arXiv:2602.12772v1 [math.AC] 13 Feb 2026

在 Lean 中形式化格罗布纳基理论

Junyu Guo[1†], HaoShen[2†], Junqi Liu[2], Lihong Zhi[2]

[1]中山大学逻辑与认知研究所、哲学系，中国广东省广州市，邮编 510275。[2]中国科学院大学数学与系统科学研究院数学科学国家重点实验室，中国北京市中关村东路 55 号，邮编 100190。

通讯作者：guojy228@mail2.sysu.edu.cn;shenhao24@amss.ac.cn;liujunqi@amss.ac.cn; lzhi@mmrc.iss.ac.cn;

†贡献相同。作者按姓氏字母顺序排列。

摘要

我们介绍了在 Lean 4 中对格罗布纳基理论进行的形式化，该工作构建在 Mathlib 为多元多项式和单项式序提供的基础设施之上。我们的开发涵盖格罗布纳基理论的核心基础，包括带余多项式除法、布赫伯格判据，以及约化格罗布纳基的存在性与唯一性。我们针对由任意类型索引的多项式环统一发展这套理论，从而能够处理含有无限多个变量的环中的格罗布纳基。此外，我们通过单项式序嵌入和基于滤子的极限构造，证明无限变量情形下的约化格罗布纳基可以由有限变量子环上的约化格罗布纳基刻画，从而连接了有限与无限两种情形。

关键词： 格罗布纳基、Lean、Mathlib、计算机代数

1 引言

格罗布纳基由 Buchberger [1], 引入，是计算交换代数和代数几何的基石。它们为理想成员判定、消元理论以及求解多项式方程组等基础问题提供了统一的算法框架。因此，它们是代数推理的核心引擎，支持多项式中的自动演绎

代数及密码学、机器人学和控制理论等应用。因此，全面发展格罗布纳基理论，对于推进形式化数学并验证代数算法的正确性至关重要。迄今为止，最全面的格罗布纳基理论形式化工作是在 Isabelle/HOL 证明助手中完成的 [2]，其中包括对 Faugère 等先进方法的认证实现 F_4 算法 [3] 以及基于签名的技术 [4–6]。在 Coq 中 [7]，早期工作专注于形式化 Buchberger 算法，并确立格罗布纳基构造性存在性 [8, 9]。其他工作还包括在 ACL2 [10] 和 Mizar 中进行的形式化 [11]。Dermitzakis 在 Lean 中实现了 S 多项式和基本的格罗布纳基构造 3 [12]。尽管该工作验证了此类开发的可行性，但它在很大程度上仍与 Mathlib 的核心代数基础设施相互独立。Poulsen 和 Guo 分别在 Lean 3 [13]和全新的 Lean 4 生态系统中探索了格罗布纳基理论的初步形式化 [14]。然而，他们既没有继续维护其形式化成果，也没有将工作提交至 Mathlib。Lean 策略 `polyrith` 成功运用格罗布纳基技术实现证明自动化。该策略 `polyrith` 依赖 SageMath [15] 计算证书，再在 Lean 中验证这些证书。Lean 4 策略 `grobner`为多项式目标提供基于格罗布纳基的求解功能，并作为通用自动化策略 `grind`的轻量封装实现。尽管这些策略在目标导向的自动化方面效果显著，但它们辅助的是证明过程，而非提供格罗布纳基理论的形式化发展。与此不同，我们的工作在 Lean 4 中发展格罗布纳基理论的核心基础设施。这项工作的部分内容已合并至 Mathlib，且该形式化旨在支持库级别的长期复用。我们工作的一个关键特色是其一般性。不同于局限于有限格罗布纳基的经典形式化工作，我们复用 Mathlib 的通用基础设施，处理由任意给定类型中的元素索引的多项式环，并且也支持 Iima 和 Yoshino 意义下的无限Gröbner基 [16]。除了定义层面的一般性之外，我们还形式化了连接不同多项式环上格罗布纳基的重要结果，其中包括通过有限约化格罗布纳基的基于滤子的极限刻画无限约化格罗布纳基。这在单一形式化框架中衔接了有限与无限情形，为未来形式化计算代数几何提供了可复用的基础设施。

组织结构

本文其余部分安排如下。第2节回顾数学预备知识和现有的 Mathlib 基础设施，包括多元多项式和单项式序。第3节对格罗布纳基理论的核心内容进行了形式化。我们首先引入带底元的次数，以解决零多项式相关的算术歧义，接着形式化多项式除法和余式的存在性。随后，我们定义格罗布纳基和约化格罗布纳基，确立其基本性质，并对布赫伯格判据进行形式化验证。第4 节介绍单项式序嵌入，证明格罗布纳基在这些嵌入下的不变性，并通过极限构造确立无限格罗布纳基的刻画。最后，第5 节讨论外部计算认证方面的未来工作。

2 预备知识

Lean 是一种基于依赖类型论的开源交互式定理证明器，具有一个小型可信内核 [17]。它集成交互式证明语言、高效的编译型编程语言以及强大的元编程框架。其最新版本为 Lean 4 [18]。在 Lean 生态系统中，Mathlib 提供了一个规模庞大且结构统一的形式化数学库 [19]。Mathlib 为多项式代数奠定了基础，其中包括 `MvPolynomial`、`Finsupp`，以及近年来对单项式序和多元多项式除法的形式化工作。我们的形式化直接构建于这一代数层级之上，重用核心抽象，同时保持与库在定义上的兼容性，从而为格罗布纳基理论提供所需的基础设施。设 σ 是一个指标类型，并设

$$\mathcal{M} := (\sigma \to_0 \mathbb{N}) \tag{1}$$

表示从 σ 到 $\mathbb{N}$ 的有限支撑函数类型。$\mathcal{M}$ 上的单项式序定义如下。

定义 1 (单项式序)。单项式序是一个二元关系 $\leq$ 在 $\mathcal{M}$上满足以下条件：

(1) 关系 $\leq$ 是 $\mathcal{M}$ 上的良序，即它是线性序，且不存在无限严格下降链 $a_1 > a_2 > a_3 > \cdots$.*(2)* 对于所有 a、b、$c \in \mathcal{M}$，如果 $a \leq b$，则 $a + c \leq b + c$。*(3)* 零元素 0 是最小元素，即对于所有 $a \in \mathcal{M}$，$0 \leq a$ 。

