}{b_i}$ 也是 $\mathcal{P}_{\text{Factors}}(n, s, d)$ 中多项式的乘积。于是，我们只需计算

[2] 通用算法可找出输入的多二次稀疏不可约因子。为使这里的表述尽可能简单，我们只考虑多线性因子。[3] 很容易将问题归约到情形 $f_0 \neq 0$，因此，为使表述尽可能简单，我们假设该情形成立。

$\phi(G_j)$ 作为 $\frac{a_i(G_j)f_0(G_j)}{b_i(G_j)}$ 的因子时的重数（其中 $x_j \in$ 变量集 (ϕ)）。这样就能恢复 b_i，进而也能恢复 a_i，从而完成该算法。

在上述内容中，我们实际上解决了一个有理插值问题。也就是说，我们可以访问 $\frac{af}{b}$，其中$b|f$，并需要根据这些信息恢复 a 和 b。我们断言，上述算法实际上适用于更一般的情形。该推广的精确表述及其证明见定理 5.1和定理 5.2。

最后，我们给出以下几点说明。首先，我们注意到，可以修改该算法，从而找出 f的所有多二次不可约因子，而不仅仅是多线性的那些。我们还要指出，即使去掉对域 $\mathbb{F}$的假设，上述算法仍然有效，不过运行时间会稍长一些。这些断言的完整证明见第 7 节。

1.4.3 寻找稀疏多项式的稀疏因子

现在我们来介绍主要结果（定理 1.9）背后的思路。回想一下，我们的目标是输出给定 (n, s, d)-稀疏多项式 f的所有稀疏因子。也就是说，我们希望高效地找出 f 中那些能够高效地“写出”的因子。

为了得到这样的算法，一种自然的做法是遵循第1.4.2节中所介绍算法的思路。该算法的第一步是进行归一化，然后对常数项 f_0进行递归调用以作因式分解。但现在出现了一个问题——这次递归调用只会返回 f_0的不可约稀疏因子，而 f (b_i 按照第1.4.2节中的术语 f)的每个因子的常数项，可能是可约的稀疏因子 f_0，其不可约因子并不稀疏；在这种情况下，这次递归调用并未提供任何有关b_i的有用信息。特别是，我们无法像第1.4.2节中那样，通过研究 f_0 的每个不可约稀疏因子整除它的次数来重构 b_i。事实上，第1.4.2节中所介绍算法之所以有效，真正的原因在于，在该情形下，b_i 是稀疏多线性（更一般的情形下则是稀疏多二次）的，因此其不可约因子也如此；所以，我们确实能从递归调用中获得有关 b_i 的信息。换句话说，那里的常数项属于一个在因式分解下封闭的稀疏多项式类。

这促使我们考虑找出稀疏多项式的所有稀疏因子这一问题。这样，我们仍然能够获得因子的常数项信息，而对于 f ——递归调用实际上会计算出这些常数项。事实上，稀疏多项式的稀疏因子的 (n, s, d)——稀疏多项式本身也是 (n, s, d)——稀疏多项式的稀疏因子，因此这个类在因式分解下是封闭的。这使我们能够在确定性多项式$(n, s^{\mathrm{poly}(d)})$——时间内恢复所有这些稀疏因子。事实上，递归调用返回的是常数项的所有稀疏因子 f_0，而不仅仅是其不可约因子的集合，因此我们可以直接穷举这些因子的列表来找出b_i。这就得到定理 1.9。

乍看之下，这种方法可能令人惊讶，因为存在这样的算法，似乎意味着可能的输出数量至多为 $\mathrm{poly}(n, s^{\mathrm{poly}(d)})$。事实上，定理 1.8 保证了 s 个体次数有界的稀疏多项式 d 至多有 $d \log s$ 个不可约因子（不包括单项式），因此，总共至多有 s^d 个因子（不能被单项式整除）。为了证明定理 1.8，我们证明，如果 $f \in \mathcal{P}(n, s, d)$ 不能被任何单项式整除，那么就可以找出一个包含 $\log s$ 个变量的集合，使得 f 的每个不可约因子都至少依赖其中一个变量。从这一断言很容易推出，f至多有 $d \log s$个不可约因子，因此总共至多有 $2^{d \log s} = s^d$ 个因子。为证明这样的变量集合确实存在，我们分析 f的牛顿多面体，并证明如果满足该性质的最小变量集大小为 k，那么 f 的牛顿多面体至少有

个 2^k 个顶点。这在某种程度上是直观的，因为集合中每增加一个变量，实际上都会给多面体增加一个维度，因此使其顶点数翻倍。

我们注意到定理 1.9 并不能由 [BSV20]的方法推出；事实上，那里需要对所有不可约因子（即 f 的那些因子）的稀疏度给出界，才能覆盖不同因子的所有结式；因此，将他们的方法应用于该问题，运行时间至少为 $s^{O(d^2 \log n)}$，而这正是他们对 f的因子稀疏度所给出的界。

我们的算法的一个主要缺点在于，它并不能告诉我们何为不可约稀疏因子（即输入多项式的那些因子），因为我们并不熟悉一种能从所得输出中高效识别不可约多项式的方法。另一方面，如果可以保证 f 是不可约稀疏因子的乘积，那么我们就能轻易地在所有稀疏因子中找出不可约因子。由此便得到推论 8.6，从而解决了问题 1.3 的特殊情形，亦即(n, s, d)-稀疏多项式的情形。

这些思路也用于定理 1.10的证明。回想一下，问题 1.4 要求设计一个确定性多项式时间算法：给定对若干个 (n, s, d)项稀疏多项式之乘积的黑盒访问，返回这些因子。虽然当项数无界时，我们不知道是否存在多项式时间算法，但我们给出了一种算法，其运行时间关于项数为准多项式时间，这是解决该问题的第一步。该算法是对上文所述算法自然而直接的推广。不幸的是，它需要准多项式时间，因为我们实际上要找出 s的所有有界个体次数稀疏因子 d，而 f这类因子（可能）有准多项式多个（如推论 8.3所示）。

1.4.4 稀疏多项式的因式分解

最后，我们讨论对 (n, s, d) 项稀疏多项式及其乘积进行因式分解的算法。这些算法与我们之前介绍的算法基本相同；唯一的区别是，现在处理 $\mathcal{P}_{\text{Factors}}(n, s, d)$ 中的元素时，我们将它们视为 $\mathcal{P}(n, s^{O(d^2 \log n)}, d)$ 的元素。

例如，考虑我们用于找出某个 (n, s, d)-稀疏多项式的所有稀疏因子的算法。稍作修改后，我们可以使其在多项式 $(n, s^{O(d^2 \log n)}, d)$-时间内输出输入多项式的所有$(n, s^{d^2 \log n})$-稀疏因子。根据 [BSV20],，这些就是所有可能的因子。很容易从中识别出不可约因子。例如，我们可以尝试对任意两个因子 ϕ 和 ψ，检查是否 $\psi|\phi$，方法是对它们的商进行稀疏插值——该商应为一个 $s^{O(d^2 \log n)}$-稀疏多项式。

至于我们用于对 $\mathcal{P}_{\text{Factors}}(n, s, d)$ 个元素的乘积进行因式分解的算法，它与第1.4.2节中描述的算法完全相同，只需考虑到所有相关的不可约因子都是 $s^{O(d^2 \log n)}$-稀疏的，而非 s-稀疏且多二次。

1.5 组织结构

在第2节我们给出本论文所需的全部背景。在第3节我们构造生成元 G，它是本论文的主要技术工具。在第 4 节我们证明 G所需的基本性质，既针对任意域，也针对特征为零或足够大的情形，并应用它们得到我们的整除性测试与精确幂测试结果（定理 1.16，定理 1.15）。在第 5 节我们证明有理插值结果。

我们在第 *6* 节中给出一个元算法，我们的所有算法都以此为模型。随后，我们展示如何应用它来找出 $\mathcal{P}_{\text{Factors}}(n, s, d)$ 个多二次稀疏不可约因子，这些因子来自多项式的乘积（定理 1.14）。在第8节中，我们给出 (n, s, d)稀疏多项式的因子个数上界（定理 1.8），并利用该上界找出这些多项式的全部

稀疏因子（定理 1.9），并为问题 1.4 （定理 1.10）提供一个部分解答。在第 9 节中，我们给出用于对 (n, s, d)稀疏多项式及其乘积进行因式分解的算法，该算法基于 [BSV20]（定理 1.12、定理 1.13）。第二个结果为问题 1.4提供了另一种部分解答。最后，我们在第 10 节中作简短讨论。在附录 A中，我们为第 4 节中的一些技术性断言给出不同但较弱的证明。由于证明技巧不同，尽管结果略弱，我们仍决定给出这些证明。

2 预备知识

我们用 $\boldsymbol{x}$ 表示由变量 $\boldsymbol{x} = (x_1, \ldots, x_n)$ 构成的 n 元向量。一般而言，我们将使用 $\boldsymbol{x}$、$\boldsymbol{e}$ 等粗体字母表示 n 元组，其中该元组的定义域总能从上下文中明确。

我们称两个非零多项式 f、g 相伴，如果其中一个是另一个的标量倍数。我们还陈述高斯引理的以下特殊情形。

引理 2.1 (高斯引理)。多项式 $f(x_0, \boldsymbol{x}) = \sum_{i=0}^{d} c_i(\boldsymbol{x}) x_0^i \in \mathbb{F}[\boldsymbol{x}][x_0] \setminus \mathbb{F}[\boldsymbol{x}]$ 不可约，当且仅当它在 $\mathbb{F}(\boldsymbol{x})[x_0]$ 中不可约，且 最大公因式$(c_0, \ldots, c_d) = 1$。

2.1 反首一多项式与归一化

在本文中，我们将使用以下定义。

定义 2.2. 如果多项式 $f \in \mathbb{F}[x_0, \boldsymbol{x}]$ 关于 x_0 的常数项（视为关于 $\mathbb{F}(\boldsymbol{x})[x_0]$ 的多项式）为 1，则称其关于 x_0 是反向首一的。等价地，如果 f 具有形式 $f = 1 + \sum_{i \geqslant 1} f_i x_0^i$，其中 $f_i \in \mathbb{F}[\boldsymbol{x}]$，则称其关于 x_0 是反向首一的。

我们证明关于反向首一多项式的以下简单断言。

断言 2.3. 设 $f \in \mathbb{F}[x_0, \boldsymbol{x}]$ 是关于 x_0的反向首一多项式。那么，f 的不可约因子个数至多为次数 $_{x_0}(f)$。

证明。首先注意，f 的所有不可约因子都必须依赖于 x_0。事实上，写作 $f = \sum_{i=0}^{\deg_{x_0}(f)} f_i x_0^i$，其中 $f_i \in \mathbb{F}[\boldsymbol{x}]$。根据反向首一多项式的定义，$f_0 = 1$。显然，任何不依赖于 x_0 的 f 的因子都必须整除 f_0，因此不存在这样的非平凡因子。断言由此成立，因为依赖于x_0的 f 的不可约因子个数至多为 $\deg_{x_0}(f)$ 。 □

接下来定义 一种归一化 操作，可将多项式转化为反向首一多项式。

定义 2.4 （归一化）。设 $f(x_0, \boldsymbol{x}) = \sum_{i=0}^{d} f_i(\boldsymbol{x}) x_0^i \in \mathbb{F}[x_0, \boldsymbol{x}]$，使得 $x_0 \nmid f$。f 关于 x_0 的归一化是反向首一多项式

$$\tilde{f}(y, \boldsymbol{x}) = f(y f_0, x_1, \ldots, x_n)/f_0 = 1 + \sum_{i=1}^{d} f_i(\boldsymbol{x}) f_0^{i-1}(\boldsymbol{x}) y^i. \tag{1}$$

归一化是一种标准且广为人知的技术，我们下一个断言的不同版本也可在 [BSV20，BV25]中找到。我们对该断言的版本描述了 $\tilde{f}$ 的因式分解结构，这将在我们的证明和重构算法中发挥重要作用。

命题 2.5. 设 f 如 *定义 2.4* 中所述。假设 $f = \prod_{i=0}^{N} h_i \in \mathbb{F}[x_0, \boldsymbol{x}]$ 是将 f 分解为不可约因子 h_i 的因式分解，并将每个因子记为 $h_i = \sum_{j=0}^{d_i} c_{i,\,j} x_0^{\,j}$ （其中 $c_{i,\,j} \in \mathbb{F}[\boldsymbol{x}]$）。则 $\tilde{f}$ 的因式分解由下式给出

$$\tilde{f} = \prod_{i=0}^{N} \tilde{h}_i \,, \textit{where} \quad \tilde{h}_i = \sum_{j=0}^{d_i} \frac{c_{i,j}}{c_{i,0}} f_0^{\,j} y^j \,.$$

此外，f 的任意形如 $\sum_{j=0}^{d} \zeta_j x_0^{\,j}$（其中 $\zeta_j \in \mathbb{F}[\boldsymbol{x}]$）的因子（不一定是不可约因子）都对应于 $\tilde{f}$ 的一个形如 $\sum_{j=0}^{d} \frac{\zeta_j}{\zeta_0} f_0^{\,j} y^j$ 的因子。特别地，f 的每个形如 $a_2 x_0^2 + a_1 x_0 + b$ 的多二次因子都对应于 $\tilde{f}$ 的一个形如 $1 + \frac{a_1 f_0}{b} y + \frac{a_2 f_0^2}{b} y^2$ 的因子。此外，我们可以通过对 f 的黑盒访问，获得对 $\tilde{f}$ 的黑盒访问。

证明。 首先，注意到 $f_0 = \prod_{i=0}^{N} c_{i,0}$。接下来，令

$$\tilde{f}(y, \boldsymbol{x}) = f(y f_0, \boldsymbol{x}) / f_0 = \prod_{i=0}^{N} \frac{h_i(y f_0, \boldsymbol{x})}{c_{i,0}} \,.$$

定义

$$\tilde{h}_i(y, \boldsymbol{x}) := \frac{h_i(y f_0, \boldsymbol{x})}{c_{i,0}} = \sum_{j=0}^{d_i} \frac{c_{i,j} (y f_0)^j}{c_{i,0}} = \sum_{j=0}^{d_i} \frac{c_{i,j}}{c_{i,0}} f_0^{\,j} y^j \in \mathbb{F}[y, \boldsymbol{x}] \,.$$

我们断言 $\tilde{h}_i$ 是不可约的。如果 $d_i = 0$，那么 $\tilde{h}_i = 1$，因此显然成立。假设 $d_i > 0$。根据定义，h_i 是作为 $\mathbb{F}[x_0, \boldsymbol{x}]$ 中的元素的不可约多项式，因此它也是作为 $\mathbb{F}(\boldsymbol{x})[x_0]$ 中的元素的不可约多项式。在这个环中有 $\tilde{h}_i(x_0) = h_i(f_0 x_0)/c_{i,0}$。特别地，$\tilde{h}_i$ 与 h_i 相差一个变量的线性变换以及一个标量乘法（该标量取自域 $\mathbb{F}(\boldsymbol{x})$），所以它也不可约。由于 $\tilde{h}_i$ 的常数项为 1，根据高斯引理（*引理 2.1*），$\tilde{h}_i$ 作为 $\mathbb{F}[x_0, \boldsymbol{x}]$ 的元素是不可约的，正如我们所希望的。

“此外”部分源于这样一个事实：关于 $\tilde{h}_i$ 形式的论证，无论 h_i 是否不可约都成立。

关于 $\tilde{f}$ 的黑盒访问的断言可由简单插值得出：给定一个求值点 $(y_0, \boldsymbol{\alpha})$，我们将多项式 $f(x_0, \boldsymbol{\alpha})$ 作为关于 x_0 的多项式进行插值，从而获得对每个 $f_i(\boldsymbol{\alpha})$ 的访问。然后，只需使用（1）的右侧即可得到 $\tilde{f}(y_0, \boldsymbol{\alpha})$。特别地，每次查询~ f 至多需要 $d+1$ 次对 f的查询。 □

2.2 多项式的因式分解与插值

我们首先陈述一个关于一元多项式因式分解的结果。第一个是 Lenstra、Lenstra 和 Lovász 的经典算法 [LLL82]。

定理 2.6（在 $\mathbb{Q}$ 上的一元因式分解）。设 $f \in \mathbb{Q}[x]$ 的次数为 d，位复杂度为 B。那么，存在一个确定性算法，能够输出 f 的所有不可约因子，其运行时间为 多项式(d, B)。

以下定理可参见，例如： [vzGS92，第 9]节。

定理 2.7（有限域上的一元因式分解）。 设 $\mathbb{F}_q$ 是一个大小为 $q = p^k$、特征为 $p > 0$的域。设 $f(x) \in \mathbb{F}_q[x]$ 是一个次数为 d的多项式。那么，存在一种确定性算法，可在 poly$(p, \log q, d)$时间内输出 f 的所有不可约因子。

接下来，我们陈述一个关于多变量情形下确定性多项式因式分解的定理。参见例如 [Len85、Kal85b]。对于所有调用多项式因式分解的算法，这些结果都确定了它们对域、域的特征以及输入多项式位复杂度的依赖关系。

定理 2.8. 存在一种确定性算法，给定一个 n元、总次数为 d且位复杂度为 B的多项式 $f \in \mathbb{F}[x]$，可输出其不可约多项式因式分解。其运行时间为多项式$(d^n) \cdot T(d, B, \mathbb{F})$，其中 $T(d, B, \mathbb{F})$ 是对次数为 d、位复杂度为 B的多项式在 $\mathbb{F}$上进行一元因式分解的运行时间。

接下来，我们陈述一个关于少量变量的多项式插值的众所周知的引理。

引理 2.9. 设 $f \in \mathbb{F}[x_1, \ldots, x_n]$ 是一个 n元、总次数为 d的多项式。那么，存在一种确定性多项式时间算法，给定对 f 的黑盒访问，可返回其单项式表示。其运行时间为多项式$((d+1)^n)$。

最后，我们通过回顾拉格朗日插值的定义来结束本节。

定义 2.10。 设 $\alpha_1, \ldots, \alpha_r \in \mathbb{F}$ 是互不相同的域元素。那么多项式

$$A_i(x) = \prod_{1 \leqslant j \leqslant r, j \neq i} \frac{x - \alpha_j}{\alpha_i - \alpha_j} \tag{2}$$

满足 $\deg(A_i) = r - 1$ 且 $A_i(\alpha_j) = \delta_{i,j}$，其中 $\delta_{i,j}$ 是克罗内克 δ。 A_i 称为 $\alpha_1, \ldots, \alpha_r$ 的拉格朗日插值多项式。

2.3 稀疏多项式

我们给出一些定义，并陈述若干关于稀疏多项式的已知结果。

定义 2.11。 s-稀疏多项式 $f \in \mathbb{F}[x_1, \ldots, x_n]$ 是至多由 s 个单项式相加得到的多项式。若 f 作为 $\mathbb{F}(x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n)[x_i]$ 中的元素，其次数对每个 $1 \leqslant i \leqslant n$ 都至多为 d，则称其具有有界个体次数 d。我们将具有有界个体次数 d 的 n 元、s-稀疏多项式类记为 $\mathcal{P}(n, s, d)$，并将其中的元素称为 (n, s, d)-稀疏多项式。此外，我们将 $\mathcal{P}(n, s, d)$ 中元素的因子类记为 $\mathcal{P}_{\text{Factors}}(n, s, d)$。请注意，我们所说的因子是指 (n, s, d)-稀疏多项式的所有可能因子，而不仅仅是不可约因子。

接下来的两个引理给出了两种特殊情形，在这些情形中，稀疏多项式的因子的稀疏度上界是已知的。

引理 2.12（[G KL12，引理 10]）。 一个多线性因子，对 s-稀疏多项式而言，仍然是 s-稀疏的 .

引理 2.13 （[Vol17，引理 32]）。 设 g 是一个 s-稀疏多二次多项式（即，g 的各变元次数至多为二），并假设 $f|g$。则 f 也是 s-稀疏的。

在 [BSV20], 中，Bhargava、Saraf 和 Volkovich 给出了 (n, s, d)-稀疏多项式的因子的稀疏度上界，并得到了目前已知最好的 (n, s, d)-稀疏多项式因式分解算法。接下来我们陈述这些结果。

定理 2.14 ([BSV20，定理 1])。 $\mathcal{P}_{Factors}(n, s, d) \subset \mathcal{P}(n, s^{O(d^2 \text{对数} n)}, d)$。换言之，$n$-变量中具有有界个体次数 d 的 s-稀疏多项式的每个因子都是 $s^{O(d^2 \text{对数} n)}$-稀疏的。

在接下来的结果中 最后，根据 定理 2.8，还应添加一个取决于域的因子 .

