SageMath 中的 Drinfeld 模

David Ayotte[a]、Xavier Caruso[b]、Antoine Leudière[c]、Joseph Musleh[d]

摘要

我们展示了首个完全整合到 SageMath 生态系统中的 Drinfeld 模实现。首批功能将随 SageMath 10.0 一同发布。

arXiv:2305.00422v1 [cs.SC] 30 Apr 2023

对类域论的探索长期以来一直是人们的梦想，Kronecker 本人也曾怀抱这一梦想。1854年，他通过提出 Kronecker-Weber 定理，为这一领域作出了重大贡献。该定理指出，每个阿贝尔数 $\mathbb{Q}$域都可以由一个分圆扩张生成。类似地，虚二次数域的扩张也可以借助所谓的 Hilbert 类域 [CLRS09]来描述。该领域的许多重要结果都由 Hilbert 和 Kronecker 提出猜想。其中一些直到二十世纪才由 Takagi、Artin 和 Chevalley 等数学家证明 [CLRS09]。时至今日，描述数域扩张的一般性探索依然困难重重。但如果函数域中的探索更容易呢?

1974年，Drinfeld 提出了如今所称的 Drinfeld 模[Dri74], , 其思想源自 Carlitz [Car35]。借助 Drinfeld 模，可以为函数域建立显式类域论：每个 Drinfeld 模都可赋予一个秩；秩为1 的 Drinfeld 模所对应的分圆函数域由其扭点空间生成，而秩为f的不变量2 Drinfeld 模则参与构造 Hilbert 类域的函数域类比。后来，Drinfeld 模在 Lafforgue 证明若干关于函数域的 Langlands 猜想时发挥了关键作用 [Laf02]。数域中的类比问题至今仍未解决。

近年来，已有纯算法方向的论文[Car18] 以及论 [CGS20、MS19、MS23、LS22] 相继发表，着重强调效率。本实现最初是为了满足对统一、实用的操作工具的需求，我们希望它能服务于广大的用户群体。我们特别注重为代码配备示范性文档，并尽可能利用 SageMath 现有的功能。我们的三项核心原则是可靠性、优雅的用户界面和集成。原始工单（见GitHub拉取请求#35026）于2022年4月创建，并于2023年3月合并。此后，人们陆续提出了许多拉取请求，以增强最初贡献的功能；相关工作仍在积极开展，推动着人们对 Drinfeld 模日益增长的兴趣。

数学背景。 在进入本次报告的核心内容之前，我们需要回顾 Drinfeld 模的基本定义。设 $\mathbb{F}_q$ 是一个含q 个元素的有限域，设K是 $\mathbb{F}_q$ 的一个扩张，并设K 是K的代数闭包。此外，我们赋予K一个 $\mathbb{F}_q[T]$-域结构，这意味着我们给出一个 $\mathbb{F}_q$-代数态射γ： $\mathbb{F}_q[T] \to K$。记 τ 为K 上的 $\mathbb{F}_q$-线性自同态，其定义为$x \mapsto x${v18>。我们将K 上的 *Ore* 多项式环$\{\tau\}$定义为这样一个环，其元素是形如$a_0 + a_1\tau + \cdots + a_n\tau^n$ 的和，其中$n \in \mathbb{Z}_{\geqslant 0}$且对所有满足 0 6$i6n$的$a_i \in$，都有$a_i \in K$。在$K\{\tau\}$中，只要$a \in K$，就有恒等式 $\tau a = a^q\tau$。

定义在K 上的一个 *Drinfeld* 模 是一个 $\mathbb{F}_q$-代数态射 ϕ: $\mathbb{F}_q[T] \to K\{\tau\}$，满足 $\phi(T)$ 的常数项为 $\gamma(T)$，且其在 τ 中的次数非零。注意， $\phi(T)$ 完全决K定了 ϕ；我们通常将其简记为 ϕ_T。模这个名称源于 ϕ 赋予K一个由 $\mathbb{F}_q[T]$,定义的作用：对所有属于 $\mathbb{F}_q[T]$ 的 a 以及属于 K 的 x， 令 $a \cdot x = \phi(a)(x)$。