这种序的一个标准例子是字典序。假设指标类型 σ 带有线性序。相应的字典序 $\leq_{\text{lex}}$ 在 $\mathcal{M}$ 上定义如下：

定义 2 (字典序)。假设 σ 配备了一个线性序 $\geq$，其中 $>$是良基的，即不允许存在无限严格上升序列 $a_1 <a_2 < a_3 < \cdots$。对于不同的 a，$b \in \mathcal{M}$，我们定义 $a <_{\text{lex}} b$ 如果

$$a(i) < b(i), \quad \textit{where } i = \min\{k \in \sigma \mid a(k) \neq b(k)\}.$$

*Sin*由于 a和 b 具有有限支撑，该最小值存在。

设 $f = \sum_{j \in \mathcal{M}} a_j x^j$ 是关于 $k[x_i]_{i \in \sigma}$的多项式，其中只有有限个系数 a_j 非零；j 称为指数向量，对于任意满足 $a_j \neq 0$的 j，表达式 $a_j x^j$ 称为 f的项。

一旦确定了单项式序，我们就可以定义 f的首项、首项系数和首单项式。这些概念在格罗布纳基理论中发挥着基础作用。

定义 3 (首项)。对于非零多元多项式 f，其首项记作 $\mathrm{LT}(f)$，是 f 中相对于固定单项式序最大的单项式所对应的项。对于零多项式，其首项定义为 0。**定义 4** (首项系数)。多项式 f的首项系数记作 $\mathrm{LC}(f)$，定义为 f的首项的系数。

定义 5 (首单项式)。 固定一个单项式序，多项式 f 的首单项式记作 $\mathrm{LM}(f)$，定义为 f 的支撑集中相对于该单项式序最大的指数向量所对应的单项式。如果 $f = 0$，那么 $\mathrm{LM}(f) = 0$。

尽管 Lean 中没有将首单项式作为专门的定义提供，但如果 $f \neq 0$，则可以使用表达式 monomial (m.degree f) 1 来表示它。我们将首项的形式化推迟到下一节。

3 在 Lean 中形式化格罗布纳基理论

本节将介绍我们对格罗布纳基理论的形式化。经典处理通常考虑有限个变量的多项式环，而 Mathlib 中的多元多项式库则定义在任意指标类型 σ 上，该指标类型不一定是有限的。因此，我们针对具有任意多个变量的多项式环发展格罗布纳基理论。在无限变量的情形下，多项式环通常不是诺特的，理想也不一定具有有限生成集。为此，我们将格罗布纳基的概念扩展到允许无限集，并证明大多数基本性质在这一情形下仍然成立。尽管本节中的主要定理是在域上陈述的，但我们也将若干结果形式化到了更一般的交换环上。我们的形式化所依赖结构的概览见图??。

3.1 带底元的次数

设 $f = \sum_{j \in \mathcal{M}} a_j x^j$ 是一个非零多项式。 f 的次数记为 deg(f)，定义为

$$\deg(f) := \max\{j \in \mathcal{M} \mid a_j \neq 0\}$$

其中最大值是相对于单项式序取得的。在 Mathlib 中，多元多项式的单项式指数类型实现为 `Finsupp` $\sigma\mathbb{N}$，数学上记作 $\sigma \to_0 \mathbb{N}$。这对应于 (1) 中引入的指数类型 $\mathcal{M}$。按照 Mathlib 现有的约定，零多项式的次数定义为 0。虽然这一约定便于对指数向量进行计算，却无法保留次数函数的某些性质。尤其是，加法性质

$$\deg(fg) = \deg(f) + \deg(g)$$

在没有额外假设时并不成立。例如，若 $f = 0$ 和 g 的次数均为正，则 deg(fg) $= 0$，而 deg(f)+deg(g) =deg(g)。为在 Lean 中解决这一问题，我们将次数函数的陪域扩展为 WithBot（$\sigma \to_0 \mathbb{N}$）。这种类型添加了一个底元素 $\bot$，它严格小于每个指数向量。这样就可以将零多项式的次数与常数多项式的次数区分开来：