定理 2.15 ([BSV20，定理 2])。 存在一个确定性多项式$(n, d^d, s^{d^7}$ 对数$^n)$-时间算法，它以一个 (n, s, d)-稀疏多项式为输入，并返回其不可约因子的因式分解。

为了证明定理 2.14，作者通过研究稀疏多项式的牛顿多面体来研究稀疏多项式。由于我们也会使用牛顿多面体的相关术语，因此在此给出牛顿多面体的定义。

定义 2.16。 多项式 $f = \sum_e \gamma_e x^e$ 的牛顿多面体记作 P_f，它是点集 $e \in \mathbb{R}^n$ 的凸包。多面体 P_f 的顶点集合记作 $V(P_f)$。

以下众所周知的事实可见于 [BSV20，推论 3。5]。

引理 2.17。 设 $f|g$ 是两个多项式，并假设 g 是 s-稀疏的。那么，$|V(f)| \leqslant |V(g)| \leqslant s$。

2.4 结式与判别式

接下来我们讨论结式与判别式。详细介绍请参见 [GKZ08] 。

定义2.18。 设 $f, g \in \mathbb{F}[x]$ 是两个一元多项式，其次数分别为 d_1, d_2。定义 $\phi : \mathbb{F}[x]_{\leqslant(d_1-1)} \times \mathbb{F}[x]_{\leqslant(d_2-1)} \to \mathbb{F}[x]_{\leqslant(d_1+d_2-1)}$ 为

$$\phi(h_1, h_2) = h_1 g + h_2 f. \tag{3}$$

这里，$\mathbb{F}[x]_{\leqslant(d)}$ 是由 $\mathbb{F}$ 上次数至多为 d 的一元多项式构成的 $(d+1)$ 维 $\mathbb{F}$ 向量空间。该空间的标准单项式基由单项式 1、x、$\ldots$、x^d 构成。

f 和 g 的西尔维斯特矩阵 $M(f, g)$ 是表示 ϕ 的 $(d_1 + d_2) \times (d_1 + d_2)$ 矩阵，所依据的基为标准单项式基。该矩阵的行列式称为 f 和 g 的结式，记作 $\mathrm{res}(f, g) \in \mathbb{F}$。若 $f, g \in \mathbb{F}[x_0, x_1, \ldots, x_n]$ 是两个多元多项式，当将它们视为 $\mathbb{F}(x_1, \ldots, x_n)[x_0]$ 中的元素时，我们用 $M_{x_0}(f, g)$ 和 $\mathrm{res}_{x_0}(f, g) \in \mathbb{F}[x_1, \ldots, x_n]$ 分别表示 f 和 g 的西尔维斯特矩阵与结式。

定义 2.19. 设 $f \in \mathbb{F}[x]$为一元多项式。f 的判别式定义为

$$\Delta(f) = \mathrm{res}(f, f'), \tag{4}$$

其中，f' 是 f的标准形式导数。对于多元多项式 $f \in \mathbb{F}[\mathbf{x}]$，用 $\Delta_{x_i}(f) \in \mathbb{F}[x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n]$ 表示将 f 视为 $\mathbb{F}(x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n)[x_i]$ 上的元素时的判别式。

引理 2.20. 设 $\mathbb{F}$ 为一个域。

(1) 设 $f, g \in \mathbb{F}[x]$ 为两个首一多项式。那么，$\mathrm{res}(f, g) = \prod_{\alpha: f(\alpha)=0} g(\alpha)$，其中乘积遍历$\mathbb{F}$上 f 的所有根 α。 −

(2) $f, g \in \mathbb{F}[x]$ 在 $\mathbb{F}[x]$ 中有公因子，当且仅当它们的结式为零。

(3) 若 $char(\mathbb{F}) = 0$，多项式 f 无平方因子当且仅当 $\Delta(f) \neq 0$。此外，即使 $deg(f) < char(\mathbb{F})$，最后一条结论也成立。

(4) 设 $f, g \in \mathbb{F}[x_0, \mathbf{x}]$ 为两个个体次数有界为 d 的 s-稀疏多项式。那么，它们的西尔维斯特矩阵 $M_{x_0}(f, g)$ 是一个 $2d \times 2d$ 矩阵，其元素是 $\mathbb{F}[\mathbf{x}]$中的 s-稀疏元素，并且，维数(核 $(M_{x_0}(f, g))) =$次数$_{x_0}$(最大公因式(f, g))。特别地，

(1) 若 $f, g \in \mathbb{F}[x_0, \boldsymbol{x}]$ 为两个个体次数有界为 d 的 s-稀疏多项式，则 结果式$_{x_0}(f, g) \in \mathbb{F}[\boldsymbol{x}]$ 是一个 $(2d)!s^{2d}$-稀疏多项式，其个体次数有界为 $2d^2$。*(2)* 若 $f \in \mathbb{F}[x_0, \boldsymbol{x}]$ 是一个个体次数有界为 d 的 s-稀疏多项式，则 $\Delta_{x_0} f \in \mathbb{F}[\boldsymbol{x}]$ 是一个 $(2d)!s^{2d}$-稀疏多项式，其个体次数有界为 $2d^2$。

证明。 前三个陈述是标准的。最后一个陈述也是众所周知的，其解释例如在 [卷17], 中以子结式的语言给出。我们在此也给出“此外”部分的完整证明，因为它在引理 4.15的证明中起着重要作用。

设 f、 $g \in \mathbb{F}[x_0, \boldsymbol{x}]$ 是两个个体次数有界为 d的 s-稀疏多项式。将二者都视为 $\mathbb{F}(\boldsymbol{x})[x_0]$中的元素，并写成 $f = \prod_i h_i^{a_i}$、 $g = \prod_i h_i^{b_i}$，其中 h_i 是 fg的全部不可约因子，且 a_i、 $b_i \geqslant 0$（例如，若 f、 g 互素，则对每个 i有 $a_i b_i = 0$）。注意最大公因式$(f, g) = \prod_i h_i^{\min(a_i, b_i)}$ 与最小公倍式$(f, g) = \prod_i h_i^{\max(a_i, b_i)}$。

考虑一个元素 $(w_1, w_2) \in \ker(M_{x_0}(f, g))$。根据定义，

$$w_1 g = -w_2 f. \tag{5}$$

因此，考虑等式两边的不可约因子因式分解，必有 $\prod_i h_i^{\max(a_i - b_i, 0)} | w_1$ and $\prod_i h_i^{\max(b_i - a_i, 0)} | w_2$。于是，我们可以写成

$$w_1 = \prod_i h_i^{\max(a_i - b_i, 0)} \tilde{w}_1, \quad w_2 = \prod_i h_i^{\max(b_i - a_i, 0)} \tilde{w}_2 \,. \tag{6}$$

由此可得

$$\begin{aligned} w_1 g &= \prod_i h_i^{b_i + \min(a_i - b_i, 0)} \tilde{w}_1 = \prod_i h_i^{\max(a_i, b_i)} \tilde{w}_1 = \operatorname{lcm}(f, g) \tilde{w}_1 \\ w_2 f &= \prod_i h_i^{a_i + \min(b_i - a_i, 0)} \tilde{w}_2 = \prod_i h_i^{\max(a_i, b_i)} \tilde{w}_2 = \operatorname{lcm}(f, g) \tilde{w}_2 \,. \end{aligned} \tag{7}$$

结合（5）和（7），得到 $\tilde{w}_1 = -\tilde{w}_2$。由于 $\deg w_1$，$\deg w_2 \leqslant d - 1$，由（6）可知，$\ker$ 中的一个元素 $(M_{x_0}(f, g))$ 对应于一个次数至多为 $\tilde{w}$ 的多项式

$$\begin{aligned} d - 1 - \sum_i \max(a_i - b_i, 0) \deg h_i &= \sum_i (a_i - \max(a_i - b_i, 0)) \deg(h_i) - 1 \\ &= \sum_i \min(a_i, b_i) \deg(h_i) - 1 = \deg_{x_0}(\gcd(f, g)) - 1 \,. \end{aligned} \tag{8}$$

因此，$\ker(M_{x_0}(f, g))$ 同构于 $\mathbb{F}[x]_{\leqslant (\deg_{x_0} \gcd(f, g) - 1)}$，其维数为 $\deg_{x_0} \gcd(f, g)$，如断言所述。

□

3 生成元构造

本节中，我们构造生成器，它是一个映射 $G : \mathbb{F}^6 \to \mathbb{F}^n$，满足 第 1.4.1 节中描述的性质。

接下来，令 m 为某个参数（在我们的设定中，处理 (n, s, d)-稀疏多项式时，通常取 $m = (nsd)^2$）。为定义生成器 $G_{(m)}$，我们采用以下记号约定。令 $q = \Theta(m)$ 为一个素数（具体而言， q 可以是大于 m 的最小素数）。我们还假设 $|\mathbb{F}| > ndq$。若不满足这一条件，则我们在大小为 $dnq < |\mathbb{E}| \leqslant (dnq)^2$ 的扩域 $\mathbb{F} \subset \mathbb{E}$ 上进行运算。

固定以下集合：

s[(1)] 一个包含 m 个不同模剩余类的大小为 m 的整数集合 $\mathcal{I} = \{1, \ldots, m\}$ q.

（2）一个大小为 m 的域元素集合 $\mathcal{A} = \{\alpha_1, \ldots, \alpha_m\} \subset \mathbb{F}$。对于每个 $1 \leqslant i \leqslant m$，令 A_i 为关于 $\mathcal{A}$ 的拉格朗日插值多项式。

（3）一个大小为 n 的域元素集合 $\mathcal{B} = \{\beta_1, \ldots, \beta_n\} \subset \mathbb{F}$。对于每个 $1 \leqslant i \leqslant n$，令 B_i 为关于 $\mathcal{B}$ 的拉格朗日插值多项式。

（4）一个大小为 n 的域元素集合 $\mathcal{C} = \{\gamma_1, \ldots, \gamma_n\} \subset \mathbb{F}$。对于每个 $1 \leqslant i \leqslant n$，令 C_i 为关于 $\mathcal{C}$ 的拉格朗日插值多项式。

（5）一个大小为 ndq 的域元素集合 $\mathcal{T} = \{\delta_1, \ldots, \delta_{ndq}\} \subset \mathbb{F}$。

为了定义我们的生成器，我们先定义 Klivans 和 Spielman 的生成器，并陈述它的一些性质。

定义 3.1（KS 生成器）。沿用上述记号，KS 生成器定义为

$$G^{KS}_{(m)}(x, y, z, w) = \sum_{i=1}^{m} A_i(y) \left((1 + B_1(z)(w-1))x^{i \bmod q}, \ldots, (1 + B_n(z)(w-1))x^{i^n \bmod q} \right). \quad (9)$$

定理 3.2 ([KS01，定理 11])。设 $f \in \mathbb{F}[x]$ 是一个 (n, s, d)-稀疏多项式。设 $m = (nds)^2$。那么，f 可以通过以下求值集合在时间 $\mathrm{poly}(n, s, d)$ 内重构：

$$\left\{ f\left(G^{KS}_{(m)}(x, y, z, w)\right) \mid y \in \mathcal{A},\ z \in \mathcal{B},\ x, w \in \mathcal{T} \right\}.$$

证明。注意，令 $y = a_i$ 和 $z = b_j$ 时，我们得到

$$G^{KS}_{(m)}(x, a_i, b_j, w) = \left(x^{i \bmod q}, \ldots, x^{i^{j-1} \bmod q}, w x^{i^j \bmod q}, x^{i^{j+1} \bmod q}, \ldots, x^{i^n \bmod q} \right).$$

该断言可由 [KS01，定理 11]中的分析得出。 □

我们还将使用 [SV09], 中定义的生成器，其公式为

$$G^{SV}(u, v) = (C_1(v)u, \ldots, C_n(v)u). \quad (10)$$

这个生成器的基本性质是，它使我们能够“复活”我们选定的某个变量。例如，通过设定$v = \gamma_j$，我们得到一个向量，该向量在除第 j个坐标之外的所有坐标上都为零，而在第 j个坐标处我们有变量 u。

我们现在可以定义我们的生成器了。

定义 3.3（生成器 $G_{(m)}$）。生成器 $G_{(m)} : \mathbb{F}^6 \to \mathbb{F}^n$ 定义为

$$G_{(m)}(x, y, z, w, u, v) = G^{KS}_{(m)}(x, y, z, w) + G^{SV}(u, v). \quad (11)$$

我们进一步定义通过复活某个变量得到的生成器。用 $G^{KS}_{(m)}(x, y, z, w)_i$表示 $G^{KS}_{(m)}(x, y, z, w)$的第 i个坐标。

对于任意 $1 \leqslant i \leqslant n$，令 $G_{(m,i)}$ 为通过将 $G_{(m)}$ 限制为 $u = u - G^{KS}_{(m)}(x, y, z, w)_i$ 和 $v = \gamma_i$ 而得到的生成器。也就是说，

$$G_{(m,i)}(x, y, z, w, u) = \left(G^{KS}_{(m)}(x, y, z, w)_1, \ldots, u, \ldots, G^{KS}_{(m)}(x, y, z, w)_n \right). \quad (12)$$

让我们解释一下这个定义。第一部分 $G^{KS}_{(m)}$ 使我们能够对 (n,s,d)-稀疏多项式进行插值（定理 3.2）。第二部分允许我们“恢复”任意选择的一个变量，这一特性将在分析中发挥重要作用。

下面的引理表明，我们的生成器确实满足第 1.4.1 节中所述的第一个性质。

引理 3.4. 设 f 是一个 (n,s,d)-稀疏多项式，且 $m=(nds)^2$。则可以在 多项式(m) 时间内从 $f(G_{(m)})$ 恢复 f。特别地， $f=0$ 成立当且仅当 $f(G_{(m)})=0$。事实上，对每一个 $1\leqslant i\leqslant n$，将 $G_{(m)}$ 替换为 $G_{(m,i)}$ 以及替换为 $G^{KS}_{(m)}$ 时，该陈述同样成立。

证明。 对每一个 $1\leqslant i\leqslant n$， $G_{(m)}$、 $G_{(m,i)}$ 和 $G^{KS}_{(m)}$ 中每一个的像都包含$G^{KS}_{(m)}$的像，因此该断言由定理 3.2得出。 □

We c用前一引理的如下三个推论来结束本节 .

推论 3.5. 假设 $f\in\mathcal{P}_{Factors}(n,s,d)$。则对 $m=(nds)^2$ 有 $f(G_{(m)})\neq 0$。事实上，对每一个 $1\leqslant i\leqslant n$，将 $G_{(m)}$ 替换为 $G_{(m,i)}$ 以及替换为 $G^{KS}_{(m)}$ 时，该陈述同样成立。

证明。 令 $g\in\mathcal{P}(n,s,d)$ 使得 $f|g$，并令 G 为以下任意一个 $G_{(m)}$、 $G_{(m,i)}$ 和 $G^{KS}_{(m)}$。 引理 3.4 表明 $g(G)\neq 0$。断言由以下事实得出： $f(G)|g(G)$。 □

推论 3.6。 令 $f\in\mathcal{P}_{Factors}(n,s,d)$。则，对于每个 $1\leqslant i\leqslant n$ 和 $m=(nsd)^2$，都有 $\deg_u(f(G_{(m,i)}))=\deg_{x_i}(f)$。

证明。 考虑首项系数[4] $\mathrm{LC}_{x_i}(f)$，将 f 视为关于 x_i 的多项式，并令 $g\in\mathcal{P}(n,s,d)$ 为满足 $f\mid g$的任意多项式。则首项系数 $\mathrm{LC}_{x_i}(g)$（将其视为关于 g 的多项式）可被 $\mathrm{LC}_{x_i}(f)$整除。由于 g是 s-稀疏的，其首项系数也是 s-稀疏的。因此，$\mathrm{LC}_{x_i}(f)$ 是某个 $(n-1,s,d)$-稀疏多项式的因子，于是根据推论 3.5，当代入除 x_i 之外的所有变量时，有 $\mathrm{LC}_{x_i}(f)(G^{KS}_{(m)})\neq 0$。

由（12），$\mathrm{LC}_{x_i}(f)(G^{KS}_{(m)})$ 恰好是 $f(G_{(m,i)})$ 作为 u 的多项式时的首项系数。因此，我们必须有 $\deg_u f(G_{(m,i)})=\deg_{x_i}(f)$。 □

推论 3.7。存在一个确定性的 多项式(n,d,s,ℓ) 时间算法，给定由 $\mathcal{P}_{Factors}(n+1,s,d)\subset\mathbb{F}[x_0,\mathbf{x}]$ 中 ℓ 个多项式的乘积 f 的黑盒访问，可返回 f 中 x_0 作为因子的重数 k。此外，若 $f=x_0{}^k g$，则可以通过对 f 的黑盒访问，获得对 g 的黑盒访问。

证明。设 f 的不可约因式分解为 $f=x_0{}^k\prod_i f_i$，其中 $f_i\neq x_0$。对于每个 i，令 g_i 为一个 $(n+1,d,s)$-稀疏多项式，使得 $f_i|g_i$。令 $m=((n+1)ds)^2$。考虑多项式

$$f(x_0,G_{(m)})=x_0{}^k\prod_i f_i(G_{(m)})\,.$$

固定某个 i，并记 $f_i=\sum_{j=0}^{d}f_{i,j}x_0{}^j$ 和 $g_i=\sum_{j=0}^{d}g_{i,j}x_0{}^j$，其中 $f_{i,j},g_{i,j}\in\mathbb{F}[\mathbf{x}]$。由于 $x_0\nmid f_i$，可知对每个 i，都有 $f_{i,0}\neq 0$。因此，通过除去 x_0 的幂，我们可以假设 $x_0\nmid g_i$（注意，这样做不会改变 g_i 的稀疏度）。

由于 g_i 是一个 $(n+1,d,s)$-稀疏多项式，因此可知 $g_{i,0}$ 是 (n,s,d)-稀疏多项式。此外，对于每个 i，我们都有 $f_{i,0}|g_{i,0}$。因此，根据推论 3.5可知$f_{i,0}(G_{(m)})\neq 0$，从而有 $x_0\nmid f_i(G_{(m)})$。也就是说， x_0 作为 f 的因子的重数与其相同

[4]关于变量 x 的多项式的首项系数，是该多项式中非平凡出现的最高次幂 x 的系数。

作为 的因子的重数。后者是一个定义在常数个变量上的、具有有界个体次数的多项式 $\mathrm{poly}(n, d, s, \ell)$，我们可以使用引理 2.9 对其进行插值，并直接求出 k （这是使 x_0 整除 的所有单项式的最小幂次）。

对于“此外”部分，设 $f = x_0{}^k g$，并假设 $\boldsymbol{\alpha} = (\alpha_0, \alpha_1, \ldots, \alpha_n) \in \mathbb{F}^{n+1}$ 是我们要对其求值的点 g。考虑 $h(x) = f(x, \alpha_1, \ldots, \alpha_n) = x^k g(x, \alpha_1, \ldots, \alpha_n)$ 作为关于$\mathbb{F}[x]$的多项式。我们可以使用标准的单变量插值（如引理 2.9中所述），通过对 f的黑盒访问来恢复 $h(x)$，从而恢复多项式 $g(x, \alpha_1, \ldots, \alpha_n) = h(x)/x^k$。代入 $x = \alpha_0$ 后，便可恢复 $g(\boldsymbol{\alpha})$，正如我们所希望的。 □

推论 3.8 (寻找单项式因子)。 存在一个确定性的多项式(n, d, s, ℓ)时间算法，给定乘积 f 中 ℓ 个复数域 $\mathcal{P}_{Factors}(n, s, d) \subset \mathbb{F}[\boldsymbol{x}]$ 上多项式的黑盒访问，可返回 f 中 x_i 作为因子的重数 k_i。此外，若 M 是 f 的最大单项式因子，且满足 $f = Mg$，则我们可以通过对 f 的黑盒访问，获得对 g 的黑盒访问。

证明。由推论 3.7 可立即得到此结论。我们找出整除 f 的每个变量的最大幂。这便得到单项式 M。为了获得对 g 的黑盒访问，我们执行以下操作：给定一个点 $\boldsymbol{\alpha} \in \mathbb{F}^n$，如果 $\boldsymbol{\alpha}$ 的所有坐标都非零，那么我们可以计算 $g(\boldsymbol{\alpha}) = f(\boldsymbol{\alpha})/M(\boldsymbol{\alpha})$。如果 $\boldsymbol{\alpha}$ 的某些坐标为零，因而无法除以 $M(\boldsymbol{\alpha})$，则执行以下操作。考虑代换 $\boldsymbol{x} \leftarrow \boldsymbol{\alpha} + (t, \ldots, t)$，其中 t 是一个新变量。通过在超过 dn 个满足 $t \notin \{-\alpha_1, \ldots, -\alpha_n\}$ 的 t 取值处计算 $\tilde{g}_{\boldsymbol{\alpha}}(t) := (f/M)(\boldsymbol{\alpha} + (t, \ldots, t))$，我们可以对 $\tilde{g}(t)$ 进行插值。然后我们就可以计算 $g(\boldsymbol{\alpha}) = \tilde{g}_{\boldsymbol{\alpha}}(0)$。因此，对 g 的每次查询都需要对 f 进行 $O(dn)$ 次查询。 □