给定两个 Drinfeld 模 ϕ, ψ，一个态射 $\phi \to \psi$是一个 Ore 多项式$u \in K\{\tau\}$，满足$u\phi_T = \psi_T u$。非零态射称为同源 。

1 实现方案

启动这个项目时，我们的目标是为 Drinfeld 模提供一个通用且文档完善的软件包，面向该领域的从业数学家。我们的主要关注点是开发一款易于使用的软件，涵盖理论的所有基础内容，而不专注于某个特定应用。

1.1 为什么选择 SageMath? 我们选择在 SageMath 中实现我们的软件包，原因有几个。首先，SageMath 是基于广泛使用的 Python 编程语言构建的数学计算工具。作为一款免费开源软件，它受益于来自不同背景的数学家的贡献。凭借通用的设计理念，它还支持广泛的数学领域。因此，SageMath 吸引了众多用户，也契合了我们的理念：为所有从事 Drinfeld 模研究的数学家提供工具。其次，SageMath 尤其适合我们的项目，因为它已经实现了两个对我们至关重要的基础功能，即:

1. Ore 多项式环；当基环是有限域时，还提供许多额外功能。2. 用于处理环扩张的框架（这对我们很有用，因为我们需要将 K视为一个 $\mathbb{F}_q[T]$-代数）。

做出这一选择后，我们优先考虑与 SageMath 生态系统进行谨慎集成。这要求我们极其严谨，而我们也从 SageMath 核心开发者的反馈中获益良多；尤其是在项目开发过程中，

[a]康考迪亚大学[b]法国国家科学研究中心、波尔多大学、法国国家信息与自动化研究院
[c]洛林大学、法国国家信息与自动化研究院、法国国家科学研究中心[d]滑铁卢大学

我们始终非常注重接口的简洁性、文档的清晰性与完整性，以及单元测试。具体而言，每个类、方法或函数都配有文档测试，其中包含描述、测试和示例。文档的入口是 `DrinfeldModule` 的文档字符串，可在 Sage Math 控制台中运行 `DrinfeldModule?` 来访问。对于特定方法，也会使用 `?` 关键字，例如 `phi.rank?`。文档也收录于 SageMath 参考手册[Dev]。

我们的库完全开放，因此我们鼓励所有数学家和计算机科学家通过任何他们感兴趣的贡献来改进它。

1.2 Drinfeld 模对象的基类型。 项目伊始，我们遇到的第一个难题是，Drinfeld 模并非经典意义上的实际模。具体来说，Drinfeld 模没有底层集合，Drinfeld 模之间的态射也不是集合论意义上的映射。然而，在 SageMath 惯用法中，大多数对象要么是带有附加结构的集合——所谓的 `Parent` ——，要么是此类集合中的元素——即 `Element`。这种理念称为父对象/元素框架。在 SageMath 中，人们通常会默认这一点。例如，父对象的默认测试套件 会检查其范畴是否为**集合范畴**的子范畴； `Morphism`对象的构造函数假定定义域和陪域都是父对象，等等。 对于 Drinfeld 模，这引发了许多问题，最终我们不得不在以下三种折衷方案中做出艰难选择：

1. 将 Drinfeld 模作为元素（因为它们是本质上的态射）来实现，并将其集合设为父对象（SageMath 中所谓的“同态集”）；此选项采用标准的父对象/元素框架。2. 将 Drinfeld 模实现为不含元素的父对象，实际上遵循椭圆曲线的实现方式[1]。此选项让 Drinfeld 模之间的态射（以及更一般地，Drinfeld 模范畴）的实现更为容易。此外，从某种意义上说，将 Drinfeld 模视为椭圆曲线在函数域上的类比对象，此选项有坚实的数学基础。3. 将 Drinfeld 模实现为 `CategoryObject`。SageMath 中确实存在这个类，而且它本就不应包含元素。然而，遗憾的是，它仅偶尔使用，目前与 `Morphism` 对象不兼容，也不再维护（它最终可能会被弃用）。