1. deg(0) $= \bot$，2. deg(c) $= 0$，其中 c 是非零常数。

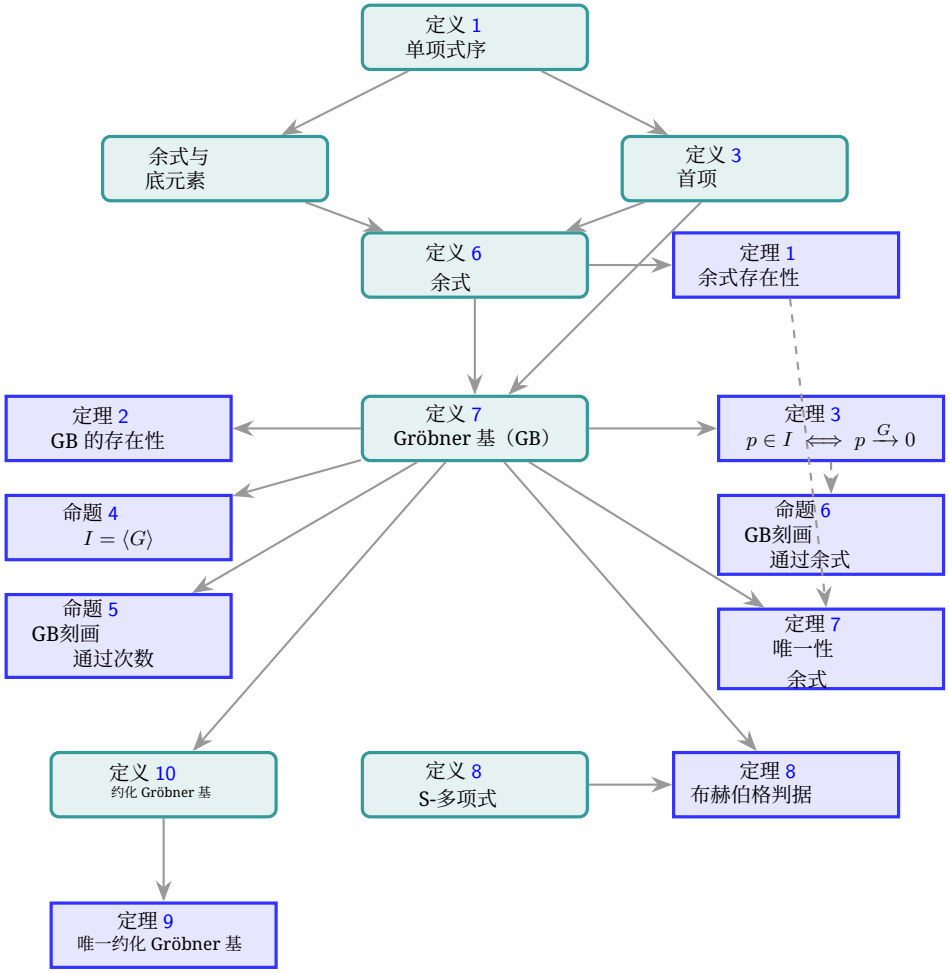

图 1：格罗布纳基形式化的结构

按照这一约定，deg(0) <deg(c)是有效的，并且可以恢复次数的经典加法性质，因为 $\bot + a = \bot$ 对所有数值 a都成立。我们将次数函数形式化为带有一个底元素的函数，如下所示。

```
变量 (m) 在定义 withBotDegree :WithBot(σ →₀ ℕ): =
f.support.image m.toSyn  |>.max.mapm.toSyn.symm
```

尽管 withBotDegree 消除了零多项式次数的歧义，我们有时仍会继续使用现有的次数函数，因为多元多项式和单项式序都是定义在指数向量类型 $\sigma \to_0 \mathbb{N}$ 上，而非附加底元的类型上（$\sigma \to_0 \mathbb{N}$）。因此，使用次数可以直接与现有的 Mathlib API 交互，而无需反复消去附加底元类型的构造子，也无需用 unBot 将 $\bot$转为 0。

3.2 多项式除法与余式

格罗布纳基理论从根本上建立在多项式约化之上。因此，我们首先形式化余式的概念。

定义 6 (余式)。 设 f 是一个多元多项式， B 是域 k上的多元多项式集合，并固定一个单项式序。假设存在函数 $g: B \to k[x_i]_{i\in\sigma}$ ，其具有有限支撑，以及一个多项式 r ，满足

(1) $f = \sum_{b\in B} g(b)\, b + r$，*(2)* 任意 $g(b)b$ 的次数，其中 $b \in B$ 小于或等于 f的次数。
(3) r 的任何项都不能被 B中任意非零元素的首单项式整除。

那么， r 称为 f 除以 B所得的余式。我们也将 f 除以集合 B中的多项式所得的余式记作 $f \xrightarrow{B} r$ 。

```
定义 是余式 :=(∃ (g : B  →₀ MvPolynomial  σ R), f
 =                          _              ↦          +   ∧
 ∀                                                  +
Finsupp.linearCombination (fun (b : B) b.val) g r (b :B),m.toWithBotSyn
(m.withBotDegree b.val) m.toWithBotSyn(m.withBotDegree (g b))  ≤     ∧
m.toWithBotSyn(m.withBotDegree f)) ∀   ∈            ∀   ∈            c r.support,b
B, b  ≠ 0 → ¬ (m.degree b  ≤ c)
```

尽管实际计算通常涉及有限个多项式，但余式的概念最自然的定义方式是相对于一个抽象多项式集合。为联系这两种观点，我们建立以下定理。

```
定理是余式_范围 {ι :  类型*} (f : MvPolynomial  σ R)(b:  ι → MvPolynomial  σ R) (r :
MvPolynomial  σ R) : m.IsRemainder f (Set.range b) r  ↔(∃ g :  ι →₀ MvPolynomial  σ R,
f  =                          _     +   ∧ Finsupp.linearCombination b gr i : ,
 ∀     ι                              ≼ [
m.withBotDegree (b i * g i) 'm
                                        ]                    ∧ m.withBotDegree f) c
∀   ∈            ∀     ι      ≠ 0 → ¬                 ≤
r.support, i : ,b i (m.degree (b i) c)
```

为了便于实际形式化，我们建立该定理的一个变体，它针对有限指标集进行表述，从而简化其在具体开发中的使用。

```
定理是余式_范围_有限集 {ι :  类型*}  [有限类型  ι] (b :  ι → MvPolynomial  σ R)(r:
MvPolynomial  σ R) : m.IsRemainder p (Set.range b) r  ↔(∃g:  ι →MvPolynomial σ R,
p  = Σ     ι                +   ∧ i : , (b i * g i) r i : ,m.degree(b i * g i)
 ∀     ι
                                  ≼[ ]              ∧ m m.degree p) c r.support,  i : , b i
∀   ∈            ∀     ι      ≠ 0 → ¬
(m.degree (b i)  ≤ c)
```

定理 1 (余式的存在性)。 固定一个单项式序。对于任意多项式$f \in k[x_i]_{i\in\sigma}$ 以及任意有序元组的多项式 $B = (f_1, \ldots, f_s)$ 属于 $k[x_i]_{i\in\sigma}$，存在多项式 $a_1, \ldots, a_s \in k[x_i]_{i\in\sigma}$ 和一个多项式 $r \in k[x_i]_{i\in\sigma}$ 使得

$$f = a_1 f_1 + \cdots + a_s f_s + r,$$

且 r 是 f 关于集合 $\{f_1, \ldots, f_s\}$ 的余式，符合定义6。

```
定理存在_isRemainder {B : Set (MvPolynomial  σ R)} (hB :  ∀ b  ∈ B, 是单位  <|
m.leadingCoeff b) (p : MvPolynomial  σ R) : ∃ (r : MvPolynomial  σ R), m.IsRemainder
p B r
```