注 3.9. 前一个推论的第一部分是第 4 节中技术引理的一个简单特例。特别地，总能利用 f 的因式分解来确定不可约稀疏多项式作为此类 $f(x_0, G_{(m)})$的因子的重数。不过，正如我们将在下一节中所示，一般情况下需要更大的 m 值。

4 整除性测试与精确幂测试

本节将证明我们的整除性测试和精确幂测试结果。为此，我们首先需要证明关于生成器 G的一些技术结果。我们将这些技术结果分为两部分讨论。在第 4.1 节 中，我们得到一些对任意域均成立的结果；在第 4.2 节 中，我们证明这些结果的一个更强版本，但它仅在零特征或足够大的特征下成立。

4.1 任意特征结果

本小节基于 [BV25] 的方法，得到适用于任意特征域的整除性测试结果。为此，我们定义并研究多项式类的本原因子这一概念，并开发算法所需的工具；这些内容将在第 4.1.1 节中完成。在第 4.1.2 节 中，我们利用这些工具得到相应的算法。

4.1.1 任意特征工具

我们从本原因子的定义开始。

定义 4.1. 设 $\mathcal{P} \subset \mathbb{F}[\boldsymbol{x}]$ 是一个多项式类。若多项式 $g \in \mathbb{F}[\boldsymbol{x}]$ 满足以下两个条件，则称其为 $\mathcal{P}$ 的本原因子。

(1) 对于每个 $f \in \mathcal{P}$ 和每个 $t \in \mathbb{N}$，多项式最大公因式 $\gcd(f, g^t)$ 是 g 的幂。等价地，对于每个 $f \in \mathcal{P}$，存在 $k \geqslant 0$ 和一个与 g 互素的多项式 $\tilde{g}$，使得 $f = g^k \cdot \tilde{g}$。

(2) g 的任何非数乘倍数都不满足第一条件。

例 4.2。 不可约多项式 $f \in \mathcal{P}$ 显然是 $\mathcal{P}$的一个本原因子。

注 4.3。 $\mathcal{P}$ 的每个本原因子都整除某个多项式 $f \in \mathcal{P}$。事实上，假设 g是 $\mathcal{P}$ 的一个本原因子，但不整除 $\mathcal{P}$中的任何多项式。那么，对于每个 $f \in \mathcal{P}$，我们都有最大公因式$(f, g) = 1$，因此 g 与 $\mathcal{P}$中的每个元素都互素。由此可知，对于每个 $f \in \mathcal{P}$，都有最大公因式$(f, g^2) = 1$，所以 g^2 也满足第一条件。这与 g的第二条件矛盾。

注 4.4。 $\mathcal{P}$ 的本原因子不一定是 $\mathcal{P}$的元素。例如，考虑由某个非常数多项式 $g \in \mathbb{F}[x]$确定的多项式类 $\mathcal{P} = \{g^2, g^3\}$。那么， $\mathcal{P}$ 的唯一一个本原因子（在标量乘法意义下）是 g，而它不是 $\mathcal{P}$的元素。

在某种意义上，本原因子之于多项式类 $\mathcal{P}$，犹如不可约多项式之于所有多项式构成的类。为了使这一说法更加明确，我们证明每个 $f \in \mathcal{P}$ 都能唯一地写成若干本原因子的乘积。也就是说，$\mathcal{P}$ 的本原因子确实是 $\mathcal{P}$ 中多项式的“最小构件”。

为了证明这一点，我们先说明，对于多项式类 $\mathcal{P}$，任意两个不同的本原因子都互素。

引理 4.5. 设 $\mathcal{P} \subset \mathbb{F}[x]$ 是一个多项式类， g 和 h 是它的两个非相伴本原因子。那么 $\gcd(g, h) = 1$。

证明。 反设最大公因式 $(g, h) \neq 1$，且令 f_1，...， f_N 为它的不可约因子，其中 $N \geqslant 1$。写作：

$$g = \prod_{1 \leqslant i \leqslant N} f_i^{u_i} \cdot \prod_j g_j^{e_j}, \quad h = \prod_{1 \leqslant i \leqslant N} f_i^{v_i} \cdot \prod_j h_j^{r_j} \tag{13}$$

其中 f_i、g_j、 h_j 是互不相同的不可约多项式，而所有指数 u_i、 v_i、e_j、 r_j 都是正整数。

我们断言，向量 $(u_1, \ldots, u_N)$ 和 $(v_1, \ldots, v_N)$ 成比例。事实上，考虑某个$\phi \in \mathcal{P}$ 使得 $g|\phi$。那么，根据定义 4.1的第一条件，我们必有 k' 以及 t' 使得

$$\phi = g^{k'} \cdot \tilde{g}, \text{ where } \gcd(g, \tilde{g}) = 1 \text{ and}$$
$$\phi = h^{t'} \cdot \tilde{h}, \text{ where } \gcd(h, \tilde{h}) = 1 .$$

因此 ，通过考察 f_i 作为 ϕ 的因子的重数，我们必有 t

$$k'(u_1, \ldots, u_n) = t'(v_1, \ldots, v_n). \tag{14}$$

令 $(k, t) := (k'/\text{最大公因式}(k', t'),\ t'/\text{最大公因式}(k', t'))$。也就是说， (k, t) 是满足（14）的最小一对正整数。考虑由下式给出的多项式：

$$\tilde{\phi} = \prod_{1 \leqslant i \leqslant n} f_i^{ku_i} \cdot \prod_j g_j^{ke_j} \cdot \prod_j h_j^{tr_j}. \tag{15}$$

注意，由于 k， $t \geqslant 1$ （$k \geqslant 1$，因为 $g|\phi$，且 $t \geqslant 1$ 由（14）），ϕ 是 g 和 h的非平凡倍数。

我们断言， $\tilde{\phi}$ 满足定义 4.1的第一条件。事实上，设 $\psi \in \mathcal{P}$。由于 g和 h是本原因子，我们可以写成：

$$\psi = g^{k_0} \cdot \hat{g} \text{ where } \gcd(g, \hat{g}) = 1. \tag{16}$$
$$\psi = h^{t_0} \cdot \hat{h} \text{ where } \gcd(h, \hat{h}) = 1.$$

通过考察 f_i 作为 ψ的因子的重数，我们得到 $k_0(u_1, \ldots, u_N) = t_0(v_1, \ldots, v_N)$。由于$(k, t)$ 是满足 (14) 的正整数最小整数对，因此对于某个非负整数 c，必有 $k_0 = ck$ 和 $t_0 = ct$。现在，将这些信息代回 (16)，不难验证，对于某个与 $\tilde{\phi}$互素的多项式 $\hat{}\,\psi$ ，有 $\psi = \tilde{\phi}^c \cdot \hat{}\,\psi$ 。这证明 $\tilde{\phi}$ 满足定义 4.1 的第一条件。然而，这与 g 和 h 满足其第二条件这一事实矛盾。这证明 g 和 h 不能共享不可约因子，并完成了证明。

□

接下来我们陈述一个引理，以表达这样一种直觉：本原因子所起的作用类似于不可约多项式。

引理 4.6. 设 $\mathcal{P}$ 是某一类多项式，且设 $f \in \mathcal{P}$。那么， f 可以唯一地写成关于 $\mathcal{P}$的本原因子的乘积。

证明。令 $f \in \mathcal{P}$。考虑 f的某个不可约因子 f_0。显然， f_0 满足定义 4.1 的第一条件。令 g_0 为 f 的一个极大因子，且它被 f_0整除并满足定义 4.1 的第一条件。我们断言， g_0 必定是一个本原因子。为此，我们必须证明它满足定义 4.1 的第二条件。反设存在某个 h，它是 g_0的非平凡倍数，并满足定义 4.1 的第一条件。由 g_0的极大性可知， $h \nmid f$。由第一条件可得最大公因式$(f, h) = 1$。然而，由于 $f_0 \mid g_0$ 且 $g_0 \mid h$，我们得到 f_0|最大公因式(f, h)，矛盾。

因此，我们找到了某个本原因子 $g_0|f$。根据定义，我们可以写成 $f = g_0^k \tilde{g}_0$ ，其中最大公因式 $(g_0, \tilde{g}_0) = 1$。若 $\tilde{g}_0 = 1$ ，则得证。否则，固定一个不可约因子 $f_1|\tilde{g}_0$。按照前面的相同推理，存在 f_1的一个倍数，记作 g_1，它是 $\mathcal{P}$的一个本原因子，且整除 f。由引理 4.5，以及 f_1 整除 g_1 但不整除 g_0这一事实可知， g_0 与 g_1 必定互素。因此，我们可以写成 $f = g_0^{k_0} g_1^{k_1} \tilde{g}_1$，其中 $\tilde{g}_1$ 与 g_0 和 g_1均互素。如此继续下去，我们便可得到所需的 f的因式分解，即将其写成若干本原因子的乘积。

最后，为完成证明，我们注意到，由此得到的因式分解的唯一性可由引理 4.5得出。

□

因此，我们可以定义本原因子整除该类多项式时的重数。

定义 4.7。 设 $f \in \mathcal{P}$，且设 g 是 $\mathcal{P}$的一个本原因子。 g 作为f的因子时的重数，记作 $\nu_g(f)$，是 f 分解为本原因子的因式分解中 g 的幂次。

接下来，我们讨论本原因子定义与 [BV25] 的非退化条件之间的联系。下一个引理的证明由 [BV25，引理 4.5,4。6] 得出。

引理 *4.8*（[*BV25*，引理 *4.5*、*4.6*]）。设 ϕ 和 ψ是两个非相伴不可约多项式。固定某个 i，使得 $x_i \in var(\phi)$。假设存在 s-稀疏多项式 f和 g，其有界个体次数为 d，且满足 $\phi|f$ 和 $\psi|g$。记 $\nu_\phi(f)$ 为 ϕ 作为 f的因子的重数；对于ψ 和 g，定义类似。假设矩阵 $\begin{pmatrix} \nu_\phi(f) & \nu_\phi(g) \\ \nu_\psi(f) & \nu_\psi(g) \end{pmatrix}$ 可逆。那么，对于某个显式的 $m =$，多项式 $\phi(G_{(m,i)})$和$\psi(G_{(m,i)})$ 作为 u 中的多项式互素 $poly(n, (d^2)!, s^{d^3})$。

引理的条件可以用本原因子来更好地理解，以下断言对此作了说明。

命题 4.9. 设 $\mathcal{P}$ 是一个多项式类，并设 ϕ 和 ψ 是两个不可约多项式，它们分别整除 $\mathcal{P}$中的某个多项式（不一定是同一个多项式）。则以下命题等价。

(1) ϕ 和 ψ 整除 $\mathcal{P}$的同一个本原因子。
(2) 对每个 f， $g \in \mathcal{P}$，矩阵 $\begin{pmatrix} \nu_\phi(f) & \nu_\phi(g) \\ \nu_\psi(f) & \nu_\psi(g) \end{pmatrix}$ 都不可逆。

证明. 首先注意，“被 ϕ 整除的本原因子”这一说法定义明确，因为根据引理 4.5，这样的本原因子至多有一个（相差一个标量倍数），而由 ϕ 整除 $\mathcal{P}$中的某个多项式可知，至少存在一个这样的因子；显然， ψ也同样如此。

对于蕴含（(1)） $\implies$ (2)），注意，若 h 是 ϕ 和 ψ共同整除的本原因子，根据 引理 4.5 和 引理 4.6 可知，对每个 $f \in \mathcal{P}$，都有 $(\nu_\phi(f), \nu_\psi(f)) = \nu_h(f)(\nu_\phi(h), \nu_\psi(h))$。因此，显然，对于每一对 f， $g \in \mathcal{P}$，向量 $(\nu_\phi(f), \nu_\psi(f))$和 $(\nu_\phi(g), \nu_\psi(g))$ 成比例，所以以这些向量为列的矩阵不可逆。

接下来的断言证明了蕴含关系（(2)） $\implies$ （(1)）。

命题 4.10。 设 h_1 和 h_2 是类 $\mathcal{P}$的两个非相伴本原因子。那么，存在多项式f_1， $f_2 \in \mathcal{P}$，使得矩阵 $\begin{pmatrix} \nu_{h_1}(f_1) & \nu_{h_1}(f_2) \\ \nu_{h_2}(f_1) & \nu_{h_2}(f_2) \end{pmatrix}$ 可逆。

证明。 考虑向量集合 $(\nu_{h_1}(f), \nu_{h_2}(f))_{f \in \mathcal{P}}$。要证明该断言，只需证明这个集合中存在两个线性无关向量。反设并非如此。那么，该集合的 $\mathbb{Z}$-张成空间是一个一维格，由某个向量 (k, t)生成。

我们断言，$h := h_1^k h_2^t$ 必须满足定义 4.1 的第一条件。事实上，取某个元素 $\tilde{f} \in \mathcal{P}$。由引理 4.6，可将其写为 $\tilde{f} = h_1^{\nu_{h_1}(f)} h_2^{\nu_{h_2}(f)} g$，其中某个多项式 g与 h_1 和 h_2均互素。根据 (k, t)的定义，存在某个非负整数 c，使得$(\nu_{h_1}(f), \nu_{h_2}(f)) = c(k, t)$。因此，$\tilde{f} = h_1^{ck} h_2^{ct} g = h^c g$，其中 g与 h互素。也就是说， h确实满足定义 4.1 的第一条件。由于 h_1 是本原因子， $\nu_{h_1}(\tilde{f}) \geqslant 1$ 对某个$\tilde{f} \in \mathcal{P}$ 成立，因此必有 $k \geqslant 1$。类似地，我们得到 $t \geqslant 1$。也就是说， h 是 h_1 的非平凡倍数，而 h_2 满足定义 4.1 的第一条件，矛盾。证明至此完成。

□

注意到命题 4.10意味着 $\neg$ （（1）） $\implies$ $\neg$（（2））。事实上，设 h_1 和 h_2 分别为使 ϕ 和 ψ 整除的本原因子。假设 h_1 和 h_2 不相伴。根据命题 4.10，存在 f_1、 $f_2 \in \mathcal{P}$ 使得矩阵 $\begin{pmatrix} \nu_{h_1}(f_1) & \nu_{h_1}(f_2) \\ \nu_{h_2}(f_1) & \nu_{h_2}(f_2) \end{pmatrix}$ 可逆。由整除性可得

$$\begin{pmatrix} \nu_{h_1}(f_1) & \nu_{h_1}(f_2) \\ \nu_{h_2}(f_1) & \nu_{h_2}(f_2) \end{pmatrix} = \begin{pmatrix} \frac{\nu_\phi(f_1)}{\nu_\phi(h_1)} & \frac{\nu_\phi(f_2)}{\nu_\phi(h_1)} \\ \frac{\nu_\psi(f_1)}{\nu_\psi(h_2)} & \frac{\nu_\psi(f_2)}{\nu_\psi(h_2)} \end{pmatrix}$$

因此

$$\det \begin{pmatrix} \nu_\phi(f_1) & \nu_\phi(f_2) \\ \nu_\psi(f_1) & \nu_\psi(f_2) \end{pmatrix} = \nu_\phi(h_1)\nu_\psi(h_2) \det \begin{pmatrix} \nu_{h_1}(f_1) & \nu_{h_1}(f_2) \\ \nu_{h_2}(f_1) & \nu_{h_2}(f_2) \end{pmatrix} \neq 0.$$

因此，矩阵 $\begin{pmatrix} \nu_\phi(f_1) & \nu_\phi(f_2) \\ \nu_\psi(f_1) & \nu_\psi(f_2) \end{pmatrix}$ 可逆。这就完成了证明。

□

命题 4.9 以一种更自然的方式解释了 引理 4.8中的条件：如果重数矩阵是退化的，那么 ϕ 和 ψ 对 $\mathcal{P}(n,s,d)$的同一个本原因子作除子，因此从这个类的角度来看，它们是“不可分的”。这促使我们为 引理 4.8 提出一个适用于 $\mathcal{P}(n,s,d)$的任意两个本原因子的类似命题。

引理 4.11。 设 ϕ 和 ψ 是 $\mathcal{P}(n,s,d)$的两个非相伴本原因子。固定某个 $1 \leqslant i \leqslant n$，使得 $x_i \in$ 变量集(ϕ)。那么，对于某个显式值 $m =$ 多项式$(n,(d^2)!\ ,\ s^{d^3})$ 有 $\phi(G_{(m,i)})$ 与 $\psi(G_{(m,i)})$互素，其中将二者都视为关于 u的多项式。

证明将与引理 4.8的证明几乎完全相同，后者刻画了 [BV25，引理 4。6]，主要区别在于引理 4.8涉及不可约多项式，而引理 4.11考察本原因子。为完整起见，我们对此给出一个概要。

证明。 在此证明中，每当我们提到本原因子时，指的都是$\mathcal{P}(n,s,d)$的本原因子。

根据 命题 4.9，存在 $f_1, f_2 \in \mathcal{P}(n,s,d)$，使得矩阵

$$\begin{pmatrix} k_1 & t_1 \\ k_2 & t_2 \end{pmatrix} := \begin{pmatrix} \nu_\phi(f_1) & \nu_\phi(f_2) \\ \nu_\psi(f_1) & \nu_\psi(f_2) \end{pmatrix}$$

可逆。令 $h_1 = f_1^{t_1+t_2}$和 $h_2 = f_2^{k_1+k_2}$。经过简单计算可得，交换 h_1和 h_2后，有 $h_1 =$最大公因式$(h_1,h_2)\phi^a H_1$且 $h_2 =$最大公因式$(h_1,h_2)\psi^b H_2$，其中 $a,b \geqslant 1$和 H_1, H_2与 ϕ,ψ互素（由唯一分解和本原因子的性质可知）。

考虑 $M_{x_i}(h_1,h_2)$的西尔维斯特矩阵。由于 f_1 和 f_2 的个体次数均有界于 d，因此 k_1+k_2，$m_1+m_2 \leqslant 2d$。所以， h_1和 h_2是 s^{2d}-稀疏多项式，其个体次数有界于 $2d^2$。因此，根据 引理 2.20，$M_{x_i}(h_1,h_2)$的子式的行列式是 $(4d^2)!\ s^{8d^3}$-稀疏多项式，个体次数至多为 $8d^4$。 引理 3.4 意味着，对于

$$m = (n(8d^4)((4d^2)!s^{8d^3}))^2 = \mathrm{poly}(n,(d^2)!,s^{d^3})$$

与 $G_{(m,i)}$ 复合时，子式的行列式均不为零。因此可得

$$\mathrm{rank}(M_{x_i}(h_1,h_2)) = \mathrm{rank}(M_u(h_1(G_{(m,i)}),h_2(G_{(m,i)})))\,.$$

引理 2.20((4)) 因而保证 $\deg_{x_i}$(最大公因式$(h_1,h_2)) = \deg_u$(最大公因式$(h_1(G_{(m,i)}),h_2(G_{(m,i)})))$。因此，与 $G_{(m,i)}$ 复合并未改变 u中的次数。因此， $\phi(G_{(m,i)})$ 和$\psi(G_{(m,i)})$ 作为 u中的多项式互素，否则它们的公因子会使最大公因式的次数增加。

至此，证明的概要就结束了。我们指出，在引理 4.15中，我们证明了一个类似的断言，并更详细地解释了上述论证中的技术要点。当然， [BV25，引理 4.5,4。6]中也有解释。

□

接下来我们陈述一个重要的推论。

推论 4.12. 设 $\phi \in \mathcal{P}(n,s,d)$ 是一个不可约多项式，且 $\psi \in \mathcal{P}_{Factors}(n,s,d)$ 是某个 $g \in \mathcal{P}(n,s,d)$的一个不可约因子，它与 ϕ不相伴。固定某个 $1 \leqslant i \leqslant n$，使得 $x_i \in var(\phi)$。那么，对于一个显式的 $m =$ 多项式$(n,(d^2)!\ \ s^{d^3})$，将 $\phi(G_{(m,i)})$和 $\psi(G_{(m,i)})$视为关于 u的多项式时，它们互素。此外，如果 ϕ 是多二次的，我们可以取 $m =$多项式$(n,(d^2)!\ \ s^{d^2})$。