所有这些选项各有利弊。我们与 SageMath 核心开发者讨论了所有选项（参见GitHub PR #37313 和 GitHub PR#34534）。在某个阶段，我们认为第三种选项在数学上最具吸引力；然而，鉴于 `CategoryObject`尚未得到充分支持，我们决定排除这种可能性。另一方面，第一种选项似乎更实用，但我们认为它在数学上具有误导性；而且，要让态射正常工作，还需要采用变通方案。最终，我们选择了第二种选项。

[1]在 SageMath 中，椭圆曲线 `E` 是概形，而 `E.an_element()` 返回的元素，其父对象不是 `E`，而是 `E`的点群 `G`。在这种情况下，`G`和`E` 是不同的对象。

2 我们软件包概览

我们的软件包可在 GitHub 上公开获取：
https://github.com/xcaruso/sage/tree/drinfeld-modules。
我们的最终目标是将其纳入 SageMath 的标准发行版。实际上，该软件包约有一半内容将随 SageMath 10.0 发布，另一半仍在审核中；我们希望 SageMath 社区很快会批准它。

另外，我们也提供了在 plm-binder 平台上在线试用软件包的方式。请访问以下网址：
https://caruso.perso.math.cnrs.fr/notebook/drinfeld-modules
几秒钟后，系统会打开一个 Jupyter 笔记本，其中包含完整教程，介绍软件包的主要功能。除了阅读教程，plm-binder 还支持编辑笔记本、执行命令、创建新工作表等。不过请注意，关闭会话后所做的修改不会保存；如果想保留修改，请记得下载笔记本！

2.1 构造与基本性质。 Drinfeld 模是一种相当复杂的数学对象，其定义本身就涉及若干非平凡要素：一个态射γ：$\mathbb{F}_q[T] \to K$，以及 Ore 多项式环$K\{\tau\}$。在我们的软件包中，我们尽可能减少创建 Drinfeld 模所需的代码行数。尤其是在大多数情况下，无需显式定义γ和$K\{\tau\}$。

```
sage: K.<w> = GF(4)
sage: phi = DrinfeldModule(GF(2)['T'], [w, 0, 0, 1])
sage: phi
Drinfeld module defined by T |--> t^3 + w
```

一旦实例化 Drinfeld 模，我们就可以使用一系列方法来访问其最重要的不变量，例如`phi.characteristic()`, `phi.rank()`, `phi.height()`等。只需使用语法 `phi(a)`，还可以计算 $\phi(a)$的值。

2.2 态射与同源。鉴于 Drinfeld 模没有元素，模之间的态射是我们理解其结构的主要工具。我们的软件包提供了 `hom` 方法，可用于轻松构造态射。

```
sage: t = phi.ore_variable()
sage: phi.hom(t + w)
Drinfeld Module morphism:
  From: Drinfeld module defined by T |--> t^3 + w
  To:   Drinfeld module defined by T |--> t^3 + t^2 +
      w*t + w
  Defn: t + w
```

我们可以看到，软件已自动确定陪域。一旦构造了一个态射f，就可以使用许多方法，例如 `f.codomain()`、`f.is_isomorphism()`，等等。对于 Drinfeld 模本身，方法 `is_isomorphic`可用于检查两个 Drinfeld 模是否同构。当K是有限的时，一个非常重要的态射是由 Ore 多项式定义的 Frobenius 自同态 $\tau^{[K:\mathbb{F}_q]}$（另见 §2.4）。我们的软件包

提供方法 `phi.frobenius_endomorphism()`，可快速实例化该自同态。

当然，态射的加法和复合，以及同构的逆，都已实现。我们还注意到，任意多项式 $P \in \mathbb{F}_q[T]$ 都定义了 ϕ 的一个自同态（对应于 Ore 多项式 ϕ_P）。特别地，Hom 空间 $\mathrm{Hom}(\phi, \psi)$ 继承了一个左模结构，其系数环为 $\mathbb{F}_q[T]$，；在我们的程序包中，可以通过运算符 `*` 使用这一结构。这种简洁的语法便于写出复杂的公式。