余式的存在性是多项式约化的基础性质。

3.3 格罗布纳基

现在我们转向在 Lean 中对格罗布纳基进行形式化。在前几节介绍的多元多项式、单项式序和多项式约化等概念的基础上，我们将引入格罗布纳基的形式定义及其基本性质。首先，我们将形式化定义于定义 3中的首项。

```
不可计算定义 leadingTerm(f:MvPolynomial σ R):MvPolynomial σ R:= monomial(m.degree
f)(m.leadingCoefff)
```

有了首项的概念后，我们现在可以定义格罗布纳基。

定义 7 (GröbnerBasis)。 固定多项式环上的一个单项式序 $k[x_i]_{i\in\sigma}$。如果子集 G 属于理想 $I \subseteq k[x_i]_{i\in\sigma}$，并且 I 中各元素的首项 G 生成 I的首项理想，则称其为

$$\langle \mathrm{LT}(G)\rangle = \langle \mathrm{LT}(I)\rangle.$$

其中，$\mathrm{LT}(G)$ 和$\mathrm{LT}(I)$分别表示 G 和 I 中元素的首项集合。

```
定义 IsGroebnerBasis {R:Type*} [CommSemiring R](G:Set(MvPolynomial σ R))(I : Ideal
(MvPolynomial σ R)) : = G ⊆ I ∧ 理想生成 (m.leadingTerm" ↑I) = 理想生成
(m.leadingTerm " G)
```

在确立格罗布纳基的定义后，我们转而讨论它们是否存在的问题。虽然该定义在形式上具有一般性，但任意理想存在有限格罗布纳基，从根本上说依赖于多项式环的诺特性质。因此，下面的定理针对域上具有无穷多个变量的多项式环陈述，从而确保这类基总是存在。

定理 2. 在 R 上固定一个单项式序，其中 $R = k[x_1, \ldots, x_n]$。那么，对于任意理想 $I \subseteq R$，都存在一个有限集合 $G = \{g_1, \ldots, g_t\} \subset I$，使得 G 是 I 的格罗布纳基。

```
定理 存在_是格罗布纳基_有限 [有限 σ] : ∃G :Finset (MvPolynomial σ
k), 是格罗布纳基 m G ↑I
```

除了存在性结果外，格罗布纳基在解决理想成员关系问题中也发挥着核心作用。定理 3。

定理3. 在多项式环 $k[x_i]_{i\in\sigma}$上固定一个单项式序。设 I 是一个理想，且 $G \subseteq I$。若 G 是 I的格罗布纳基，那么对于任意多项式 p，p 除以 G 所得的余式为零，当且仅当 $p \in I$。

```
定理是余式_零_当且仅当_属于_理想 {p : MvPolynomial σ R}{G :
Set(MvPolynomial σ R)} {I : 理想 (MvPolynomial σ R)} (hG: 对 ∀ g
 ∈ G，首项系数 m.leadingCoeff g 是单位) (h: m 是格罗布纳基 G I): m 是
余式 pG 0 ↔ p ∈ I
```

由于每个多项式 $p \in I$ 都能化简为零的余式，因此 p 可以表示为 G中元素的线性组合。这就得到命题4。

命题 4。 设 I 是一个理想， G 是多项式环 $k[x_i]_{i\in\sigma}$ 的子集。若 G 是 I 的格罗布纳基，则有 $I = \langle G \rangle$.

```
定理理想_eq_span {G : Set (MvPolynomial σ R)}{I: Ideal (MvPolynomial σ R)}
(hG : ∀ g ∈ G,IsUnit (m.leadingCoeff g)) (h : m.IsGroebnerBasis G I) : I
 = Ideal.span G
```

为了关联代数定义与约化行为，我们建立以下等价性，用理想生成和约化来刻画格罗布纳基。这是迈向基于余式的判据的关键一步。

命题 5。在多项式环 $k[x_i]_{i\in\sigma}$ 上固定一个单项式序。设 I 是一个理想，且令 $G \subseteq I$。那么 G 是 I 的格罗布纳基，当且仅当 $I = \langle G \rangle$ ，并且对于每个非零多项式 $p \in I$，都存在 $g \in G$ ，使得 $\mathrm{LM}(g) \mid \mathrm{LM}(p)$。

```
定理 是格罗布纳基_当且仅当_子集_且_次数_小于等于_等于_且_次数_小于等于 (G : Set (MvPolynomial σ R)) (I :
Ideal (MvPolynomial σ R)) (hG : ∀ g ∈ G, 是单位 (m.首项系数 g)) : m.是格罗布纳基 G I ↔理想生成 G =
I ∧ ∀ p ∈ I, p ≠ 0 → ∃ g ∈ G, m.次数 g ≤ m.次数 p
```

以下命题通过余式刻画格罗布纳基，并提供了一种判定理想成员关系的方法。

命题 6。 在多项式环 $k[x_i]_{i\in\sigma}$上固定一个单项式序。设 I 是 $k[x_i]_{i\in\sigma}$的一个理想。集合 G 是 I 的格罗布纳基，当且仅当 $G \subseteq I$ ，并且对每个 $f \in I$， f 除以 G 所得的余式都等于0。

```
定理 是格罗布纳基_当且仅当_包含于_理想_并且_是余式_zero(G: Set (MvPolynomial σ
R)) (I : Ideal (MvPolynomial σ R)) (hG : ∀ g ∈ G,IsUnit (m.leadingCoeff
g)) : m.IsGroebnerBasis G I ↔ G ⊆ I ∧ ∀ p ∈ I, m.IsRemainder p G 0
```

最后，格罗布纳基的一个重要性质是，它们能够为模掉其生成的理想后的元素提供唯一余式。与一般的多元除法不同，在一般的多元除法中，余式通常取决于除数的顺序，而用格罗布纳基进行除法则能保证得到唯一余式。

定理7. 在多项式环 $k[x_i]_{i\in\sigma}$ 上固定一个单项式序。令 I 为一个理想，并令 $G = \{g_1, \ldots, g_t\}$ 为 I 的格罗布纳基。那么，对于任意多项式 p， p 除以 G 所得的余式是唯一的。

```
定理 存在唯一_是余式 {G: Set (MvPolynomial σ R)} {I : 理想 (MvPolynomial σ
R)} (h : m.是格罗布纳基 G I) (hG : ∀ g ∈ G, 是单位 (m.首项系数 g)) (p :
MvPolynomial σ R) : ∃! (r : MvPolynomial σ R), m.是余式 p G r
```