证明。直观上，这应当可由 引理 4.11 推出（回想一下，$\mathcal{P}(n,s,d)$ 中的一个不可约多项式也是其本原因子，见 例 4.2）。严格来说，我们手头有一个不可约多项式 ψ，它不一定是本原因子（因为它可能不属于 $\mathcal{P}(n,s,d)$）。我们采用与 命题 4.9的证明类似的方法来解决这个问题。我们改为˜考虑本原因子 ψ（在相差一个非零标量的意义下唯一），它ψ整除。显然，ϕ˜ ˜与 ψ 非相伴，因此只需将 引理 4.11 应用于 ϕ 和 ψ，即可得出该断言。

对于“此外”部分，我们注意到，引理 4.11 的证明实际上给出了更好的结果，当 ϕ <style id='8}>为多二次时。我们说明应如何修改引理 4.11的证明以看出这一点；从现在起，我们使用该论证中的记号。

在这种情况下，由于 $\phi \in \mathcal{P}(n,s,d)$ 是不可约的，可以取 $f_1 = \phi$，并令 f_2 为某个被 $\tilde{\psi}$ 整除的多项式。因此，我们有 $k_1 = 1$、$k_2 = 0$。由此可得：

- $h_1 = f_1^{t_1+t_2} = \phi^{t_1+t_2}$ 作为关于 x_i 的多项式，其次数至多为 $2(t_1+t_2) \leqslant 4d$，而不是 $2d^2$。其稀疏度至多为 s^{2d}。
- $h_2 = f_2^{k_1+k_2} = f_2$ 作为关于 x_i 的多项式，其次数至多为 d，而不是 $2d^2$。其稀疏度至多为 s。

因此，$M_{x_i}(h_1,h_2)$ 的任意子式的行列式都是一个 $s^{O(d^2)}$-稀疏多项式，其有界个体次数至多为 $O(d^2)$。这意味着在这种情况下，取$m = \mathrm{poly}(n,(d^2)!,s^{d^2})$ 就足够了。

□

4.1.2 任意特征算法

我们的下一个目标是为有界个体次数的稀疏多项式乘积得到一个整除性测试结果。为此，我们需要下面这个引理，它是整除性测试算法的核心。在证明有理插值结果时，这个引理将在第 5 节发挥关键作用。

引理 *4.13.* 存在一个确定性的 $poly(n,(d^2)!,s^{d^3},\ell)$-时间算法，给定多项式乘积 f（由 ℓ (n,s,d)-稀疏多项式构成）的黑盒访问，以及 $\mathcal{P}(n,s,d)$的本原因子 ϕ 的黑盒访问，该算法返回 f 中 ϕ 作为因子的重数（定义 *4.7* 所述的意义下）。事实上，该算法只需知道生成器 $G_{(m)}$的像上 f 的取值，其中某个明确的取值为 $m = poly(n,(d^2)!,s^{d^3})$。此外，如果 ϕ 是一个不可约的 s-稀疏多项式，那么当 f 是$\mathcal{P}_{Factors}(n,s,d)$ 中的多项式乘积（而不是 $\mathcal{P}(n,s,d)$）时，该算法仍然适用；若进一步假设 ϕ 是多二次的，则取 $m = poly(n,(d^2)!,s^{d^2})$便足够，且运行时间可降至 $poly(n,(d^2)!,s^{d^2},\ell)$。

证明。固定一个 i使得 $x_i \in$变量集(ϕ)，并令 $m =$多项式$(n,(d^2)!,s^{d^3})$从而引理 4.11 的结论成立。推论 3.6 意味着 $\deg_u(\phi(G_{(m,i)})) = \deg_{x_i}(\phi) \geqslant 1$。

根据 引理 4.6 我们可以写成 $f = \phi^k \cdot \prod \psi_j$，其中 ψ_j 是 $\mathcal{P}(n,s,d)$ 的本原因子，且不与 ϕ相伴。由 引理 4.11可知，对每个 ψ_j，都有 $\phi(G_{(m,i)})$ 与 $\psi_j(G_{(m,i)})$互素（将二者视为关于 u的多项式时）。特别地，$\gcd_u(\phi(G_{(m,i)}),\prod\psi_j(G_{(m,i)})) = 1$。由于 $\deg_u(\phi(G_{(m,i)})) \geqslant 1$，因此整除 $\phi(G_{(m,i)})$的 f 最高次幂，作为关于 u 的多项式，恰为 k。

因此，为了恢复所需的重数，我们将对每个 $1 \leqslant i \leqslant n$计算 $\phi(G_{(m,i)})$ 作为 f 中的多项式所能整除的最高次幂，并返回其中的最小值。该幂可以通过使用 $\phi(G_{(m,i)})$ 和 $f(G_{(m,i)})$ 进行插值，结合 引理 2.9，并计算 $\phi(G_{(m,i)})$ 整除 $f(G_{(m,i)})$ 作为 u 中的多项式时的最高次幂来得到（只需进行常规多项式

除法，因此这里不需要进行多项式因式分解）。注意，两个多项式的次数至多为 $\mathrm{poly}(n, d, s, m, \ell)$，因此关于运行时间的断言成立。至此，算法的描述结束。

至于“此外”部分，只需考虑不可约多项式的因式分解（而非本原因子），并将引理 4.11替换为推论 4.12，即可由同一证明得出。

□

我们终于可以给出整除性测试结果了。这是对 [Vol17中的整除性测试结果（定理 7]）的推广。

定理 4.14。 存在一个确定性 $\mathrm{poly}(n, (d^2)!, s^{d^3}, \ell)$-时间算法，该算法通过黑盒访问两个 ℓ (n, s, d)-稀疏多项式的乘积 f, g，判断 $f|g$ 是否成立。此外，若 f 的所有不可约因子实际上都是稀疏的（即 f 是不可约 (n, s, d)-稀疏多项式的乘积），则即使 g 是 $\mathcal{P}(n, s, d)$ 中的多项式（而非 $\mathcal{P}_{Factors}(n, s, d)$）的乘积，该算法仍然适用；若进一步假设 f 的所有因子都是多二次的，则该算法的运行时间可缩短至 $\mathrm{poly}(n, (d^2)!, s^{d^2}, \ell)$。

证明。固定某个 $m = $多项式$(n, (d^2)!\ ,\ s^{d^3})$ 使得引理 4.13 适用。对每一个 $1 \leqslant i \leqslant n$ 检查 $f(G_{(m,i)})$是否整除 $g(G_{(m,i)})$，并且若所有这些检查都通过，则返回 $f|g$。显然，若 $f|g$我们返回正确答案。现在假设 $f \nmid g$。由引理 4.6，我们知道存在一个本原因子 ϕ，它作为 f 的因子的重数超过它在 g 中的重数。然后，由引理 4.13 的证明可知，对每一个使得 $x_i \in$ 变量集(ϕ) 的 i，使 $\phi(G_{(m,i)})^k$ 整除 $f(G_{(m,i)})$ 的最高幂 k实际上等于 ϕ 作为 f 的因子的重数，对 $g(G_{(m,i)})$ 亦类似。特别地，对于 i 的这些取值，不可能出现 $f(G_{(m,i)})|g(G_{(m,i)})$，因此我们确实会返回 $f \nmid g$。

至于“此外”部分，在考虑将 f 和 g 因式分解为不可约多项式（而不是分解为本原因子），并使用 引理4.13的“此外”部分时，可由相同的证明得到。

□

正如我们在本小节中所看到的， [BV25] 方法对于处理有关单项次数有界的稀疏多项式乘积的因式分解问题非常强大。实际上，我们将在第 5 节 中看到，引理 4.13 正是我们得到有理插值结果所需要的工具，并因此也是我们从单项次数有界的稀疏多项式乘积中恢复多二次稀疏不可约因子所需算法的基础。

然而，这种方法存在一些局限性。例如，考虑判断给定的单项次数有界的稀疏多项式乘积是否为完全幂的问题。一个自然的做法是应用一种类似于定理 4.14 中所述的方法——检查$f(G_{(m,i)})$ 对于每个 i是否为完全幂。然而，这种算法并不奏效。为了解释原因，假设我们的输入多项式是形如 $\phi = \phi_1\phi_2$ 的本原因子，其中 ϕ_1和 ϕ_2 是两个非相伴不可约多项式。本小节的方法无法说明 $\phi_1(G_{(m,i)})$ 与 $\phi_2(G_{(m,i)})$之间的相互作用。特别地，可能对每个 i，都有$\phi_1(G_{(m,i)}) = \phi_2(G_{(m,i)})$，从而 $\phi(G_{(m,i)})$ 是一个完全平方。

下一小节的目标是将本小节的结果从仅处理本原因子推广到处理不可约多项式。例如，这使我们能够使所提出的完全幂判定得以实现。此外，它使我们能够将所有结果从稀疏多项式的乘积这一类推广到稀疏多项式的因子的乘积这一类。

然而，我们的推广仅适用于特征零域，或当特征大于个体次数界的两倍时。特别地，对于小特征域，本小节的结果已是我们所能得到的最好结果。

4.2 零特征或大特征结果

在本小节中，我们证明了第 4.1 节结果的更强版本，无论是在一般性还是运行时间方面。然而，本小节的结果仅当 $\mathrm{char}(\mathbb{F}) = 0$或当 $\mathrm{char}(\mathbb{F}) > 2d$ 时成立（其中 d是个体次数的界）。

与 第 4.1 节类似，在第 4.2.1 节中，我们构建算法中将用到的工具，而在 第 4.2.2 节中，我们给出算法。

4.2.1 零特征或大特征工具

我们需要的主要技术工具是对 引理 4.11 和 引理 4.13 的推广，将其应用于不可约因子，而非本原因子。我们证明以下与引理 4.11类似的结果。

引理 4.15. 设 $\mathbb{F}$ 是一个特征为零或大于 $2d$ 的域。设 ϕ，ψ 是 $\mathcal{P}_{Factors}(n, s, d)$ 中的非相伴不可约多项式，分别整除 (n, s, d)-稀疏多项式 f 和 g（后者可能与 f相同）。固定某个满足 $x_i \in var(\phi)$的 i。那么，对于某个明确值$m = \mathrm{poly}(n, d!\ s^d)$，当视为关于 u的多项式时，$\phi(G_{(m,i)})$与 $\psi(G_{(m,i)})$互素。此外，$\phi(G_{(m,i)})$ 和 $\psi(G_{(m,i)})$ 作为关于 u的多项式，均无平方因子。

证明。 固定某个 i，使得 $x_i \in \mathrm{var}(\phi)$。从现在起，将所有多项式都视为关于 x_i 的多项式。考虑 $M_{x_i}(fg, (fg)')$，即 fg 及其关于 x_i 的形式导数的西尔维斯特矩阵。令$fg = \prod_j h_j^{e_j}$为 fg 的不可约因子因式分解（视为 $\mathbb{F}(\boldsymbol{x} \setminus x_i)$上的关于 x_i 的多项式）。

断言 4.16。 令 $\mathbb{F}$ 如 引理 *4.15*所述。令 $f \in \mathbb{F}[x]$ 为一个次数至多为$2d$的非恒定一元多项式。假设 $f = \prod_i h_i^{e_i}$是 f 的不可约多项式因式分解。那么 维数(核 $(M(f, f'))$ =$\sum_i$次数$(h_i)(e_i - 1)$。

证明。显然，$f' = \sum_i e_i h_i^{e_i - 1} h_i' \hat{h}_i$，其中我们用 $\hat{h}_i$ 表示 f 中不同于 h_i 的因子的乘积。由于 h_i 是一个次数至多为 $2d$ 的非恒定多项式，关于 $\mathrm{char}(\mathbb{F})$ 的条件蕴含$h_i' \neq 0$。由于 h_i 不可约，它必与 h_i' 互素。此外，由于 h_i 和 $\hat{h}_i$ 显然互素，且 $e_i \neq 0$ （根据关于 $\mathrm{char}(\mathbb{F})$ 的条件），因此 h_i 作为 f' 的因子的重数恰为 $e_i - 1$。由于 h_i 是 f 的唯一不可约因子，可得 $\gcd(f, f') = \prod_i h_i^{e_i - 1}$。于是由引理 2.20 得 $\dim(\ker(M(f, f')) = \deg(\gcd(f, f')) = \deg(\prod_i h_i^{e_i - 1}) = \sum_i \deg(h_i)(e_i - 1)$，如所断言。 □

继续证明，显然有 $\deg_{x_i}(fg) \leqslant 2d$。因此，$r := (4d-1)\times(4d-1) - \dim\ker(M_{x_i}(fg, (fg)')) = \sum_j \deg_{x_i}(h_j)(e_j - 1)$ 由断言 4.16 。由于 $M_{x_i}(fg, (fg)')$ 是一个矩阵，它包含一个可逆的 $(4d-1-r)\times(4d-1-r)$ 子式 M。由于 fg 和 $(fg)'$ 是 s^2-稀疏的，由引理 2.20 可知，M 的元素是个体次数为 $2d$ 的 s^2-稀疏多项式，因此 $\det M$ 是一个个体次数为 $8d^2$ 的 $(4d)!s^{8d}$-稀疏多项式。因此，推论 3.5 意味着，对于 $m = (n(8d^2)((4d)!s^{8d}))^2 = \mathrm{poly}(n, d!, s^d)$，有 $\det M(G_{(m,i)}) \neq 0$ （其中，除第 i 个坐标外，将所有坐标都代换为 $G_{(m,i)}$ ）。注意，$\det M(G_{(m,i)})$ 恰好是矩阵$M_u(fg(G_{(m,i)}), (fg')(G_{(m,i)}))$的一个 $(4d-1-r)\times(4d-1-r)$ 子式的行列式，因此可得

$$\dim(\ker\left(M_u(fg(G_{(m,i)}), (fg)'(G_{(m,i)}))\right) \leqslant r. \tag{17}$$

现在，我们知道 $fg(G_{(m,i)}) = \prod_j h_j(G_{(m,i)})^{e_j}$；令 $fg(G_{(m,i)}) = \prod_k \tilde{h}_k^{r_k}$为 $fg(G_{(m,i)})$ 的不可约因子因式分解。我们现在提出如下断言。

命题 4.17。

$$\sum_k \deg \tilde{h}_k(r_k - 1) \geqslant \sum_j \deg h_j(G_{(m,i)})(e_j - 1)$$

且等号成立　　当且仅当集合 $\{h_j(G_{(m,i)})\}$ 是无平方因子、两两互素的多项式集合时　　*s.*

证明。考虑分解为不可约多项式的因式分解，我们可以写成 $h_j(G_{(m,i)}) = \prod_k \tilde{h}^{e_{j,k}}$。现在，对于每个 k，都有 $\sum_j e_j e_{j,k} = r_k$。因此：

$$\begin{aligned}\sum_k \deg \tilde{h}_k(r_k - 1) &= \sum_k \deg \tilde{h}_k(\sum_j e_j e_{j,k} - 1) \\ &\geqslant \sum_k \deg \tilde{h}_k(\sum_j e_{j,k}(e_j - 1)) = \sum_j (e_j - 1) \sum_k e_{j,k} \deg \tilde{h}_k \\ &= \sum_j \deg h_j(G_{(m,i)})(e_j - 1).\end{aligned}$$

这就证明了该不等式。注意，当且仅当对每个 k，都有 $\sum_j e_{j,k} = 1$ 时，等号成立。但显然，当且仅当集合 $\{h_j(G_{(m,i)})\}$ 是一个由无平方因子、两两互素的多项式组成的集合时，这一条件成立，正如所断言的那样。 □

因此，由 命题 4.17、断言 4.16 以及（17），我们可推出：

$$\begin{aligned}r &\geqslant \dim(\ker\left(M_u(fg(G_{(m,i)}), (fg(G_{(m,i)}))'\right)) = \sum_k \deg \tilde{h}_k(r_k - 1) \\ &\geqslant \sum_j \deg h_j(G_{(m,i)})(e_j - 1) = \sum_j \deg h_j(e_j - 1) = r\end{aligned}$$

这里我们用到了 $\deg h_j(G_{(m,i)}) = \deg h_j$，其中次数在左侧是相对于 u 而言，而在右侧是相对于 x_i 而言。实际上，注意到将其视为 $\mathbb{F}[\boldsymbol{x}]$中的一个元素时，有 $h_j \in \mathcal{P}_{\text{Factors}}(n, s^2, 2d)$，因此该等式由 推论 3.6得出。

因此可知，我们处于命题 4.17的等号成立情形，所以集合 $\{h_j(G_{(m,i)})\}$是由无平方因子且两两互素的多项式组成的集合。证明至此结束，因为根据定义，$\{h_j\}$ 包含 ϕ。若 ψ 也在其中，则结论成立。否则，当 $x_i \notin$ 变量集(ψ)时，断言显然成立。 □

注 4.18。事实上，即使 $d <$ 特征$(\mathbb{F}) \leqslant 2d$，我们也可以利用第 4.1 节 的结果得出一些结论：考虑本原因子 h_1，h_2 它们被 ϕ，ψ 整除。分两种情况：

- 若 h_1 和 h_2 不是相伴元，则该断言由引理 4.11推出；但现在我们必须取 $m =$ 多项式$(n, (d^2)!\, s^{d^3})$。
- 若 $h_1 = h_2$，取某个 $\tilde{h} \in \mathcal{P}(n, s, d)$ 使得 $h_1 = h_2|\tilde{h}$。然后，在上述证明中，我们可以用多项式 $\tilde{h}$ 替换多项式 fg，从而得到关于 $m =$ 多项式$(n, d!\ s^d)$的结果。注意，现在 $\deg_{x_i}(\tilde{h}) \leqslant d$，因此我们确实只需要 $\text{char}(\mathbb{F}) \geqslant d + 1$（或 $\text{char}(\mathbb{F}) = 0$）。

4.2.2 零特征或大特征算法

根据引理 4.15，我们可以得到如下对引理 4.13的推广。

引理 4.19. 设 $\mathbb{F}$是特征零的域，或特征大于 2d的域。存在一个确定性多项式$(n, d!,\ s^d,\ \ell)$时间算法：给定$\mathcal{P}_{Factors}(n, d, s)$ 中 ℓ 个多项式的乘积 f的黑盒访问，以及 $\mathcal{P}_{Factors}(n, s, d)$中不可约多项式 ϕ 的黑盒访问，二者均定义在 $\mathbb{F}$上，该算法返回 ϕ 作为 f的因子的重数。事实上，该算法只需知道 f 和 ϕ在生成器 $G_{(m)}$ 的像上的值，该生成器对应于某个显式的 $m = $多项式$(n, d!,\ s^d)$。

证明。 令 $m = $多项式$(n, d!\ \ s^d)$ 根据 引理 4.15的要求。对于每个 $1 \leqslant i \leqslant n$，利用 引理 2.9，我们可以恢复 $\phi(G_{(m,i)})$ 和 $f(G_{(m,i)})$。因此，利用一元多项式的常规除法，我们可以找到最大的 k，使得对于每个 i，都有 $\phi(G_{(m,i)})^k | f(G_{(m,i)})$ 作为 u中的多项式成立。

记 t 为 ϕ 作为 f 的因子的重数。我们断言 $k = t$。事实上，对于每个 i，都有 $\phi(G_{(m,i)})^k | f(G_{(m,i)})$，因此显然 $t \leqslant k$。为证明另一方向的不等式，取满足 $x_i \in \mathrm{var}(\phi)$ 的某个i，并写作 $f = \phi^k \cdot \prod_j \psi_j$，其中 ψ_j 都是不可约的，且与 ϕ 非相伴。于是，根据引理 4.15， $\phi(G_{(m,i)})$ 与 $\psi_j(G_{(m,i)})$ 互素，对任意 j 均如此（作为u中的多项式），因此 $\phi(G_{(m,i)}) \nmid \prod_j \psi_j(G_{(m,i)})$。注意，这里我们用到了 $x_i \in \mathrm{var}(\phi)$，因为在这种情况下，根据推论 3.6，我们知道 $\deg_u(\phi(G_{(m,i)})) = \deg_{x_i}(\phi) \geqslant 1$。因此， $\phi(G_{(m,i)})$ 恰好整除$f(G_{(m,i)})$ 次（作为 u 中的多项式），引理得证。 □

如同上一小节所做的那样，我们可以利用引理 4.19的证明，得到一个整除性测试结果。由于本小节的结果适用于稀疏多项式的一般不可约因子，因此所得的整除性测试算法不仅适用于稀疏多项式的乘积，也适用于稀疏多项式的因子的乘积。相比之下，第 4.1.2 节中的结果仅适用于由 $\mathcal{P}(n, s, d)$中的多项式构成的乘积。

定理 4.20。设 $\mathbb{F}$ 是一个特征为零或大于 2d的域。存在一个确定性$\mathrm{poly}(n, d!, s^d, \ell)$-时间算法，它通过对多项式 f和 g的黑盒访问来判定 $f|g$是否成立，其中二者各自都是 $\mathcal{P}_{Factors}(n, s, d)$中的 ℓ 个多项式的乘积。

证明。固定某个满足引理 4.19 结论的 $m = \mathrm{poly}(n, d!, s^d)$。对于每个 $1 \leqslant i \leqslant n$，检查 $f(G_{(m,i)})$ 是否整除 $g(G_{(m,i)})$（将它们视为 u中的多项式；在整个论证中，我们都以此方式看待它们），并且仅当所有这些检查都通过时，才返回 $f|g$。注意，这样做不需要多项式因式分解；常规的一元多项式除法实际上就足够了。显然，如果 $f|g$，我们会返回正确答案。现在假设 $f \nmid g$；令 ϕ 为一个不可约因子，它作为 f 的因子的重数大于其作为 g的因子的重数。于是，由引理 4.19 的证明可知，对于每个满足 $x_i \in$ 的 i （其中 (ϕ)为变量集），使得 $\phi(G_{(m,i)})^{k_1}$ 整除$f(G_{(m,i)})$ 以及 $\phi(G_{(m,i)})^{k_2}$ 整除 $g(G_{(m,i)})$的最高次幂 k_1和 k_2，实际上分别等于 ϕ 作为 f和 g的因子的重数。特别地，对于这些 i 值，不可能有 $f(G_{(m,i)}) | g(G_{(m,i)})$，因此我们确实会返回 $f \nmid g$。 □

现在我们可以证明定理 1.16。为方便起见，我们重述其内容。

定理 1.16。 设 $\mathbb{F}$ 是特征为零或大于 2d 的域。存在一个确定性多项式$(n, d!, s^d, \ell)$-时间算法：给定两个多项式（在 $\mathbb{F}$ 上） f 和 g 的黑盒访问，其中它们是 ℓ 个 $\mathcal{P}_{Factors}(n, s, d)$ 中多项式的乘积，判断 $f|g$ 是否成立。此外，若 f 和 g 实际上是 $\mathcal{P}(n, s, d)$ 中 ℓ 个多项式的乘积，则存在一个适用于任意域的算法，其运行时间为多项式$(n, (d^2)!, s^{d^3}, \ell)$

证明。 证明可由结合定理 4.14 与定理 4.20得出。 □

仔细考察定理 4.20的证明，我们能够得到定理 1.15，该定理推广了 [BV25]的一个结果。正如上一小节末尾所述，我们目前还不知道一个适用于任意域的版本，因为我们需要充分利用引理 4.15。

定理 1.15。 设 $\mathbb{F}$ 是一个特征零或大于等于 $2d$的域。存在一个确定性的多项式$(n, d$时间算法， s^d该算法 ℓ)在给定一个多项式 f 的黑盒访问时，其中该多项式是 $\mathcal{P}_{Factors}(n, s, d)$ 上（定义在 $\mathbb{F}$上的） ℓ 个多项式的乘积，并给定一个正整数 e，如果 f 是一个完整的 e次幂，则返回相应结果。

证明。和之前一样，固定一个足够大的 $m =$ 多项式$(n, d!\ s^d)$，使得引理 4.15 成立。对于每个 $1 \leqslant i \leqslant n$，使用 $f(G_{(m,i)})$计算其因式分解，利用引理 2.9 和定理 2.8。根据这些因式分解，很容易检查每个 $1 \leqslant i \leqslant n$，判断 $f(G_{(m,i)})$是否为完全的 e次幂。事实上，一个多项式是 e次幂，当且仅当它的每个不可约因子整除该多项式的重数都能被 e整除，且它的首项系数是 e中的一个 $\mathbb{F}$次幂。对于 $f(G_{(m,i)})$，前一个条件可以根据其不可约因子分解轻松检查。至于后一个条件，则可以直接对每个多项式 $h \neq 0$进行检查。例如：

- 若 $\mathbb{F} = \mathbb{F}_q$ 对某个素数幂 q成立，则检查 h 的首项系数 α 是否满足 $\alpha^{\frac{q-1}{e}} = 1$。
- 若 $\mathbb{F} = \mathbb{Q}$，则使用定理 2.6 来检查 $x^e - \alpha$ 是否有有理根，其中 α 是 h的首项系数。

我们将返回：如果对每个 $1 \leqslant i \leqslant n$， $f(G_{(m,i)})$ 都是一个完全 e 次幂，那么 f 就是一个完全 e 次幂。显然，如果 f 是一个完全 e 次幂，该算法就会返回正确答案。因此，假设f 不是一个完全 e 次幂。那么，根据上面的讨论，以下情况必有一种成立。

- 存在一个不可约因子 $\phi|f$，它在 f 中作为因子的重数是某个整数 k，且该整数不能被 e整除。取一个满足 $x_i \in \mathrm{var}(\phi)$的值 i，并取 $\phi(G_{(m,i)})$的一个不可约因子 $\tilde{\phi}$。根据推论 3.6，我们注意到， u 中存在这样的正次数不可约因子。我们断言， $\tilde{\phi}$ 作为 $f(G_{(m,i)})$ 的因子的重数为 k。事实上，根据引理 4.15，对于每个不与ϕ相伴的不可约 $\psi|f$，它都与 $\psi(G_{(m,i)})$ 互素。此外，它恰好整除 $\phi(G_{(m,i)})$ 一次（根据引理中“此外”部分）。因此， $\tilde{\phi}$ 是 $f(G_{(m,i)})$ 的一个不可约因子，且恰好整除它 k 次，所以 $f(G_{(m,i)})$不是一个完全 e次幂。因此，我们确实会返回 f 不是一个 e次幂。
- $f = \alpha g^e$ 对于某些 $g \in \mathbb{F}[x]$ 和 $\alpha \in \mathbb{F}$，使得 α不是 e 次幂（在 $\mathbb{F}$中）。在这种情况下，对于每个 $1 \leqslant i \leqslant n$，都有 $f(G_{(m,i)}) = \alpha g(G_{(m,i)})^e$ 也不是 e次幂，因此我们确实会返回 f 不是 e次幂。

证明至此完成。 □

5 有理插值定理

本节将证明我们的有理插值结果。

我们陈述有理插值结果。该结果需要大量关于所涉及函数的信息，但在我们的应用中，我们会说明如何获得这些信息。

定理 5.1. 假设我们可以黑盒访问某个形式为 $\frac{a}{b}f$ 的有理函数，该函数定义在域 $\mathbb{F}$ 上，其中：

(1) a 是一个 (n, s, d)稀疏多项式。

(2) 我们可以黑盒访问 f，它是 $\mathcal{P}_{Factors}(n, s,$中的 ℓ 个多项式的乘积 $d)$。 *(3)* 我们有一个 $\mathcal{F}$ 的集合，其中包含 s-稀疏不可约因子 f，将它们称为 ϕ_i。

(4) b 是 f 的一个 s-稀疏因子，其不可约因子全部属于 $\mathcal{F}$。

(5) a 与 b 互素。

那么，存在一种算法，通过在某个显式的 m （取决于 特征$(\mathbb{F})$）的生成器 $G_{(m)}$的像上对 $\frac{af}{b}$ 进行求值，来恢复 a 与 b，其运行时间为 多项式(m, ℓ)，其中：

- 如果 特征$(\mathbb{F}) = 0$ 或 特征$(\mathbb{F}) > 2d$，则可以取 $m =$ 多项式$(n, d!\ s^d)$。
- 在任意域上，$m =$ 多项式$(n, (d^2)!\ s^{d^3})$。如果进一步假设 $\mathcal{F}$ 的所有元素都是多重二次的，那么可以取 $m =$ 多项式$(n, (d^2)!\ s^{d^2})$。

证明。 首先假设域的特征为零或大于 $2d$。固定某个 m =多项式$(n, d!\ s^d)$，使得 引理 4.19 的结论成立。我们首先恢复 b。由于 b 的所有不可约因子都属于 $\mathcal{F}$，因此只需理解每个 $\phi \in \mathcal{F}$ 作为 b的因子的重数。固定这样一个不可约因子 $\phi \in \mathcal{F}$。利用 引理 4.19，我们可以计算ϕ 作为 f的因子的重数，将其记为 t。

由于 a 和 b 互素，因此对于任意 k，$\phi^k | b$ 当且仅当 $\phi^{t-k+1} \nmid \frac{af}{b}$。所以，只需了解 ϕ 作为 $\frac{af}{b}$ 的因子的重数。由于 f 是s-稀疏多项式的因子的乘积，$b | f$，且$\frac{f}{b}$是 s-稀疏的，因此$\frac{af}{b}$ 也是 s-稀疏多项式的因子的乘积。因此，我们可以应用引理 4.19 来确定 ϕ 作为 $\frac{af}{b}$ 的因子的重数。由此可知，我们可以将 b 因式分解为不可约多项式。特别地，我们可以利用引理 3.4 显式地恢复 b。要确定 a，只需计算 $a(G_{(m)}) = f(G_{(m)})/b(G_{(m)})$ 并应用引理 3.4。

对于小特征域以及多二次情形，我们重复相同的论证，将 引理4.19替换为 引理 4.13的“此外”部分。 □

接下来，我们陈述上述定理的一个轻微推广，稍后将会用到。

定理 5.2. 假设我们对一组定义在域 $\mathbb{F}$ 上、形如 $\frac{a_j f^j}{b}$，对于 $1 \leqslant j \leqslant D$ 的有理函数具有黑盒访问，其中：

(1) 对每一个 $1 \leqslant j \leqslant D$，$a_j$ 都是一个 (n, s, d)-稀疏多项式。

(2) f 是 $\mathcal{P}_{Factors}(n, s, d)$ 中 ℓ 个多项式的乘积，且我们对这些多项式具有黑盒访问。

(3) 我们有一个集合 $\mathcal{F}$，由 f 的 s-不可约稀疏因子组成，称它们为 ϕ_i。

(4) b 是 f 的一个 s-稀疏因子，其不可约因子全部是 $\mathcal{F}$ 的元素。

(5) $(a_1,$的最大公因式$\ldots\ldots,\ a_D,\ b) = 1.$

那么，存在一种算法，可以根据生成器$G_{(m)}$的像上的 $\frac{a_j f^j}{b}$ 的求值，恢复a_js和 b，其中某个显式的 m （取决于特征$(\mathbb{F})$），使得运行时间为多项式(m, ℓ, D)，其中：

- 如果特征$(\mathbb{F}) = 0$ 或特征$(\mathbb{F}) > 2d$，则可以取 $m =$ 多项式$(n, d!,\ s^d,\ D)$。
- 对于任意域，我们有 $m =$ 多项式$(n, (d^2)!\ s^{d^3},\ D)$。如果进一步假设 $\mathcal{F}$的所有元素都是多二次的，那么可以取 $m =$ 多项式$(n, (d^2)!\ s^{d^2},\ D)$。

证明。 该证明与定理 5.1的证明思路相同。唯一的区别在于，现在在恢复 b时，并不必然有最大公因式$(a_j, b) = 1$成立。为克服这一障碍，注意到对每一个 $\phi \in \mathcal{F}$，其重数为 t、作为 f的因子，对每一个 $k \geqslant 1$，都有$\phi^k | b$ 当且仅当存在 $1 \leqslant j \leqslant D$ 使得 $\phi^{tj-k+1} \nmid \frac{a_j f^j}{b}$。这一点可由与先前相同的推理得出。事实上，存在某个 $1 \leqslant j \leqslant D$ 使得 $\phi \nmid a_j$，否则 $\phi | $最大公因式$(a_1, \ldots, a_D,\ b)$。随后该算法以与先前完全相同的方式进行。 □

6 元算法

在本节中，我们给出一个元算法；该元算法将在若干变体下用于定理1.9、1.10、1.11、1.12、1.13 以及1.14的证明，这些定理陈述了我们的算法结果。由于定理 1.10是定理 1.11的推论，我们在此不予讨论。

这些算法均遵循元算法的框架；在其证明中，我们将引用元算法的各个步骤，并说明它们在相应设定下是如何实现的。

(1) **设定：**

We 假设我们想了解多项式 f 的性质，使得 f 要么是 r

(1) (n, s, d)-稀疏（定理 1.9，定理 1.12）（2）变量集为 $\mathcal{P}_{\text{Factors}}(n, s, d)$（定理 1.11，定理 1.13，定理 1.14 ）的多项式乘积

并且我们希望找到列表 $\mathcal{F}(f)$ ，即 f 的具有某些性质的因子。对于每个变量x_i，多项式 $f|_{x_i=0}$ 也属于与 f 相同的类型。当 f 为 (n, s, d)-稀疏时，这一点显然成立。当 f 是变量集 $\mathcal{P}_{\text{Factors}}(n, s, d)$ 中的多项式乘积时，这可由 引理 6.2 得出，其表述与证明在本节末尾给出。

在下文中，为方便起见，我们假设 f 是一个 $n+1$元多项式，并且是 $\mathbb{F}[x_0, \mathbf{x}]$中的一个元素。

(2) **预处理：**

（1）从 f 中去掉其最大单项式因子（使用推论 3.8）。（2）将 f 归一化，得到 $\tilde{f}(y, \mathbf{x}) = f(yf_0, x_1, \ldots, x_n)/f_0$。针对适当的生成器 G设置$\hat{}\ f = f(y, G)$。注意到 $\tilde{f}(y, \mathbf{x})$ 关于 y是反向首一的，因而$\hat{}f$也是如此。由断言 2.3 可知，若 f 是 (n, s, d)-稀疏的，则$\hat{}f$ 至多有$\deg_y(\hat{}f) \leqslant d$ 个不可约因子；若 f 是 $\mathcal{P}_{\text{Factors}}(n, s, d)$中 ℓ 个多项式的乘积，则至多有$\deg_y(\hat{}f) \leqslant \ell d$ 个。（3）使用引理 2.9 将$\hat{}f$ 插值为含 $O(1)$ 个变量的多项式。然后应用定理 2.8，得到其分解为不可约因子的因式分解。

(3) 递归调用以求解 f_0**：** 对 $f_0 = f|_{x_0=0}$递归求解该问题，得到一个列表$\mathcal{F}(f_0)$，其中包含根据该问题不可被单项式整除的因子或不可约因子。

(4) **生成候选解列表：**

根据命题 2.5，每个因子 $h = \sum_{i=0}^{d} c_i x_0^i$ （属于 f ）都会产生一个因子$\sum_{i=0}^{d} \frac{c_i}{c_0} f_0^i y^i$ （属于 $\tilde{f}$），从而产生一个因子$\hat{}h = \sum_{i=0}^{d} \frac{c_i(G)}{c_0(G)} f_0^i y^i$ （属于$\hat{}f$）。我们希望获得 $c_i(G)$，以便对它们进行插值，并由此获得 h。这使我们面临有理插值问题，其中我们拥有关于 c_0 （甚至 c_0 本身）的一些信息，这些信息来自$\mathcal{F}(f_0)$。

(1) 在定理 1.9、定理 1.11 中，我们将有 $c_0 \in \mathcal{F}(f_0)$。在定理 1.12 c_0 是元素的乘积，这些元素来自 $\mathcal{F}(f_0)$，但我们有足够的时间枚举所有相关乘积，因此可以假定 $c_0 \in \mathcal{F}(f_0)$。

事实上，上述说法并不完全准确。递归调用会返回 f_0 的所有不被单项式整除的因子的集合$\mathcal{F}(f_0)$，以及整除 f_0的最大单项式。 c_0 可能是 f_0 的一个因子，而 f_0 可被单项式整除。在这种情况下， c_0 不会属于 $\mathcal{F}(f_0)$。不过， c_0 确实属于 $\mathcal{F}(f_0)$，相差一个单项式因子。正如我们将看到的，这对我们的目的而言已经足够。

(2) 在定理 1.13和定理 1.14中，情况略有不同。在这里， c_0 可能是 $\mathcal{F}(f_0)$中的若干元素的乘积，而我们无法遍历所有可能的方式来得到 c_0。不过，在这种情况下， $\mathcal{F}(f_0)$ 将包含 c_0的所有不可约因子，这正是定理 5.2所要求的。

因此，对于每个因子 $\hat{h} = \sum_{i=0}^{d} e_i y^i$，我们按如下方式构造一个列表 $\mathcal{F}_{\hat{h}}$ ：

(1) 当我们得到一个列表，其中包含（相差一个单项式乘法的）正确 c_0(定理 1.9、定理 1.11、定理 1.12)时，我们遍历该列表中的所有可能候选项c_0 。对于每个这样的候选项，我们对形如$\big((x_1 \cdots x_n)^d c_0/f_0^i\big)(G)\, e_i$ 的多项式进行插值，以恢复系数 c_i。然后，将$\mathcal{F}_{\hat{h}}$ 那些在除以其最大单项式因子后 $\sum_{i=0}^{d} c_i x_0^i$ 符合问题要求的多项式添加到列表中：

(1) 在定理 1.9、定理 1.11 中，我们保留所有 (n, s, d)-稀疏多项式。
(2) 在定理 1.12 中，我们保留所有 (n, S, d)-稀疏多项式，其中 $S = s^{O(d \log^2 n)}$ 是定理 2.14 中得到的 (n, s, d)-稀疏多项式因子稀疏度的上界定理 2.14。

项 $(x_1 \cdots x_n)^d$ 是为了处理如下事实所必需的： c_0 属于 $\mathcal{F}(f_0)$ 仅在除以一个单项式的意义下成立。我们将在定理 1.9的证明中详尽说明这一点。

(2) 如果我们只有 c_0 的不可约因子列表（定理 1.13 和 定理 1.14），则应用定理 5.2 于 $\hat{h}$ 的系数，以恢复可能的 c_0 和 c_1，……， c_d。我们将所得多项式 $\sum_{i=0}^{d} c_i x_0^i$ 放入列表 $\mathcal{F}_{\hat{h}}$，如果：

(1) 对于 定理 1.13 ，它是 (n, S, d)-稀疏的。
(2) 对于 定理 1.14，它是多二次的且 (n, s, d)-稀疏的。

我们注意到，在这两种情形下，我们为 $\hat{h} = 1$ 设定 $\mathcal{F}_{\hat{h}} = \mathcal{F}(f_0)$ 。

(5) **剪枝列表：**

根据应用需求，我们需要保留因子或不可约因子。我们分两步进行：首先，在候选项列表中找出因子；然后，在需要时，从中识别出不可约多项式。这两个步骤的实现方式因算法而异；这里我们给出一种适用于所有情况的方案，尽管它并非总是最优，并在各算法的证明中提供更好的实现。

在所有情况下，我们都应用 定理 1.16 ，从列表中移除那些不能整除 f 的多项式。这对于定理 1.9 和定理 1.11 已经足够。

为了获得 定理 1.12、定理 1.13、定理 1.14 所要求的不可约因子，我们将 定理 1.16 应用于因子列表中的任意两个多项式 g、h，以检查 g 是否整除 h，并且只保留那些不能被其他任何因子整除的因子。

(6) **计算重数：**

在需要计算重数时，我们应用引理 4.19 或 引理 4.13，具体取决于特征。在这里，某些情况下也有更好的解法；这些解法在算法的证明中有所描述。

算法运行时间的分析取决于具体的设置。不过，在所有设置中，我们都恰好进行一次递归调用，因此总体复杂度由算法各个步骤的开销决定，不会因递归调用而急剧增加。

注 6.1。 所述元算法给出了我们所有算法所遵循的框架。不过，它并不是一个能够涵盖所有这些算法的算法。我们可以给出两个具体算法，所有结果都遵循这两个算法：一个推广定理 1.9、1.11 和 1.12，另一个推广定理 1.13 和1.14。不过，这些通用算法的精确表述相当复杂，也不够直观，因此我们决定不在本文中列出。

我们通过证明以下引理来结束本节。

引理 6.2. 设 f 是 $\mathcal{P}_{Factors}(n+1,s,d) \subset \mathbb{F}[x_0,\boldsymbol{x}]$ 中 ℓ 个多项式的乘积。进一步假设 $x_0 \nmid f$，并设 f_0 是 f 作为关于 x_0 的多项式时的常数项。那么， f_0 是 $\mathcal{P}_{Factors}(n,s,d) \subset \mathbb{F}[\boldsymbol{x}]$ 中 ℓ 个元素的乘积。

证明. 写作 $f = \prod_{i=1}^{\ell} g_i$，其中 $g_i \in \mathcal{P}_{\mathrm{Factors}}(n+1,s,d)$。进一步记 $g_i = \sum_{j=0}^{d} g_{i,j} x_0^j$，其中$g_{i,j} \in \mathbb{F}[\boldsymbol{x}]$，并设 $\hat{g}_i$ 是一个 $(n+1,s,d)$-稀疏多项式，且 g_i 整除它。首先，注意到 $f_0 = \prod_{i=1}^{\ell} g_{i,0}$。接着，观察到 $g_{i,0}$ 整除 $\hat{g}_i$ 作为关于 x_0 的多项式时的末项单项式。由于这个末项单项式显然是 (n,s,d)- 稀疏的，因此可得 $g_{i,0} \in \mathcal{P}_{\mathrm{Factors}}(n,s,d)$。于是， f_0 确实是 $\mathcal{P}_{\mathrm{Factors}}(n,s,d)$ 中 ℓ 个多项式的乘积。