3.4 布赫伯格判据

为了在 Lean 4 中形式化验证布赫伯格判据，我们首先引入 S-多项式。我们对 S-多项式的定义与 [20, 21]中的标准表述略有不同。我们采用这一变体，以便在 Lean 中获得更自然的实现，尤其是避免假设首项系数可逆。

定义8 (*S*-多项式)。 固定一个单项式序。设 f， $g \in k[x_i]_{i\in\sigma}$。定义*S*-多项式如下：

$$\mathrm{S}(f,g) = \frac{\mathrm{LC}(g)x^\gamma}{\mathrm{LM}(f)} \cdot f - \frac{\mathrm{LC}(f)x^\gamma}{\mathrm{LM}(g)} \cdot g,$$

其中， x^γ 表示$\mathrm{LM}(f)$ 与$\mathrm{LM}(g)$的最小公倍数。

由于这个表述与标准定义的区别仅在于乘以一个非零标量因子，该因子源自消去首项系数，

$$\mathrm{S}(f,g) = \frac{x^\gamma}{\mathrm{LT}(f)}\, f - \frac{x^\gamma}{\mathrm{LT}(g)}\, g,$$

这不会影响布赫伯格判据的有效性。现在，我们按照定义 *S*给出 8 中多项式的正式定义。

```
不可计算定义 sPolynomial(fg:MvPolynomial σ R):MvPolynomial σ R :=
monomial(m.degree g-m.degreef)(m.leadingCoeffg)*f- monomial(m.degree
f-m.degreeg)(m.leadingCoefff)*g
```

在证明布赫伯格判据之前，我们先建立一个刻画首项抵消的引理。如果两个非零多项式具有相同的首次数，那么它们的 *S*-多项式的次数严格更小。更一般地，如果若干个具有相同首次数的多项式之和出现首项抵消，那么该和可以表示为它们的 *S*-多项式的线性组合，其系数属于 k。

```
引理 sPolynomial_decomposition_of_degree_sum_smul_le₀ {R} [CommRing R]{d : m.syn} {ι: Type*} {B :
Finset ι}{c: ι → R} {g : ι →MvPolynomial σ R} (hd : ∀b ∈B, (m.toSyn <|m.degree <|g b) =d ∧ 是单
位 (m.leadingCoeff <| g b) ∨g b = 0) (hfd : (m.toSyn <|m.degree <| Σ b ∈ B, c b ·g b) <d) :∃ (c' :
ι → ι → R), Σ b ∈ B, cb · g b = Σ b₁ ∈ B, Σ b₂ ∈ B, (c' b₁ b₂) · m.sPolynomial (g b₁) (g b₂)
```

接下来，我们尝试形式化布赫伯格判据，该判据通过约化临界对的 *S*-多项式来刻画格罗布纳基。这将关于无限集合 $\langle G\rangle$ 的全称量词，替换为相对于 G的有限次检查。

定理 8 (布赫伯格判据)。 在多项式环$k[x_i]_{i\in\sigma}$上固定一个单项式序。集合 G 是理想 $\langle G\rangle$ 的格罗布纳基，当且仅当对所有 f， $g \in G$， 0是 *S*-多项式 $\mathrm{S}(f,\ g)$ 除以 G所得的余式。

例 1。 考虑采用字典序$x_0 > x_1 > x_2$的 *Cyclic–3* 理想 $I \subset \mathbb{Q}[x_0, x_1, x_2]$ ， 其生成元为

$$\{\, f_1 = x_0 + x_1 + x_2, \quad f_2 = x_0x_1 + x_1x_2 + x_2x_0, \quad f_3 = x_0x_1x_2 - 1 \,\}.$$

I 的一个格罗布纳基为

$$G = \{\, g_1 = x_0 + x_1 + x_2, \quad g_2 = x_1^2 + x_1x_2 + x_2^2, \quad g_3 = x_2^3 - 1 \,\}. \tag{2}$$

根据布赫伯格判据，只需检查对所有的 $g_i, g_j \in G$， *S*- 多项式 $\mathrm{S}(g_i, g_j)$ 模 G 都能约化为 0 。举例来说，考虑 (g_1, g_2)。我们

有 $\mathrm{LM}(g_1) = x_0$，$\mathrm{LM}(g_2) = x_1^2$，且$\mathrm{lcm}(\mathrm{LM}(g_1), \mathrm{LM}(g_2)) = x_0x_1^2$，因此

$$\begin{aligned}\mathrm{S}(g_1, g_2) &= \frac{x_0x_1^2}{x_0}\, g_1 - \frac{x_0x_1^2}{x_1^2}\, g_2 \\ &= x_1^2(x_0 + x_1 + x_2) - x_0(x_1^2 + x_1x_2 + x_2^2) \\ &= -x_0x_1x_2 - x_0x_2^2 + x_1^3 + x_1^2x_2 \\ &= (-x_1x_2 - x_2^2)g_1 + (x_1 + x_2)g_2.\end{aligned}$$

因此，$\mathrm{S}(g_1, g_2)$ 模 G 可约化为 0 。其余配对也以类似方式进行检查。现在我们将 Buchber

ger 判据，如定理8.所述。

```
定理是格罗布纳基_当且仅当_理想_等于_生成_并且_是格罗布纳基_生成₀ {R}[交换环 R] {I : 理想
(MvPolynomial σ R)} {G : Set (MvPolynomial σ R)} (hG : ∀ g ∈ G, IsUnit (m.首项系数 g) ∨ g
= 0) : m.IsGroebnerBasis G I ↔ = ∧ I 理想生成 G (g₁ g₂ :
∀
G), m.IsRemainder (m.sPolynomial g₁ g₂ : MvPolynomial σ R) G 0
```

3.5 约化格罗布纳基

由于格罗布纳基的定义只要求其元素的首项生成首项理想 $\langle \mathrm{LT}(I) \rangle$，因此可能出现冗余生成元。例如，如果 $g_1, g_2 \in G$，$\mathrm{LM}(g_1)$ 整除 $\mathrm{LM}(g_2)$，那么 g_2 不会带来任何新信息。引入极小格罗布纳基的概念，旨在消除这类冗余生成元。