□

7 寻找多二次因子

在本节中，我们证明定理 1.14。

定理 1.14. 设 $\mathbb{F}$ 是特征为零或大于 $2d$的域。存在一个确定性$\mathrm{poly}(n, d!, s^d, \ell)$时间算法：给定对乘积 f 的黑盒访问，该乘积由 ℓ 个 $\mathcal{P}_{Factors}(n,s,d)$中定义在 $\mathbb{F}$上的元素组成，算法返回所有 s稀疏多二次不可约因子，即 f的此类因子。特别地，该算法返回 f 的所有多线性因子（这些因子是 s稀疏的，由引理 *2.12*可知）。此外，存在一种适用于任意域的此类算法，其运行时间为 $\mathrm{poly}(n, (d^2)!, s^{d^2}, \ell)$。

证明。 我们遵循第 6 节中所述元算法的提纲。由于这是我们首次实现该元算法，我们将给出实现的全部细节。我们先描述特征$(\mathbb{F}) = 0$ 或特征$(\mathbb{F}) > 2d$情形下的算法，然后说明在任意域上工作时哪些部分会发生变化。

为简化表述，我们假设输入 f 是 $\mathbb{F}[x_0,\boldsymbol{x}]$中的一个元素，并且是 ℓ 个 $\mathcal{P}_{\mathrm{Factors}}(n+1,s,d)$中元素的乘积。如 第6(2)节所述，我们先进行预处理。

预处理： 根据 推论 3.8 ，我们可以假设 f 不能被任何单项式整除。特 $x_0 \nmid f$别地，我们可以假设 $f = \sum_{i=0}^{\ell d} f_i x_0^i$，其中 $f_i \in \mathbb{F}[\boldsymbol{x}]$。令 $\tilde{f}$ 为 $\tilde{}$的归一化形式 f ，如 定义 2.4所述。根据 断言 2.5 f 是反向首一的，并且我们可以黑盒访问它。记

$$\hat{f} := \tilde{f}(y, G),$$

对于 $G = G_{(m)}$，其中 m 如定理 5.2所给。利用引理 2.9和定理 2.8，得到 $\hat{}f$的因式分解。注意，由于 $\tilde{f}$ 关于 y是反向首一的，因此 $\hat{}f$也是如此。因此，根据断言 2.3，$\hat{}f$ 至多有 ℓd 个不可约因子。因此，它总共至多有 $O((\ell d)^2)$ 个次数至多为二的（不一定不可约）因子，且这些因子属于 y 。注意，$\hat{}f$ 是一个关于 $O(1)$ 个变量、次数为 $\mathrm{poly}(m,\ell)$的多项式，因此这一步的运行时间至多为 $\mathrm{poly}(m,\ell)$。

递归调用：我们希望先解决 f_0的问题。根据引理 6.2， f_0 也是由 ℓ 个 $\mathcal{P}_{\text{Factors}}(n, s, d)$中的元素相乘得到的。因此，我们可以通过一次递归调用，找出它的所有多二次、 s-不可约因子构成的集合。将此集合记为 $\mathcal{F}$。

生成候选解列表： 回想一下，命题 2.5 表明，对于每个不可约因子 $\phi = b + a_1 x_0 + a_2 x_0^2$， f都有一个对应的不可约因子 $\tilde{\phi} = 1 + \frac{a_1 f_0}{b} y + \frac{a_2 f_0^2}{b} y^2$，它是$\tilde{f}$的因子，并对应于$\hat{}f$的某个（不一定不可约的）二次因子，我们将其记为$\hat{}\phi = \tilde{\phi}(y, G)$。我们的计划是从$\hat{}f$ 和 $\mathcal{F}$的因式分解中，找出与不同因子对应的 (a_1, a_2, b)。

为此，我们遍历$\hat{}\,\phi$ 的所有因子$\hat{}f$，这些因子作为 y的多项式时次数至多为二。对于每个这样的$\hat{}\,\phi$，我们运行定理 5.2 中的算法，尝试恢复对应的 a_1、 a_2 和 b （或(a_1, b)，如果 $\deg_y(\hat{}\,\phi) = 1$）。如果 $b + a_1 x_0 + a_2 x_0^2$ 是一个 s-稀疏多二次多项式，那么我们就将其加入候选因子列表。

断言 7.1。 沿用上述记号，假设 $\phi = b + a_1 x_0 + a_2 x_0^2$ 对应于 f的一个多二次（或多线性）不可约因子。那么，在对 $\hat{}\,\phi$执行上述过程时，该过程会将 ϕ 添加到候选多二次因子列表中。

证明。 我们需要证明，所有必要信息都已具备，可以成功运行定理 5.2 。具体来说：

(1) 由于 ϕ 是 s-稀疏的，因此 a_1、 a_2 也都是 s-稀疏的。(2) f_0 是 $\mathcal{P}_{\text{Factors}}(n, s, d)$ 中 ℓ 个元素的乘积，我们可以对这些元素进行黑盒访问。(3) 通过递归调用，我们得到一个集合 $\mathcal{F}$，其中包含 f_0 的所有多二次、 s-稀疏不可约因子。(4) 由于 ϕ 是多二次的， b也如此。因此，根据引理 2.13， b 的所有不可约因子也都是s-稀疏且多二次的。由于 $b | f_0$，这些因子必定是 $\mathcal{F}$中的元素。(5) 根据 ϕ 的不可约性，我们得到最大公因式$(a_1, a_2, b) = 1$。

因此， 定理 5.2 将返回 a_1， a_2 和 b。 □

因此，遍历 y 中 $\hat{}f$ 的所有次数至多为 2 的因子，就能确保将 f 的所有不可约多二次因子都纳入候选项列表。由于 f 可能含有不依赖于 x_0 的多二次因子，我们还将 $\mathcal{F}$ 的元素加入 f 的因子候选项列表。

下一步是找出所有“真正的”因子。实际上，我们的候选项列表中也可能包含“虚假”候选项。例如，这些候选项可能来自对某个因子 ϕ 成功进行插值的过程，而该因子在某个变量 x_i 中的次数高于二；这一性质在复合、因式分解和插值过程中并未得到保留。

剪枝列表：现在我们已经得到一个集合，其中包含 f 的 $O((\ell d)^2)$ 个因子候选项，它们全都是多二次且 s-稀疏的；接下来需要从中识别出真正的不可约因子，并计算其重数。

通过运行 定理 1.16 中的算法，我们可以判断每个候选项 $\phi \in \mathcal{F}$是否确实整除 f。现在我们有一组候选项，它们全都是多二次且 s-稀疏的，并且都整除 f，接下来需要找出其中的不可约项。

引理 2.13 表明，由于每个因子 ϕ 都是多二次且 s-稀疏的，因此它的所有因子也都如此。特别地，它能被 f的某个多二次不可约因子整除，而该因子必定在列表中。因此，对于每一对 ϕ，ψ，我们检查 $\psi|\phi$ 是否成立（使用 定理 1.16）。对于那些不存在其他候选项整除它们的候选项 ϕ，它们就是所需的不可约因子。

计算它们的重数：我们已经找到了所有多二次的 s-稀疏不可约因子，它们是 f的因子。要计算它们作为 f的因子的重数，我们调用引理 4.19。

任意域：对于任意域，我们令 m 如定理 5.2中针对多二次多项式情形的设定。我们不使用定理 1.16，而是应用定理 4.14中“此外”部分。类似地，我们不使用引理 4.19，而是依赖引理 4.13中的“此外”部分。

运行时间：注意，对于特征零或大于 $2d$的域，每一步的运行时间均为 $\mathrm{poly}(n, d!\ s^d,\ \ell)$。这是定理 5.2、定理 1.16和引理 4.19所保证的运行时间。由于只有一次递归调用，总运行时间也为 $\mathrm{poly}(n, d!\ s^d,\ \ell)$。对于任意域，运行时间为 $\mathrm{poly}(n, (d^2)!\ s^{d^2},\ \ell)$，这是根据定理 5.2、定理 4.14 的“此外”部分以及引理 4.13的“此外”部分得到的。

至此完成了该算法的描述与分析。 □

8 计算稀疏多项式的全部稀疏因子

在本节中，我们分析有界个体次数的稀疏多项式的稀疏因子集合，并给出构造该集合的多项式时间算法。随后，我们将这些方法用于对 [DST24] 中所提出的问题给出部分解答，该问题涉及从对其乘积的黑盒访问中恢复有界个体次数的稀疏不可约多项式（问题 1.4）。

8.1 Nu单项次数有界的稀疏多项式的因子个数

我们首先证明有界个体次数的稀疏多项式中不被任何单项式整除的因子个数的一个上界。

定理 1.8（针对 (n, s, d)-稀疏多项式的因子界）。设 $f \in \mathcal{P}(n, s, d)$。$f$ 的非单项式不可约因子个数（按重数计）至多为 $d\log s$。特别地，一个 s-稀疏多项式 f（具有有界个体次数 d）中不被任何单项式整除的因子个数至多为 $2^{d\log s} = s^d$。

证明。 除以一个单项式后，我们可以假设 f 不被任何 x_i整除。我们按以下迭代方式选取一个变量序列。首先，选取某个 $x_{i_1} \in$ 变量(f)。假设我们已经选取了 x_{i_1}，……，x_{i_k}，并考虑 f 分解为不可约多项式的因式分解。如果存在一个不可约因子 ϕ，其中不含迄今为止选取的任何变量，我们就选取某个$x_{i_{k+1}} \in$ 变量(ϕ)。如果不存在这样的因子，则该过程终止。

假设在此过程中我们依次选择了 k 变量，不妨将它们记为 x_1、x_2、……、x_k（即假设 $i_1 = 1$ 等）。所选变量的序列自然对应于 f的不可约因子的一个多重集序列。第一个多重集 I_1 是依赖于 x_1 的不可约因子组成的多重集（按重数计）；第二个多重集I_2 由不依赖于 x_1 但依赖于 x_2的不可约因子组成；第 j 个多重集 I_j 是由不依赖于 x_1、……、x_{j-1} 但依赖于 x_j的不可约因子组成。容易看出，$\{I_j\}_{1\leqslant j\leqslant k}$ 构成了 f的不可约因子多重集的一个划分。

由于 f的有界个体次数为 d，且 I_j 的每个因子都依赖于x_j，因此必有 $|I_j| \leqslant d$ 作为多重集。于是，f的不可约因子总数（按重数计）至多为 kd。因此，为证明我们的定理，只需证明 $k \leqslant \log s$。

为此，令 h_j 为 I_j 中所有多项式的乘积，从而有 $f = \prod_{j=1}^{k} h_j$。

断言 8.1。 记 $f_j = \prod_{t=k-j+1}^{k} h_t$。则， $2^j \leqslant |V(P_{f_j})|$，其中 P_{f_j} 是与 f_j 相对应的牛顿多面体，而 $V(P_{f_j})$是其顶点集合（如 定义 *2.16*中所定义）。

我们先完成定理的证明，然后再证明该断言。由该断言可知， $2^k \leqslant |V(P_{f_k})| = |V(P_f)| \leqslant s$，因此 $k \leqslant \log s$，这正是我们要证明的结论。注意，这里用到了 f 是 s-稀疏的这一事实；利用 引理 2.17，可得 $|V(P_f)| \leqslant s$。注意，由于 $d \log s$ 是因子数量（包括重数）的上界， f 的因子数至多为$2^{d \log s} = s^d$。

□

证明断言 *8.1*。 该证明通过对 j的归纳法。对于 $j = 1$，该断言由以下事实得出：$f_1 = h_k$ 不是单项式（因为我们已移除了 f 的所有单项式因子）。

对于归纳步骤，假设 $2^j \leqslant |V(P_{f_j})|$。我们将证明 $2^{j+1} \leqslant |V(P_{f_{j+1}})|$。为此，将 h_{k-j} 表示为关于x_{k-j}的多项式。也就是说， $h_{k-j} = \sum_{i=0}^{d_{k-j}} g_i x_{k-j}^i$ ，其中 $g_i \in \mathbb{F}[x_{k-j+1}, \ldots, x_n]$。由构造， $x_{k-j} \in$ 变量集(h_{k-j})，因此 $d_{k-j} =$次数$_{x_{k-j}}(h_{k-j}) > 0$。此外，由于 $x_{k-j} \nmid h_{k-j}$，必有 $g_0 \neq 0$。注意到 $f_{j+1} = f_j h_{k-j} = \sum_{i=0}^{d_{k-j}} f_j g_i x_{k-j}^i$，其中 $f_j g_i \in \mathbb{F}[x_{k-j+1}, \ldots, x_n]$。我们断言如下。

(1) 任意顶点 $u \in V(P_{f_j g_0}) \subset \mathbb{R}^{n+j-k}$ 都对应于 $V(P_{f_{j+1}})$ 的一个顶点 $(0, u) \in \mathbb{R}^{n+j-k+1}$。

(2) 任意顶点 $v \in V(P_{f_j g_{d_{k-j}}}) \subset \mathbb{R}^{n+j-k}$ 都对应于 $V(P_{f_{j+1}})$ 的一个顶点 $(d_{k-j}, v) \in \mathbb{R}^{n+j-k+1}$。

(3) 我们有 $|V(P_{f_j g_0})| \geqslant 2^j$ 和 $|V(P_{f_j g_{d_{k-j}}})| \geqslant 2^j$。

前两个断言显然成立，因为所描述的顶点集合恰好分别是第一坐标（即对应于 x_{k-j} 的指数的坐标）取最小值和最大值的顶点集合。对于第三个断言，注意到根据 引理 2.17和归纳假设，我们得到

$$|V(P_{f_j g_0})| \geqslant |V(P_{f_j})| \geqslant 2^j, \quad \text{and} \quad |V(P_{f_j g_{d_{k-j}}})| \geqslant |V(P_{f_j})| \geqslant 2^j .$$

因此，我们证明了 $V(P_{f_{j+1}})$中总共有至少 2^{j+1} 个元素，从而完成了归纳步骤的证明。

□

注 8.2。 显然，上述定理的界是紧的，例如考虑 $\mathbb{C}$上的 $f(x_1, \ldots, x_n) = \prod_{i=1}^{\log s}(x_i^d - 1)$。

作为推论，我们得到 (n, s, d)-多项式的因子乘积的稀疏因子数量的以下上界。

推论 8.3。 设 f 是 $\mathcal{P}_{Factors}(n, s, d)$ 中 ℓ 个元素的乘积。 f 的不被单项式整除的 (n, s, d)-稀疏因子的数量至多为

$$\sum_{i=0}^{d \log s} \binom{\ell d \log s}{i} \leqslant s^{d(2+\log \ell)}.$$

证明。不妨设 f 不被任何单项式整除。

设 $f = \prod_{i=1}^{\ell} \phi_i$，其中 $\phi_i \in \mathcal{P}_{\mathrm{Factors}}(n, s, d)$，并记 g_i 为一个 (n, s, d)-稀疏多项式，该多项式 ϕ_i 可整除。

根据定理 1.8，每个 g_i 至多有 $d \log s$ 个不可约因子（按照因式分解中的重数重复计算因子），因此 ϕ_i也必然如此。因此，f至多有 $\ell d \log s$个不可约因子，记为 $\{\psi_j\}$。注意，这是一个多重集，其中的因子可能根据它们在 f的因式分解中的重数多次出现。

我们断言，任意 (n, s, d)-稀疏因子 f 都是至多 $d \log s$ 个此类因子的乘积 ψ_j。事实上，f 的任意因子都是 ψ_js 的某个子集的乘积，并且根据定理 1.8，任何这样的 s-稀疏因子都不可能是超过 $d \log$ 个 s 的 ψ_js 的乘积。因此，s有界个体次数的 d -稀疏因子 f 的数量上界为

$$\sum_{i=0}^{d \log s} \binom{\ell d \log s}{i}. \tag{18}$$

现在，如果 $\ell = 1$，(18) 等于 $2^{d \log s} = s^d$。如果 $\ell \geqslant 2$，则 $\binom{\ell d \log s}{d \log s}$ 是 (18) 中最大的项，并且很容易看出，我们的表达式 (18) 的上界为 $d \log s \binom{\ell d \log s}{d \log s}$。无论哪种情况，相关因子的数量都不超过

$$\begin{aligned}
\max\left\{s^d, d \log s \binom{\ell d \log s}{d \log s}\right\} &\leqslant \max\left\{s^d, d \log s \left(\frac{e \ell d \log s}{d \log s}\right)^{d \log s}\right\} \\
&= \max\left\{s^d, d \log s (e\ell)^{d \log s}\right\} \\
&= (d \log s) s^{d(\log \ell + \log e)} \\
&\leqslant s^{d(2 + \log \ell)},
\end{aligned}$$

其中，最后一个不等式是由以下事实得出的：对于任意 $c > 0$和 $x \geqslant 1$，都有 $(ce) \ln x \leqslant x^c$；这一点可通过注意到 $\ln \frac{x}{x^c}$ 在 $x = e^{1/c}$处取最大值来验证。将此不等式用于$c = d(2 - \log e)$，并利用 $e \ln 2(2 - \log e) > 1$ 这一事实，即可得到所述结果。 □

8.2 寻找有界个体次数稀疏多项式的稀疏因子

在本节中，我们证明定理 1.9。下面回顾其表述。

定理 1.9 （寻找所有 (n, s, d)-稀疏多项式的稀疏因子）。给定，存在一个确定性的poly$(n, d!\, s^d)$-时间算法，输出 f 的所有 s-稀疏因子（这些因子不被任何单项式整除），以及整除 f 的次数最大的单项式。

证明。 与定理 1.14的证明一样，我们遵循第 6 节中元算法的步骤，首先研究满足 $\mathrm{char}(\mathbb{F}) = 0$或 $\mathrm{char}(\mathbb{F}) > 2d$的域，然后研究任意域。为便于表述，我们假设 $f \in \mathcal{P}(n + 1, s, d) \subset \mathbb{F}[x_0, \boldsymbol{x}]$。

预处理： 此步骤按照 定理 1.14 的证明执行。我们首先求出 f 的单项式因子。因此，假设 $x_0 \nmid f$。记 $f = \sum_{i=0}^{d} f_i x_0^i$，其中 $f_i \in \mathbb{F}[\boldsymbol{x}]$，并注意到$f_0 \neq 0$ 由我们的假设可知。此外，$f_0 \in \mathcal{P}(n, s, d)$。令 $\tilde{f}$ 表示 f 和 $\hat{f} := f(y, G)$的归一化，其中 $G = G_{(m)}$ 满足 $m = (2nsd)^2$，如 引理 3.4 对 $(n, s, 2d)$-稀疏多项式所给出。利用 引理 2.9 和 定理 2.8，得到 $\hat{f} = f(y, G)$ 的不可约

多项式因式分解。由于 $\tilde{f}$ 关于 y是反向首一的，因此 $\hat{}f$也是如此。根据 断言 2.3，可知 $\hat{}f$ 至多有 d 个不可约因子，因此总共至多有 2^d 个因子。由于 $\hat{}f$ 含有 $O(1)$ 个变量，次数为 $\mathrm{poly}(m)$，因此此步骤的运行时间为 $\mathrm{poly}(m)$。

递归调用： 由于 f_0 显然是一个 (n, s, d)-稀疏多项式，我们可以通过进行一次递归调用，找到它的所有 s-稀疏因子的集合。将此集合记为 $\mathcal{F}(f_0)$。根据 定理 1.8，我们有 $|\mathcal{F}(f_0)| \leqslant s^d$。

生成并筛选候选解列表： 根据 命题 2.5，每个因子 $h = \sum_{i=0}^{d} c_i x_0^i$ 都对应于 f 的一个因子 $\tilde{h} = \sum_{i=0}^{d} \frac{c_i}{c_0} f_0{}^i y^i$，并因此对应于 $\tilde{f}$的一个因子 —

$$\hat{h} := \tilde{h}(y, G) = \sum_{i=0}^{d} \frac{c_i(G)}{c_0(G)} f_0(G)^i y^i$$

f的因子。与之前一样，我们遍历 f的所有因子$\hat{}h = \sum_{i=0}^{d} e_i y^i$。对于每个这样的因子，我们按如下方式生成一个至多包含 $|\mathcal{F}(f_0)|$ 个候选项的列表$\mathcal{F}_{\hat{h}}$，这些候选项是 f 的 s-稀疏因子：对于每个 $c \in \mathcal{F}(f_0)$，令 $\hat{e}_i = \left((x_1 \cdot \ldots \cdot x_n)^d c\right)(G) \frac{e_i}{f_0^i(G)^\circ}$ 利用引理 3.4，我们将 $\hat{e}_0$、$\hat{e}_1$，$\ldots$，$\hat{e}_d$ 进行稀疏插值，得到 $(n, s, 2d)$-多项式 $\tilde{e}_0$、$\tilde{e}_1$，$\ldots$，$\tilde{e}_d$。如果插值因任何原因失败，则转到下一个$c \in \mathcal{F}(f_0)$。如果插值成功，则计算可整除多项式 $\tilde{e}_0$、$\tilde{e}_1$，$\ldots$ $\tilde{e}_d$中每一个的最大单项式 M。如果 $h = \frac{\sum_{i=0}^{d} \tilde{e}_i x_0^i}{M}$是一个 (n, s, d)-稀疏多项式，则运行定理 1.16检查它是否整除 f；若是，则将其加入 $\mathcal{F}_{\hat{h}}$。