定义 9 (极小格罗布纳基)。 在多项式环 $k[x_i]_{i\in\sigma}$上固定一个单项式序。子集 $G \subseteq I$ 称为理想 I 的极小格罗布纳基，当且仅当满足以下条件：

(1) G 是 I的格罗布纳基。*(2)* 每个 $p \in G$ 都是首一的。*(3)* 对所有互不相同的 p和 $q \in G$，$\mathrm{LM}(p) \nmid \mathrm{LM}(q)$。

现在我们按照定义 9，给出极小格罗布纳基概念的 Lean 形式化：

```
定义是格罗布纳基.IsMinimal (hG : m.是格罗布纳基 G I) := (∀p ∈ G,
m.首一 p) ∧ (∀p ∈ G, ∀q ∈G,q ≠ p → ¬m.次数 q ≤ m.次数 p)
```

例 1 (续)。 可以验证，$G = \{g_1, g_2, g_3\}$ *(2)*满足定义 *9*中的条件，因此是 I的极小格罗布纳基。

极小格罗布纳基虽然去除了冗余生成元，但对于固定的理想和单项式序，它不一定是唯一的。同一个理想可能有不同的极小格罗布纳基。为了得到规范表示，我们施加一个更强的条件：基中任意多项式所含的单项式，都不能被该集合中其他多项式的首单项式整除。由此便得到约化格罗布纳基的概念。

定义 10 (约化格罗布纳基)。 在多项式环 $k[x_i]_{i\in\sigma}$上固定一个单项式序。子集 $G \subseteq I$ 称为理想 I 的约化格罗布纳基，当且仅当满足以下条件：

(1) G 是 I的格罗布纳基。*(2)* 每个 $p \in G$ 都是首一的。*(3)* 对所有互不相同的 p、$q \in G$，p 中出现的任何单项式都不能被首单项式(q)整除。

现在，我们按照定义 10，在 Lean 中形式化约化格罗布纳基的概念:

```
定义 是格罗布纳基.是约化的 (hG : m.是格罗布纳基 G I): = (∀p ∈G, m.首一 p)
 ∧ ∀p ∈G, m.是余式 p (G \ {p}) p
```

例 1 （续）。 不难验证，

$$G' = \{g_1', g_2, g_3\}, \qquad g_1' = x_0 + x_1^2 + x_1x_2 + x_1 + x_2^2 + x_2,$$

也是同一理想 I 的极小格罗布纳基。然而， G' 并非约化的，因为g_1' 包含单项式 x_1^2，它可被 $\mathrm{LM}(g_2) = x_1^2$整除。将 g_1' 用 g_2 约化，即令 $g_1 = g_1' - g_2$，即可移除这个可约项，并得到约化格罗布纳基 $G = \{g_1, g_2, g_3\}$ （见 (2)）。

定义 10中的更强条件给出了约化格罗布纳基的以下基本性质。

定理 9. 在多项式环 $k[x_i]_{i\in\sigma}$上固定一个单项式序。每个理想$I \subseteq k[x_i]_{i\in\sigma}$ 都有唯一的约化格罗布纳基。

```
定理IsReduced.uniqueExists{k} [域 k](I : 理想 (MvPolynomial σ k)) : ∃! (B:Set
(MvPolynomial σ k)), ∃(h : m.是格罗布纳基 B I), h.IsReduced
```

下面这个版本适用于一般交换环，而非域，并额外假设格罗布纳基的首项系数是单位。

```
定理 IsReduced.uniqueExists_的_isGroebnerBasis {R} [非平凡 R] [交换环 R]{G :
Set(MvPolynomial σ R)} {I : 理想 (MvPolynomial σ R)} (hG : m.是格罗布纳基 G I) (hG' :
 ∀ g ∈ G, 是单位 (m.首项系数 g)) : ∃! (B : Set (MvPolynomial σ R)), ∃ (h : m.是格罗
布纳基 B I), h.IsReduced
```

4 无限格罗布纳基的有限刻画

本节将说明，如何通过有限变量约化来刻画含有无限多个变量的多项式环中的格罗布纳基。设 σ 和 σ' 为变量指标类型，并记两个多项式环为

$$S = k[x_i]_{i\in\sigma}, \qquad S' = k[x_i]_{i\in\sigma'}.$$

设 $\mathcal{M}$:= $(\sigma \to_0 \mathbb{N})$ 和 $\mathcal{M}'$:= $(\sigma' \to_0 \mathbb{N})$ 为相应的指数向量类型，并分别将 $\mathcal{M}$ 和 $\mathcal{M}'$ 上的单项式序记为 $(\mathcal{M}, \leq_{\mathcal{M}})$ 和 $(\mathcal{M}', \leq_{\mathcal{M}'})$。要关联这两种情形，一个关键前提是单项式序的兼容性。现在我们引入单项式序嵌入的概念。定义 11（单项式序嵌入）。给定两个单项式序 $(M, \leq_m)$ 和 $(M', \leq_{m'})$，函数 $f\colon \sigma' \to \sigma$ 是一个序嵌入，当且仅当 f 满足

(1) 函数 f 是单射。

(2) 将经由 f 从 $\mathcal{M}'$ 到 $\mathcal{M}$ 的提升记为 f' : $\mathcal{M}' \to \mathcal{M}$。对于所有 u、 $v \in \mathcal{M}'$, 若$u \leq_{m'} v$，则 $f'(u) \leq_m f'(v)$ 。

我们在 Lean 中将定义 11形式化如下:

```
结构体 Embedding 扩展 σ' ↪ σ 其中 monotone': Monotone (m.toSyn ∘
Finsupp.mapDomain toFun ∘m'.toSyn.symm)
```

给定 $\mathcal{M}$ 上的一个单项式序，任意单射映射 f: $\sigma' \to \sigma$ 都会在 $\mathcal{M}'$ 上诱导出一个单项式序。这个单项式序是通过沿着由 f 诱导的指数向量嵌入，将 $\mathcal{M}$ 上的单项式序拉回得到的

```
打开ofInjective 在 noncomputable def ofInjective{σ': Type*} {f: σ' → σ} (hf:
f.Injective): MonomialOrder σ' := { syn := Syn m f toSyn: (σ' →₀ ℕ) ≃+ (Syn
m f) := toSyn' m f ... }
```

有了这一构造， $\mathcal{M}'$ 上诱导出的单项式序可以自然嵌入到 $\mathcal{M}$ 上的单项式序中。

```
定义Embedding.ofInjective {f: σ' → σ} (hf: f.Injective):
Embedding (m.ofInjective hf) m其中toEmbedding := ⟨f, hf⟩
monotone' := ...
```

引理 1. 设 $p \in S'$、 $r \in S$，且 $B \subseteq S'$ 是一个多项式集合。若 r 是 p 在 S 中被 B 除法所得的余式，则将 r 中 σ' 以外的变量设为 0 所得到的限制 r'，是 p 在 S' 中被 B 除法所得的余式。

```
引理 MonomialOrder.Embedding.isRemainder_killCompl_的_是余式_rename{σ' σ}{m' :
MonomialOrder σ'} {m : MonomialOrder σ} (e : 嵌入 m' m) {p : MvPolynomial σ' R}
{B : Set (MvPolynomial σ' R)} {r: MvPolynomial σ R} (h : m.是余式 (p.rename e)
(rename e '' B) r) : m'.是余式 p B (r.killCompl e.coe_injective)
```

下面的引理表明，余式在嵌入下是不变的。

引理 2. 设 p、 $r \in S'$，且 $B \subseteq S'$ 是一个多项式集合。那么， r 是 p 除以 B 在 S' 上的余式，当且仅当 r 是 p 除以 B 在 S 上的余式。

```
引理MonomialOrder.Embedding.isRemainder_当且仅当_是余式_重命名 {σ' σ} {m' : 单项式
序 σ'} {m : 单项式序 σ} (e: 嵌入 m' m) (p : MvPolynomial σ' R) (B : Set
(MvPolynomial σ' R))(r: MvPolynomial σ' R) : m'.是余式p B r ↔ m.IsRemainder
(p.rename e) (MvPolynomial.rename e '' B) (r.rename e)
```

格罗布纳基也具有不变性。

引理 3. 设 $G \subseteq S'$ 是一个多项式集合， $I \subseteq S'$ 是一个理想。那么，G 是 I 在 S' 上的格罗布纳基，当且仅当 G 是由 I 生成的理想在 S 上的格罗布纳基。

```
引理 MonomialOrder.Embedding.isGroebnerBasis_当且仅当_是格罗布纳基_重命名 {σ'}{m':
MonomialOrder σ'}{m:MonomialOrder σ} (e: 嵌入 m' m) (G : Set (MvPolynomial σ' R)) (I: 理想
(MvPolynomial σ' R)) : m'. 是格罗布纳基 G I ↔m.IsGroebnerBasis (MvPolynomial.renamee''
G) (I.map(MvPolynomial.renamee))
```

一般而言，对于具有无限多个变量的多项式环中的理想，有限格罗布纳基未必存在，因为此类环不是诺特环。不过，利用上述引理，可以将问题限制到由某个有限生成集中的变量生成的子环上，从而得到有限格罗布纳基；该子环是诺特环。随后可以将这个基提升到原多项式环。特别地，一个理想是有限生成的，当且仅当它存在有限格罗布纳基。

引理 4. 设 $I \subseteq k[x_i]_{i\in\sigma}$ 是一个有限生成理想。那么 I 存在有限格罗布纳基。

```
引理存在_是格罗布纳基_有限_的_存在_生成_有限 {B : Set (MvPolynomial σ k)} (hB : B.Finite) :∃
G :Set (MvPolynomial σ k),m.IsGroebnerBasisG(Ideal.spanB) ∧ G.Finite
```

在 [16, 定理 1.12], Iima 和 Yoshino 通过有限变量子环上约化格罗布纳基的下极限，对可数多个变量的多项式环上的约化格罗布纳基进行了刻画。我们利用滤子上的下极限构造，将这一结果推广到任意指标类型。

定理 10. 固定一个理想 $I \subseteq k[x_i]_{i\in\sigma}$和一个指标集 ι，令 $F \subseteq \mathcal{P}(\iota)$ 是格 $(\mathcal{P}(\iota) \subseteq)$ 上的一个滤子，使得 $\emptyset \notin F$。假设对所有 $j \in \iota$， $\sigma_j \subseteq \sigma$ 以及 G_j 都是理想 $I \cap k[x_i]_{i\in\sigma_j}$ 在 $k[x_i]_{i\in\sigma_j} \subseteq k[x_i]_{i\in\sigma}$上的约化格罗布纳基。若 $\bigcup_{s\in F}\bigcap_{j\in s}\sigma_j = \sigma$，则

$$\bigcup_{s\in F}\bigcap_{j\in s} G_j$$

I的约化格罗布纳基。

```
定理 IsReduced.isReduced_liminf {k} [域 k] {I : 理想 (MvPolynomial σ k)} {α} {σ' : α →
Type*} {m' : (a : α) → 单项式序 (σ' a)} {f : Filter α} [inst_neBot : f.NeBot] {e : (a : α) →
(m' a).嵌入 m} (hI : Set.univ =f.liminf (fun x ↦Set.range (e x))) {G' : (a: α) → Set
(MvPolynomial (σ' a) k)} (hG' : ∀ a, (m' a).是格罗布纳基 (G' a) (I.comap <| rename (e a)))
(hG'' : ∀ a, (hG' a).IsReduced) : ∃ h : m.是格罗布纳基 (f.liminf fun a ↦ rename (e a) '' G' a)
I, h.IsReduced
```

特别地，令 σ 和 ι 为 $\mathbb{N}$， F为 $\mathbb{N}$上的余有限滤子， σ_n 为 $[0,n]\cap\mathbb{N}$，且G_n 为 $I \cap k[x_i]_{i\in[0,n]\cap\mathbb{N}}$的约化格罗布纳基，则

$$\bigcup_{\substack{n\in\mathbb{N},\\ [n,+\infty)\subseteq A\subseteq\mathbb{N}}}\ \bigcap_{i\in A} G_i = \bigcup_{n\in\mathbb{N}}\bigcap_{i\geq n} G_i$$