我们令 $\mathcal{F}'(f_0)$ 为 $\mathcal{F}(f_0)$ 中整除 f 的元素（同样，使用 定理 1.16 进行验证）。
我们返回集合 $\mathcal{F}(f) = \mathcal{F}'(f_0) \cup \left(\bigcup_{h \mid f} \mathcal{F}_{\hat{h}}\right)$，它是算法所生成的所有集合的并集。

断言 8.4。 $\mathcal{F}(f)$ 是 f的所有因子 $h = \sum_{i=0}^{d} c_i x_0^i$ 构成的集合。

证明。 由于 $\mathcal{F}(f)$ 只包含 f的因子，我们需要证明它包含 f的所有因子。

回忆一下， f 的每个因子 $h = \sum_{i=0}^{d} c_i x_0^i$ 都会产生f的一个因子 $\hat{}h = \sum_{i=0}^{d} \frac{c_i(G)}{c_0(G)} f_0^i(G) y^i$。此外，$c_0$是一个满足 $c_0 \mid f_0$的 (n, s, d)-稀疏多项式。因此，若 M 是整除 c_0 的最大单项式，那么对于某个 $c \in \mathcal{F}$，有 $c_0 = Mc$。

记 $e_i = \frac{c_i(G)}{c_0(G)} f_0^i(G)$，使得 $\hat{}h = \sum_{i=0}^{d} e_i y^i$。注意，对于 $\tilde{M} = (x_1 \cdot \ldots \cdot x_n)^d / M$，我们有

$$\hat{e}_i = \left((x_1 \cdot \ldots \cdot x_n)^d c\right)(G) \frac{e_i}{f_0^i(G)} = \tilde{M}(G) \cdot \frac{c_0(G) \frac{c_i(G)}{c_0(G)} f_0^i(G)}{f_0^i(G)} = \tilde{M}(G) c_i(G).$$

由于 c_i 是一个 (n, s, d)-稀疏多项式，且 $\tilde{M}$ 是一个个体次数有界的单项式，界为d， $\tilde{M} c_i$ 是一个 $(n, s, 2d)$-稀疏多项式。因此，引理 3.4 在作用于 $\hat{e}_i$时，将返回多项式 $\tilde{M} c_i$。因此，我们会将 $h = \sum_{i=0}^{d} c_i x_0^i$ 加入 $\mathcal{F}_{\hat{h}}$，进而加入 $\mathcal{F}(f)$。 □

任意域： 对域的假设仅用于对定理 1.16 调用于一对$(n+1, s, d)$-稀疏多项式。 [Vol17，引理 42] 给出了一种对一对 $(n+1, s, d)$-稀疏多项式进行整除性测试的确定性算法，其运行时间为 $\mathrm{poly}(n, d! \, s^d)$。因此，用这一结果替换定理 1.16 后，我们可以去除对域的假设。

运行时间： 和之前一样，预处理步骤需要 $\mathrm{poly}(nsd)$ 时间。 引理 3.4 需要 $\mathrm{poly}(nsd)$ 时间。 定理 1.16 和 [Vol17，引理 42] 需要 $\mathrm{poly}(n, d!\ s^d)$ 时间。由于我们进行了一次递归调用，总运行时间为 $\mathrm{poly}(n, d!\ s^d)$。 □

注 8.5. 上述算法还解决了另一个有理插值问题。假设我们可以黑盒访问 $\frac{a}{b}f_0$，其中 f_0 是一个已知的 s有界个体次数稀疏多项式d，而 a、b 是未知的、互素的 (n, s, d)稀疏多项式。此外，假设 $b|f_0$。那么可以在多项式 a、b 运行时间内恢复$(n, d!\ s^d)$该算法对任意域也适用。事实上，我们首先利用 f_0 计算其所有因子，所用方法见定理 1.9，然后按上述方法继续进行。

作为推论，我们得到以下结果，它解决了问题 1.3中关于有界个体次数多项式的情形。我们指出，该结果也可由早期工作的技术推出，例如 [BSV20、Vol15]，，但由于我们未能在文献中找到这一结果，故在此明确陈述。

推论 *8.6*（问题 *1.3*：关于 (n, s, d)-稀疏多项式）。设 $\mathbb{F}$ 是一个域。存在一个确定性算法，运行时间为 $\mathrm{poly}(n, d!\ ,\ s^d)$，该算法以一个 (n, s, d)-稀疏多项式 f 为输入（其定义在 $\mathbb{F}$) 上，且可分解为不可约 s-稀疏多项式 $f = \prod g_i^{e_i}$的乘积），并输出所有的 g_i。此外，存在一个确定性算法来计算重数 e_i，其中：若 $\mathrm{char}(\mathbb{F}) = 0$ 或 $\mathrm{char}(\mathbb{F}) > 2d$，该算法的运行时间为 $\mathrm{poly}(n, d!\ ,\ s^d)$。若 $\mathbb{F}$ 为任意值，该算法的运行时间为 $\mathrm{poly}(n, (d^2)!\ ,\ s^{d^3})$。

证明。由于已知 f（它是 (n, s, d)-稀疏的），计算 x_i 作为 f 的因子的重数 k_i 是显然的。因此，我们可以假设 f 不被任何单项式整除。根据定理 1.9，我们可以找到 f 的所有 s-稀疏因子组成的集合 $\mathcal{F}$。根据定理 1.8，有 $|\mathcal{F}| \leqslant s^d$。由于 f 的所有不可约因子 g_i 都是 s-稀疏的，因此 $g_i \in \mathcal{F}$。这样一来，我们剩下的问题就是从 $\mathcal{F}$ 中识别出 g_i。为此，对于 $\mathcal{F}$ 中每一对非恒定元素，使用 [Vol17] 的整除性测试，检查其中一个是否整除另一个。 $\mathcal{F}$ 中不被 $\mathcal{F}$ 的任何其他成员整除的所有非恒定元素，显然就是所需的不可约因子 g_i。

至于“此外”这一部分，我们使用 引理 4.19 计算重数，若特征$(\mathbb{F}) = 0$ 或特征$(\mathbb{F}) > 2d$；而对于引理 4.13 则适用于任意域。 □

8.3 稀疏不可约多项式乘积的因式分解

在本小节中，我们展示了对 [DST24] (问题 1.4)中开放问题的部分解法。如前几节所示，考虑稀疏多项式因子的乘积，并寻求不可约因子之外、更大一类的稀疏因子，是很自然的。因此，我们根据以下更一般的定理，推导出解决问题1.4 的算法。

定理 1.11 (稀疏多项式因子的乘积的因式分解)。设 $\mathbb{F}$ 是一个满足特征$(\mathbb{F})$ =为 0 或特征$(\mathbb{F}) > 2d$的域。存在一个确定性多项式$(n, d^d, s^{d\log\ell}, \ell^d)$-时间算法：给定域 $\mathbb{F}$ 上 $\mathcal{P}_{Factors}(n, s, d)$ 中 ℓ 个多项式的乘积 f 的黑盒访问，该算法可返回 f 的所有个体次数有界的 s- 稀疏因子 d，这些因子均不被任何单项式整除；此外，还可返回每个 x_i 作为 f 的因子时的重数 k_i。此外，如果 f 实际上是 $\mathcal{P}(n, s, d)$ 中多项式的乘积，则存在一种适用于任意域的此类算法，其运行时间为多项式$(n, (d^2)!, s^{d\log\ell+d^3}, \ell^d)$。

证明。我们再次使用第 6 节中的元算法。设 $f \in \mathbb{F}[x_0, \mathbf{x}]$ 是由 ℓ 个 $\mathcal{P}_{\mathrm{Factors}}(n+1, s, d)$中的多项式构成的乘积。

预处理： 这一步与 定理 1.14 的证明中对应的步骤相同，在 $G = G_{(m)}$ 中用 $m = (2nds)^2$替换 $\hat{}f$的定义。注意， $\tilde{f}$ 至多有 ℓd个不可约因子（作为关于 y的多项式）。因此，它在 $O((\ell d)^d)$ 中有次数至多为 d的 y个因子。

递归调用： 由于 f_0 是 ℓ 个 $\mathcal{P}_{\mathrm{Factors}}(n,s,d)$ (根据 引理 6.2)，我们可以通过一次递归调用，找出所有不被单项式整除的 (n,s,d)-稀疏因子（来自 f_0）。将此集合记为 $\mathcal{F}(f_0)$。根据 推论 8.3，有 $|\mathcal{F}(f_0)| \leqslant$ 多项式$(s^{d\log \ell})$。

生成并筛选候选解列表： 这一步与定理 1.9证明中的对应步骤完全相同。事实上，类似地，我们可以从 $\mathcal{F}(f_0)$ 生成候选项集合：考虑次数至多为h 的f 的因子 d，其中在 y中，因为我们只关注个体次数有界的因子 d。再次调用定理 1.16即可完成筛选。

任意域： 将证明推广到任意域时，唯一的问题在于用到了定理 1.16。因此，对于小特征的域，我们假设 f 是 s-稀疏多项式的乘积。在这种情况下，我们可以应用定理 1.16的“此外”部分，论证便可顺利进行。

运行时间： 与前面的证明一样，运行时间由列表的大小以及定理 1.16的运行时间决定。断言的界限由此得出。 □

最后，我们以如下直接推论结束，它给出了对问题 1.4所承诺的部分解答。

定理 1.10 (稀疏 (n,s,d)-稀疏多项式乘积的因式分解）。设 $\mathbb{F}$ 是一个特征为零或大于 $2d$ 的域。存在一个确定性 $\mathrm{poly}(n,d^d,s^{d\log \ell},\ell^d)$-时间算法，在获得乘积 $f=\prod_{i=1}^{\ell}\phi_i$ 的黑盒访问后，其中 $\phi_1,\ldots,\phi_\ell \in \mathcal{P}(n,s,d)$ 均为不可约多项式（不一定互不相同），可返回这些 ϕ_i。此外，若去掉对域的假设，则存在一个算法，可在 $\mathrm{poly}(n,(d^2)!,s^{d\log \ell+d^3},\ell^d)$-时间内解决此问题。

证明。根据定理 1.11，我们可以在确定性 $\mathrm{poly}(n,d^d,s^{d\log \ell},\ell^d)$-时间内找到所有 s-稀疏因子 f，其个体次数有界 d。应用定理 1.16，我们可以判断任意两个候选项 ϕ 和 ψ是否满足 $\psi \mid \phi$。由于 f 的每个不可约因子都是 s-稀疏的，且个体次数至多为 d，因此不可约因子恰好是那些ϕ 不被任何其他候选项整除的候选项 ψ。要计算每个所得不可约因子作为 f的因子时的重数，只需调用引理 4.19。

至于“此外”部分，使用定理 1.11 的“此外”部分，在 (n,s,d)-稀疏因子中找出所有 f，用时为 $\mathrm{poly}(n,(d^2)!,\ s^{d^3},\ \ell^d)$-时间。用其“此外”部分（或定理 1.16 的整除性测试）替换 [Vol17]，以检查任意两个候选项中一个是否整除另一个；并用引理 4.13的“此外”部分替换引理 4.19，以计算重数。 □

9 (n,s,d)-稀疏多项式的因式分解

本节将应用我们的方法，为 (n,s,d)-稀疏多项式推导因式分解结果，而不对其因子的稀疏度作任何额外假设。为此，我们依赖定理 2.14给出的稀疏度上界。

正如第 1.4.4 节中所述，我们给出两种算法：一种用于对单个 (n,s,d)-稀疏多项式进行因式分解，另一种用于对这类多项式的乘积进行因式分解。第一种算法的运行时间为 $\mathrm{poly}(n,S)$（其中 S 是定理 2.14中的界），而第二种算法的运行时间为 $\mathrm{poly}(n,d!,S^d)$-时间。我们很快就会看到，造成这种差异的主要原因是：如果给定的 S-稀疏多项式本身是 s-稀疏的，那么很容易检验它是否为输入多项式的因子，因为此时商应当

也是 S-稀疏的，因此可以进行插值和重构；但对于由 s-稀疏多项式乘积构成的因子，情况就不再如此。

9.1 稀疏多项式的因式分解

在本小节中，我们证明以下定理，该定理改进了定理 2.15中的算法。

定理 1.12。 存在一个确定性的$\mathrm{poly}(n, s^{d^2 \log n})$-时间算法，该算法给定一个具有 d 有界个体次数、含有 n个变量的 s-稀疏多项式，可以返回其不可约因子分解。

注意，仅根据定理 2.14中的界，这是我们所能期望的最佳结果。此外，正如从证明中很容易看出，对定理 2.14的改进会自动带来我们算法运行时间的改进：如果将定理 2.14 中的界改进为更小的界 M，那么我们的算法运行时间将变为 $\mathrm{poly}(n, M, s^d)$。

证明。 证明沿用定理 1.9的证明思路，但有一些小的修改，我们将对此加以说明。对于任意域 $\mathbb{F}$，该算法都是相同的，与特征无关。

记 S 为由定理 2.14 得出的一个显式稀疏度上界，适用于 $(n+1, s, d)$-稀疏多项式： $S = \mathrm{poly}(n, s^{d^2 \log n})$。

预处理步骤： 我们执行与定理 1.9的证明中相同的预处理步骤，其中定义 $\hat{f}$，相对于生成器 $G = G_{(m)}$，针对 $m = (2ndS)^2$。

递归调用： 由于 f_0 是一个 (n, s, d)-稀疏多项式，因此对算法进行递归调用将返回一个包含其不可约因子的列表 $\mathcal{F}'(f_0)$。注意，这些因子的稀疏度可能为 S。根据定理 1.8，该列表的大小至多为 $d \log s$，其中计入重数；移除其中可能包含的变量时（也就是说，我们考虑 $\mathcal{F}'(f_0) \setminus \{x_i\}_{1 \leqslant i \leqslant n}$而非 $\mathcal{F}'(f_0)$）。因此， f_0至多有 s^d 个不被单项式整除的因子。通过遍历不可约因子的所有子集（计入重数），可以在$s^d \mathrm{poly}(S)$时间内枚举并插值这些因子。我们将 f_0 的所有不被单项式整除的因子组成的列表记为 $\mathcal{F}(f_0)$。

生成并筛除候选解列表：候选项列表的生成方式与定理 1.9的证明相同。事实上， $\mathcal{F}(f_0)$ 包含 f_0的所有因子（不能被单项式整除），因此我们可以用完全相同的方法，找出 f的因子（不能被单项式整除）的候选项列表。关于筛除，我们注意到，要判断候选多项式 g 是否为 f的因子，并不需要使用定理 1.16。事实上，如果 $g \mid f$，那么 f/g 是S-稀疏的。因此，我们可以将 $f(G)/g(G)$ 插值为一个 S-稀疏多项式 h，并检查是否$g \cdot h = f$。

这表明如何找出 f 的所有不被单项式整除的因子。为了完成证明，我们需要说明如何利用这些信息得到 f 分解为不可约多项式的因式分解。我们首先说明如何在已找到的因子中识别不可约因子。注意，如果因子 g 是可约的，那么它的因子也是 f的因子。因此，我们可以检查是否存在一对因子 h_1、 h_2 使得 $g = h_1 \cdot h_2$，并且如果没有找到这样的 h_1、 h_2，则返回g 是不可约的。

计算重数：至此，我们已有 f的非单项式不可约因子列表。由于整除 f 的最大单项式已在预处理步骤中计算得到，剩下的问题是计算 f的每个不可约因子的重数。这可以按前一步的方式完成。我们逐一遍历 $k \geqslant 0$ 的所有取值，检查是否存在 f 的因子 h 使得 $h \cdot g^k = f$。我们在第一个不存在这样的因子的 k 处停止，并返回 g 的重数为 $k-1$。由于对 $i < k$而言，所有 g^i 都是 f的因子，它们的稀疏度均为poly(S)，因此该算法在poly(S)时间内运行。

运行时间：显然，运行时间关于稀疏度 S、变量数目 n 以及列表的大小呈多项式级别，而列表大小的上界为 s^d。 □

9.2 稀疏多项式乘积的因式分解

我们现在证明以下定理。注意，该定理不同于定理 1.11；定理 1.11 仅找出乘积中 s-稀疏因子，其中多项式乘积中的多项式属于 $\mathcal{P}_{\text{Factors}}(n, s, d)$，而定理 1.13 则计算此类乘积的因式分解，将其分解为不可约因子，即使这些因子并非 s-稀疏多项式。

定理 1.13。设 $\mathbb{F}$ 是一个域，满足特征$(\mathbb{F}) = 0$ 或特征$(\mathbb{F}) > 2d$。存在一个确定性多项式$(n, s^{d^3 \text{对数} n}, (d\ell)^d)$-时间算法：给定乘积 f 的黑盒访问，其中 ℓ 个多项式属于$\mathcal{P}_{Factors}(n, s, d)$ 且系数取自 $\mathbb{F}$，返回其不可约因子的因式分解。此外，还存在一个适用于任意域的算法，其运行时间为多项式$(n, (d^2)!, s^{d^5 \text{对数} n}, \ell^d)$。

证明。 用 $S = s^{O(d^2 \log^{(n+1)})}$ 表示 定理 2.14 针对 $(n+1, s, d)$-稀疏多项式给出的稀疏度上界。

我们沿用定理 1.14的证明思路，但将其与定理 2.14的稀疏度上界相结合。

预处理：这与定理 1.14 的证明相同，其中$\hat{f}$ 被定义为$\hat{f} = \tilde{f}(y, G)$，其中 $G = G_{(m)}$。这里， m 由定理 5.2 对 (n, S, d)-稀疏多项式给出（即，$m = \text{poly}(n, d!, S^d) = \text{poly}(n, d!, s^{d^3} \log^n)$）。根据断言 2.3，$\hat{f}$ 至多有 ℓd 个不可约因子（作为 y中的多项式），因而在 y中次数至多为 d的因子至多有 $O((\ell d)^d)$ 个。

递归调用：根据引理 6.2， f_0 是 ℓ 个 $\mathcal{P}_{\text{Factors}}(n, s, d)$的乘积。因此，通过进行一次递归调用，我们可以得到它的不可约多项式因式分解。记 $\mathcal{F}(f_0)$ 为 f_0的不可约因子集合。

生成并筛选候选解列表：由于我们已经得到 f_0的所有不可约因子，因此可以按照定理 1.14 的证明中的方法，利用 (n, S, d)-稀疏因子 f，并运用定理 5.2和定理 1.16，得到一个

剪枝列表：现在我们得到了一组候选项，它们都是 S-稀疏 f的因子。注意，如果 g 是这样的一个因子，但它不是不可约的，那么 g 的每个不可约因子也必定在列表中。因此，为了找出不可约因子，我们对每一对候选项定理 1.16 (针对 S-稀疏多项式)并检查 ϕ, ψ 是否成立。那些不存在其他候选项可整除它们的候选项 $\psi | \phi$，即为所需的不可约因子。 ϕ

计算重数：我们使用引理 4.19计算不可约因子的重数。

任意域： 为了得到该结果，我们将 定理 5.2 和 定理 1.16 替换为它们各自的“此外”部分。我们还使用 引理 4.13 而非 引理 4.19。

运行时间： 首先考虑零特征或大特征的域。运行时间由 $\hat{F}$ 的因子数量，以及对 $(n+1, S, d)$-稀疏多项式应用定理 5.2、定理 1.16和引理 4.19 所决定。这得到所声称的 $\mathrm{poly}(n, d!, S^d, (d\ell)^d) = \mathrm{poly}(n, s^{d^3}\log^n, (d\ell)^d)$ 运行时间。

对于任意域，定理 5.2 和定理 1.16 的“此外”部分以及引理 4.13意味着运行时间为 $\mathrm{poly}(n, (d^2)!, S^{d^3}, (d\ell)^d) = \mathrm{poly}(n, (d^2)!, s^{d^5}\log^n, \ell^d)$ □