是 I [16,定理 1。12]。

5 结论与未来工作

目前，我们的形式化基于 Mathlib 中的 MvPolynomial。这一结构支持以任意类型为索引的多项式环，具有很强的数学通用性，但并非为高效的可执行代数计算而设计。因此，我们目前的开发侧重于理论验证，在这项工作中，

我们并未形式化实现可执行的 Buchberger 算法。未来的工作将致力于弥合抽象理论与具体计算之间的差距。具体而言，我们计划结合外部计算机代数系统与我们的形式理论，实现对多项式恒等式的基于证书的验证。在这一流程中，格罗布纳基由外部计算得出，然后在 Lean 中通过我们对布赫伯格判据的形式化进行验证。这种方法避免了直接使用 MvPolynomial 时固有的计算限制，同时保留了完整的机器检查能力。

致谢

Hao Shen、Junqi Liu 和 Lihong Zhi 获得中国国家重点研发计划（项目编号：2023YFA1009401）的资助。Junyu Guo 在导师 Xishun Zhao 的鼓励和支持下参与了本项目。我们衷心感谢 Lean社区 的几位成员，他们审阅了我们提交至 Mathlib 的形式化工作，并提供了宝贵的反馈和具体的改进建议。按姓氏字母顺序，我们感谢 Dagur Asgeirsson、Riccardo Brasca、Snir Broshi、Thomas Browning、Bryan Gin-ge Chen、Johan Commelin、Jovan Gerbscheid、Aaron Liu、Jireh Loreaux、Bhavik Mehta、Kyle Miller、Ruben Van de Velde、Eric Wieser 和 Andrew Yang 的慷慨帮助与支持。特别是，Snir Broshi 建议使用 `Set` 而非 `Finset`来形式化格罗布纳基，这促使我们考虑无限Gröbner基特有的性质。此外，还有许多贡献者为我们所依赖的 Lean 生态系统构建了基础设施，其中包括 Antoine Chambert-Loir，他形式化了多元多项式除法的性质，而我们直接利用这些性质证明定理 1。

参考文献

[1] Buchberger，B.：《用于求取零维多项式理想之剩余类环基元素的算法》。博士论文，数学研究所，因斯布鲁克大学（1965）[2]Nipkow，T.、Wenzel，M.、Paulson，L.C.：《Isabelle/HOL：高阶逻辑证明助手》。施普林格，纽约（2002）[3]Faugere，J.-C.：一种计算格罗布纳基（F4）的高效新算法。《纯粹与应用代数杂志》**139(1-3)**，61–88（1999）[4] Maletzky，A.：基于签名的Gröbner基算法的通用且可执行的形式化。《符号计算杂志》**106**，23–47（2021）[5] Maletzky，A.：Isabelle/HOL中的格罗布纳基与Macaulay矩阵。《计算形式化》（2024）[6] Maletzky，A.、Immler，F.：Isabelle/HOL中的模的格罗布纳基与Faugère的F 4算法。载于：智能计算国际会议

Mathemati科学，页码 183–193（2018）. Springer

[7] Bertot, Y., Castéran,P.：交互式定理证明与程序开发：Coq'Art：归纳构造演算。施普林格，纽约（2013）

[8] Persson, H.：在 Coq 中集成开发 Buchberger 算法。博士论文，法国国家信息与自动化研究院（2001年）

[9] Théry, L.：经机器验证的 Buchberger 算法实现。《自动推理杂志》**26**(2), 107–137（2001年）https://doi.org/10.1023/A:1026518331905

[10]Medina-Bulo, I., Palomo-Lozano, F., Ruiz-Reina, J.-L.：在 ACL2 中经验证的 Buchberger 算法 Common Lisp 实现。《符号计算杂志》**45**(1), 96–123（2010）

[11] Schwarzweller, C.：Mizar 系统中的格罗布纳基 – 理论精化。载于：国际数学知识管理会议，页码 299–314（2005）。施普林格

[12] Dermitzakis, M.：《Lean 中 Buchberger 算法的实现》（2019）

[13] Poulsen, A.B.：L∃∀N 中的数学项目（2023年）

[14] Junyu, G.：Hagb/lean-groebner：格罗布纳基的 Lean 4 形式化。https://github.com/Hagb/lean-groebner (2023年)

[15] Sage 开发者：SageMath，即 Sage 数学软件系统。（2022年）。数字对象标识符 10.5281/zenodo.6259615。https://www.sagemath.org

[16]Iima, K., Yoshino, Y.：无限多个变量的多项式环的格罗布纳基及其应用。《代数通讯》**37**(2008)https://doi.org/10.1080/00927870802502878

[17]Moura, L., Kong, S., Doorn, F., Raumer, J.：Lean 定理证明器（系统描述），第 9195 卷，第 378–388 页（2015年）。https://doi.org/10.1007/978-3-319-21401-6 26_

[18] Moura, L., Ullrich, S.：Lean 4 定理证明器与编程语言。载于：自动演绎——CADE 28：第 28 届国际自动演绎会议，线上活动，2021年7月12日至15日，会议论文集，第 625–635 页。施普林格，柏林，海德堡（2021）。https://doi.org/10.1007/978-3-030-79876-5_37 。https://doi.org/10.1007/978-3-030-79876-5_37[19] mathlib 社区：Lean 数学库。载于：CPP2020，第 367–381 页。计算机协会，纽约，NY，美国（2020）。https://doi.org/10.1145/3372885.3373824 。https://doi.org/10.1145/3372885.3373824

[20] Becker, T., Weispfenning, V.：格罗布纳基。载于：格罗布纳基：交换代数的计算方法，第 187–242 页。施普林格，纽约（1993年）

[21] Cox, D.A.、Little, J.、O'Shea, D.：《理想、簇与算法——计算代数几何与交换代数导论》，第5版。《数学本科生教材》，第656页。施普林格，Cham（2025年）。https://doi.org/10.1007/978-3-031-91841-4 。https://doi.org/10.1007/978-3-031-91841-4