注 9.1。在定理 1.10 中，我们给出了一个多项式$(n, d^d, s^{d\log\ell}, \ell^d)$时间算法来解决这个问题，但额外假设 f 的不可约因子实际上属于 $\mathcal{P}(n, s, d)$。因此，定理 1.13更为一般，实际上给出了一个多项式$(n, s^{d^3\text{对数}^n}, (d\ell)^d)$ 算法来解决定理 1.10。也就是说，我们有两种算法来解决问题 1.4：一种算法在乘积长度上是准多项式时间的，在变量数量上是多项式时间的；另一种算法在变量数量上是准多项式时间的，在乘积中的项数上是多项式时间的。

10 未来工作与公开问题

最后，我们讨论未来研究的若干可能方向。

首先，令人有些失望的是，我们的主要技术引理 引理 4.15仅在零特征或足够大的特征域上成立。将这一结果推广到任意域将会很有意义。正如第 4 节所讨论的，在这一方向上已有一些结果；剩下的挑战是消除小特征下所需的额外非退化条件。

接下来，若能确定定理 1.9中的不可约因子，将会非常有意义。换言之，我们希望有一种高效算法，给定一个 (n, s, d)-稀疏多项式 f，能够在多项式时间内 f 判断其是否不可约(n, s^d)。本文提出的技术似乎不足以解决这一问题，因为它们只能提供关于$\phi(G_{(m,i)})$ 与 $\psi(G_{(m,i)})$ 在 ϕ下 $\psi \in \mathcal{P}_{\mathrm{Factors}}(n, s, d)$互素性的信息，却无法给出不可约因子数量的信息。

推广我们的有理插值定理也很有意义。例如，即便是确定性地在多项式时间内对互素稀疏多项式的商进行插值这一看似基础的问题，仍然远未解决。

要为问题 1.4设计多项式时间算法，从而改进定理 1.10，似乎也需要新的思路。正如本文多处所述，主要难点在于 c_0 可能含有非稀疏不可约因子，因此递归调用 f_0 时必须枚举所有可能情况，导致拟多项式运行时间。相比之下，定理 1.14 中的多项式时间算法依赖于这样一个事实：c_0 （或者在该情形下的 b ）分解出的因子全是稀疏多项式，这由引理 2.13保证。结合通过推论 4.12确定重数的能力，这一结构足以得到多项式时间算法。

最后，任何对 定理 2.14 中界限的改进，都会直接加强第9节的结果，并具有重要意义。

参考文献

[BKR+25] Somnath Bhattacharjee、Mrinal Kumar、Shanthanu S Rai、Varun Ramanathan、Ram- prasad Saptharishi 和 Shubhangi Saraf。由 Furstenberg 的一项结果得到的因式分解闭包性质。arXiv 预印本 arXiv:2506.23214，2025。1[BOT88] Michael Ben-Or 和 Prasoon Tiwari。稀疏多元多项式插值的确定性算法。载于第二十届 ACM 计算理论研讨会论文集，第 301–309 页，1988。1[BS25] Pranav Bisht 和 Nitin Saxena。通过对称多面体去随机化：某些稀疏多项式的多项式时间因式分解。ACM计算理论汇刊，17(2):1–20，2025。5[BSV20] Vishwas Bhargava、Shubhangi Saraf 和 Ilya Volkovich。有界个体次数稀疏多项式的确定性因式分解。美国计算机协会学报（JACM），67(2):1–28，2020。1, 2, 3, 4, 5, 6, 9, 11, 12, 14, 15, 39[BV25] Pranav Bisht 和 Ilya Volkovich。关于求解稀疏多项式因式分解相关问题。计算复杂性，34(1):7，2025。1, 2, 4, 5, 6, 7, 8, 9, 12, 19, 21, 23, 25, 28[CDGK91] Michael Clausen、Andreas Dress、Johannes Grabmeier 和 Marek Karpinski。有限域上 k-稀疏多元多项式的零测试与插值。理论计算机科学，84(2):151–164，1991。1[CL11] Annie Cuyt 和 Wen-shin Lee。多元有理函数的稀疏插值。理论计算机科学，412(16):1445–1456，2011。5[DG24]Pranjal Dutta 和 Sumanta Ghosh。SIGACT News 复杂性理论专栏 121。SIGACT News，55(2):53–88，2024年6月。1[DST24] Pranjal Dutta、Amit Sinhababu 和 Thomas Thierauf。低次因子的多元多项式因式分解去随机化。载于 Amit Kumar 和 Noga Ron-Zewi 编，近似、随机化与组合优化：算法与技术，APPROX/RANDOM 2024，伦敦政治经济学院，英国伦敦，2024年8月28–30日，第317卷 LIPIcs，第75:1–75:20页。Schloss Dagstuhl - Leibniz-Zentrum f¨ur Informatik，2024。1, 2, 5, 35, 39[For15] Michael A Forbes。通过移位偏导数进行确定性整除性测试。载于2015年 IEEE 第56届计算机科学基础年度研讨会论文集，第451–465页。IEEE，2015。5, 6[FS15] Michael A Forbes 和 Amir Shpilka。复杂性理论专栏 88：多项式因式分解中的挑战1。ACM SIGACT News，46(4):32–49，2015。2[GK87] Dima Yu Grigoriev 和 Marek Karpinski。永久式为多项式有界的二分图匹配问题属于 nc。载于第28届计算机科学基础年度研讨会（sfcs 1987）论文集，第166–172页。IEEE，1987。1[GKL12] Ankit Gupta、Neeraj Kayal 和 Satyanarayana V. Lokam。顶层扇入为2的深度4多线性电路的重构。载于 Howard J. Karloff 和 Toniann Pitassi 编，

第44届计算理论研讨会论文集（STOC 2012），美国纽约州纽约市，2012年5月19日至22日，第625–642页。ACM，2012年。14[GKS90] Dima Grigoriev、Marek Karpinski 和 Michael F. Singer。有限域上稀疏多元多项式插值的快速并行算法。《SIAM计算杂志》，19(6):1059–1063，1990年。1[GKS94] Dima Grigoriev、Marek Karpinski 和 Michael F Singer。稀疏有理插值的计算复杂性。《SIAM计算杂志》，23(1):1–11，1994年。2, 5[GKZ08] Israel M Gelfand、Mikhail M Kapranov 和 Andrei V Zelevinsky。判别式、结式与多维行列式。现代 Birkhäuser 经典著作。Birkhäuser Boston, Inc.，美国马萨诸塞州波士顿，2008年。1994年版重印。15[Kal85a] Erich Kaltofen。利用直线程序给定的多项式进行计算 I：最大公因子。载于第十七届 ACM 计算理论研讨会论文集，第131–142页，1985年。1[Kal85b] Erich Kaltofen。从多元整系数多项式因式分解到二元及一元整系数多项式因式分解的多项式时间归约。《SIAM计算杂志》，14(2):469–489，1985年。14[KRS24] Mrinal Kumar、Varun Ramanathan 和 Ramprasad Saptharishi。常数深度电路低次因子的确定性算法。载于2024年ACM-SIAM离散算法年会（SODA）论文集，第3901–3918页。SIAM，2024年。5[KS01] Adam R. Klivans 和 Daniel Spielman。多元多项式随机性高效的恒等式检验。STOC '01，第216–223页，美国纽约州纽约市，2001年。计算机协会。1, 7, 17[Len85] Arjen Klaas Lenstra。有限域上多元多项式的因式分解。《计算机与系统科学杂志》，30(2):235–248，1985年。14[LLL82] Arjen K. Lenstra、Hendrik W. jr. Lenstra 和 L´aszl´o Lov´asz。有理数系数多项式的因式分解。《数学年刊》，261:515–534，1982年。13[Sax09] Nitin Saxena。多项式恒等式检验的进展。《EATCS通报》，99:49–79，2009年。1[Sax14] Nitin Saxena。多项式恒等式检验的进展 II。载于《计算复杂性展望：Somenath Biswas纪念文集》，第131–146页。施普林格，2014年。1[SV09] Amir Shpilka 和 Ilya Volkovich。读一次公式的改进型多项式恒等式检验。载于组合优化近似算法国际研讨会论文集，第700–713页。施普林格，2009年。17[SY10] Amir Shpilka 和 Amir Yehudayoff。算术电路：近期成果综述与开放问题。《理论计算机科学前沿与趋势》，5(3–4):207–388，2010年。1

[vdHL21] Joris van der Hoeven与Gr´egoire Lecerf。关于有理函数与最大公约数的稀疏插值。*ACM*《计算机代数通讯》，55(1):1–12，2021。 5

[Vol15] Ilya Volkovich。将稀疏多项式确定性因式分解为多线性因子与单变量多项式之和。载于近似、随机化与组合优化：算法与技术（*APPROX/RANDOM 2015*），第943–958页。Schloss Dagstuhl–Leibniz-Zentrum f¨ur Informatik，2015。1， 2， 4， 5， 39

[卷17]Ilya Volkovich。关于稀疏多项式上的一些计算。载于 近似、随机化与组合优化：算法与技术（*APPROX/RANDOM 2017*），第 48–1 页。Schloss Dagstuhl–Leibniz 信息学中心，¨2017 年。1， 2， 4，5， 6， 7， 14， 16， 25， 38， 39， 40

[VZGK85] Joachim Von Zur Gathen 和 Erich Kaltofen。稀疏多元多项式的因式分解。 计算机与系统科学杂志，31(2):265–287，1985 年。1， 2

[vzGS92] Joachim von zur Gathen 和 Victor Shoup。计算 Frobenius 映射并对多项式进行因式分解。 计算复杂性，2(3):187–224，1992 年。13

A 第 4 节结果的另一种证明

本节将介绍推论 4.12的一种替代证明。虽然该引理对任意域都成立，但此处给出的证明仅适用于零特征域，或特征大于个体次数的情形。因此，推论 4.12 和引理 4.15的结论都比我们在此给出的更强。不过，这种证明方法略有不同，我们将其纳入本文，希望它能对其他应用有所帮助。

虽然推论 4.12 和引理 4.15 的证明利用西尔维斯特矩阵的秩来研究多项式的因式分解，但我们在此使用一个称为 Δ_k的算子。众所周知，多项式的判别式 Δ （定义见定义 2.19）是一种可用于检测是否存在重数大于一的不可约因子的算子。它的推广形式 Δ_k是我们即将定义的一种算子，可用于检测重数至少为 $k+1$的不可约因子，其中 k取较小值。第二个重要性质是， Δ_k 与判别式一样，不会使其输入的稀疏度增加太多。

现在我们给出 Δ_k 的定义。

定义 A.1. 令 $f \in \mathbb{F}[x]$ 为非恒定多项式。定义

$$\Delta_k(f)(\lambda_1,\ldots,\lambda_k) = \operatorname{res}(f, \lambda_1 f^{(1)} + \ldots + \lambda_k f^{(k)}) \in \mathbb{F}[\lambda_1,\ldots,\lambda_k] \tag{19}$$

其中 $f^{(i)}$ 是 f 的 i 阶导数。

对于 $f \in \mathbb{F}[x_0, x_1, \ldots, x_n]$, $\Delta_{k,x_0}(f) \in \mathbb{F}[x_1, \ldots, x_n, \lambda_1, \ldots, \lambda_k]$， 当将 f 视为 $\mathbb{F}(x_1,\ldots,x_n)[x_0]$ 中的一个元素时，它就是 $\Delta_k(f)$。

例 A.2. *F* 或者 $k=1$，可以看出 $\Delta_1 = \lambda_1\Delta$。因此，例如，当且仅当 $\Delta(f) =$ 时， $\Delta_1(f) = 0$ 0.

如前所述，应将 Δ_k 视为 Δ的一种推广，它刻画了重数至少为 $k+1$的因子的存在；这与 Δ刻画重数至少为二的因子的存在类似。这个例子证明了该说法对 $k = 1$成立；一般情形由下面的引理给出。

引理 A.3. 设 $f \in \mathbb{F}[x]$ 是次数至多为 d的非常数多项式，并假设$\operatorname{char}(\mathbb{F})$ 为零或大于 d。则 f 具有重数至少为 $k+1$ 的不可约因子，当且仅当 $\Delta_k(f) = 0$。特别地，若 $f \in \mathbb{F}[x_0, x_1, \ldots, x_n]$ 满足 $x_0 \in$ 变量集(f) 且$\deg_{x_0}(f) \leqslant d$，在关于$\mathbb{F}$的相同假设下，则 f具有重数至少为 $k+1$且依赖于 x_0的不可约因子，当且仅当$\Delta_{k,x_0}(f) = 0$。

证明。 注意，由于 $\mathrm{char}(\mathbb{F}) \in \{0\} \cup [d+1, \infty)$，且 $\deg f \leqslant d$，每个不可约因子 ϕ（属于 f）是可分的，也就是说，它在 $\mathbb{F}$上分解为不同线性多项式的乘积。看出这一点的一种方式是注意到 $\phi' \neq 0$（由于对 $\mathbb{F}$的假设），因此 ϕ' 必须与 ϕ互素，所以由 引理 2.20 我们必有 $\Delta(\phi) = \mathrm{res}(\phi, \phi') \neq 0$，从而（再次由 引理 2.20） ϕ 是无平方因子的。

因此， f 有重数至少为 $k+1$ 的不可约因子，当且仅当它在$\mathbb{F}$上有重数至少为 $k+1$ 的根。

假设，且 不失一般性，假设 f 是首一的。在这种情况下，根据 引理 2.20，我们知道 t:

$$\Delta_k(f) = \mathrm{res}(f, \lambda_1 f^{(1)} + \ldots + \lambda_k f^{(k)}) = \prod_{\alpha : f(\alpha)=0} \left(\lambda_1 f^{(1)}(\alpha) + \ldots + \lambda_k f^{(k)}(\alpha)\right) \tag{20}$$

也就是说， $\Delta_k(f)$ 是 $\mathbb{F}[\lambda_1, 、 \ldots\ldots 、 \lambda_k]$中的线性泛函的乘积，因此，当且仅当其中一个泛函为零时，它才为零。

为此，注意对于 f的一个根 α，泛函 $\lambda_1 f^{(1)}(\alpha) + \ldots\ldots + \lambda_k f^{(k)}(\alpha)$ 为零，当且仅当 $f^{(i)}(\alpha) = 0$ 对 $0 \leqslant i \leqslant k$成立；也就是说，当且仅当 f有一个重数至少为 $k+1$ 的根 α。这就证明了该引理。

至于“特别是”这一部分，注意根据高斯引理（引理 2.1），依赖于 x_0 的 $f \in \mathbb{F}[x_0, x_1, \ldots\ldots 、 x_n]$ 的不可约因子 x_0对应于 f作为$\mathbb{F}(x_1, \ldots\ldots 、 x_n)[x_0]$中的元素时的不可约因子。因此，只需在 $\mathbb{F}(x_1, \ldots\ldots 、 x_n)$上运算，即可由引理的第一部分轻易推出最后一条结论；注意其特征与 $\mathbb{F}$的特征相同。 □

注 A.4. 关于 $\deg(f)$ 与 $\mathrm{char}(\mathbb{F})$ 的条件是必要的。例如，考虑素数 p 的多项式 $f(x, y) = x^p - y \in \mathbb{F}_p[x, y]$。 f 显然是不可约的，但 $\Delta_x(f) = 0$，因为 f 作为 $\mathbb{F}_p(y)[x]$ 中的元素是一个平方（事实上是一个 p 次幂）。

上述引理给出了 Δ_k 推广 Δ 的一种方式。就我们的目的而言，我们需要研究这种推广成立的另一种方式——对稀疏多项式的作用。

引理 A.5. 设 $f \in \mathbb{F}[x_0, x_1, \ldots, x_n]$ 是一个具有有界个体次数 d 的 s-稀疏多项式。则 $\Delta_k f \in \mathbb{F}[x_1, \ldots, x_n, \lambda_1, \ldots, \lambda_k]$ 是一个个体次数至多为 $2d^2$ 的 $(2d)!(ks)^{2d}$-稀疏多项式。

证明。 根据引理 2.20，只需注意到 $\lambda_1 f^{(1)}(\alpha) + \ldots + \lambda_k f^{(k)}(\alpha)$是一个稀疏度为 ks 且个体次数有界的多项式 d。 □

现在我们说明如何使用 Δ_k 得到与推论 4.12类似的结果。为此，我们先给出以下引理。

引理 A.6。 假设 $\mathbb{F}$ 是一个域，且 特征$(\mathbb{F}) \in \{0\} \cup [d+1, \infty)$。设 f是定义在 $\mathbb{F}$上的一个 (n, s, d)-稀疏多项式，并固定一个坐标 i。假设 f 作为关于 x_i的多项式，没有重数至少为 $k+1$ 的不可约因子。那么存在一个显式的 $m =$多项式$(n, d!\ (ks)^d)$，使得 $f(G_{(m,i)})$ 作为关于 u的多项式，也没有重数至少为 $k+1$ 的不可约因子。

证明。由引理 A.3，由于 f 作为关于 x_i 的多项式没有重数至少为 $k+1$ 的不可约因子，因此我们有 $\Delta_{k,\, x_i}(f) \neq 0$。由引理 A.5， $\Delta_{k,\, x_i}(f)$ 是一个具有有界个体次数 $2d^2$、含 $n+d-1$ 个变量的 $\mathrm{poly}(d!(ks)^d)$-稀疏多项式。因此，由引理 3.4，对于 $m = ((n+d)(d!(ks)^d)(2d^2))^2 = \mathrm{poly}(n, d!, (ks)^d)$，我们知道 $\Delta_{k,\, x_i}(f)(G_{(m,i)}) \neq 0$，其中我们根据 $G_{(m,i)}$ 将每个坐标 x_j 替换为 $j \neq i$（保留 $\lambda_1, \ldots, \lambda_k$）。容易看出 $\Delta_{k,\, u}(f(G_{(m,i)})) = \Delta_{k,\, x_i}(f)(G_{(m,i)})$，因此可得 $\Delta_{k,\, u}(f(G_{(m,i)})) \neq 0$。断言由引理 A.3 的反向结论得出。 □

我们现在应用上一引理来证明以下较弱版本的推论 4.12 .

引理 A.7。设 $\mathbb{F}$ 是一个域，满足 特征$(\mathbb{F}) \in \{0\} \cup [d+1, \infty)$。设 ϕ 是一个有界个体次数为 d 的不可约 s-稀疏多项式， ψ 是某个有界个体次数为 d 的 s-稀疏多项式g 的不可约因子，且不同于 ϕ; 二者都定义在 $\mathbb{F}$ 上。固定某个满足 $x_i \in$ 的 $ivar(\phi)$。那么，对于 $m =$ 的某个显式取值 $poly(n, d^{d^2}, s^{d^3})$，当视为关于 u 的多项式时， $\phi(G_{(m,i)})$ 与 $\psi(G_{(m,i)})$ 互素。此外，若 ϕ 是多二次的，则取 $m = poly(n, d^d, s^{d^2})$ 实际上就足够了。

证明。设 $g = \prod \psi \cdot \phi^k$，每个 ψ 都不可约，且不与 ϕ 相伴。此后我们将这些多项式视为 x_i 上的多项式。

考虑多项式 $\tilde{g} = \phi^{d+1} g$。其稀疏度至多为 s^{d+2}，其次数至多为 $O(d^2)$，且 ϕ 恰好整除它 $k + d + 1$ 次。由于 $\deg_{x_i}(g) \leqslant d$， $\tilde{g}$ 的每一个其他不可约因子的重数至多为 $d-k < d+1$，因此 ϕ 必定是 $\tilde{g}$ 重数最大的因子。也就是说， $\tilde{g}$ 没有重数超过 $k+d+1$ 的因子。由引理 A.6，若 $m = \text{poly}(n, (d^2)!, ((k+d)s^{d+2})^{d^2}) = \text{poly}(n, d^{d^2}, s^{d^3})$，则对 $\tilde{g}(G_{(m,i)})$ （作为 u 上的多项式）同样成立。因此，必定有 $\psi(G_{(m,i)})$ 与 $\phi(G_{(m,i)})$ 互素——否则我们就会得到 $\phi(G_{(m,i)})$ 的一个因子，它在 $\tilde{g}(G_{(m,i)})$ 的因式分解中至少出现 $k+d+2$ 次（其中 $k+d+1$ 次来自 $\tilde{g}$的 ϕ 个因子，另有一次来自 ψ 因子）。

至于“此外”部分，注意到如果 ϕ 是多二次的，那么 $\tilde{g}$ 的次数为 $O(d)$，因此取 $m = \text{poly}(n, d!, ((k+d)s^{d+2})^d) = \text{poly}(n, d^d, s^{d^3})$ 就足够了。 □

注 A.8. 事实上，证明表明 $\phi(G_{(m,i)})$ 无平方因子，因为生成器 G 保持判别式 $\Delta(\phi)$ 非零。