最后，与椭圆曲线的情形不同，计算定义在有限域上的 Drinfeld 模之间的态射，相当于求解一个关于 $\mathbb{F}_q$ 的线性方程组。这使得我们能够高效地设计算法来寻找同源 [Wes22],，并已在我们的程序包中实现。

```
sage: psi = DrinfeldModule(GF(2)['T'], [w, w+1, 1, 1])
sage: Hom(phi, psi).an_isogeny()
Drinfeld Module morphism:
  From: Drinfeld module defined by T |--> t^3 + w
  To:   Drinfeld module defined by T |--> t^3 + t^2 +
      (w + 1)*t + w
  Defn: t^2 + (w + 1)*t + 1
```

命令 `Hom(phi, psi).basis(degree=d)` 通常会返回由次数至多为 d 的 Ore 多项式定义的、 ϕ 与 ψ 之间态射向量空间的一组 $\mathbb{F}_q$-基。

2.3 j不变量。在经典理论中，众所周知，代数闭域上的椭圆曲线在同构意义下由其 j不变量分类 [Sil09, Proposition 1.4]。此外，在二次虚域 R 上讨论时，具有复乘且复乘来自 R 的椭圆曲线的 j不变量，为 R 的阿贝尔扩张提供了显式描述 [Sil94, 第 II 章]。Drinfeld 模也有类似结果：可以为任意秩2的 Drinfeld 模 ϕ 赋予一个 j不变量，该不变量决定了其在代数闭包上的同构类；此外，某些 j不变量在研究 $\mathbb{F}_q(T)$ 的某些代数扩张时起着关键作用 [Gek83, (4.4)], [Ham03, Theorem 6.9]。

Drinfeld 秩2模的 j不变量可由一个简单的闭式公式给出：若 $\phi_T = \gamma(T) + g_1(\phi)\tau + g_2(\phi)\tau^2$，则 $j(\phi) := g_1(\phi)^{q+1}/g_2(\phi)$。这样便于计算，我们的程序包也提供了直接访问该不变量的方法。

```
sage: phi = DrinfeldModule(GF(2)['T'], [w, w+1, w+2])
sage: phi.j_invariant()
w + 1
```

在 Drinfeld 模的背景下，j不变量可以定义在任意秩上 [Pot98]。秩为 $r > 2$ 的 Drinfeld 模并非只有一个 j不变量，而是有一整族 j不变量，其指标取自 $\mathbb{R}^r$ 的某个凸子集中的整点。所幸，这些 j不变量仍有明确的闭式公式，因此可以计算。我们的软件包提供了用于计算和处理任意秩下这些 j不变量的方法（`basic_j_invariant_parameters`、`basic_j_invariants`、 `jk_invariants`等）。详情请参阅我们关于 plm-binder 的教程。

2.4 范数与特征多项式。在经典情形中，椭圆曲线之间的态射（分别为自同态）具有范数（分别为特征多项式），这些量可以通过考察其在 Tate 模上的作用来求得[Lor96，§5]。同样，Drinfeld 情形下也有类似结论 [Gek91，引理 3。10]：Drinfeld 模的 Tate 模有良好定义，而 Drinfeld 模之间的态射确实会在相应的 Tate 模上诱导线性变换。基于这一构造，可以定义一般同源的范数以及自同态的特征多项式。遗憾的是，实际计算 Tate 模通常很困难，因为它往往存在于一个相当大的 K扩张中。不过，范数和特征多项式还有其他诠释，使得高效计算它们成为可能。具体而言，基于 Anderson motive 概念的相关算法[And86]已在 [CL23]中提出。我们已在软件包中实现这些算法，可通过以下方法调用 `norm` 和 `charpoly`。

当 K有限时，Drinfeld 模 ϕ的一个特殊自同态是其 Frobenius 自同态。它的特征多项式在理论中发挥着重要作用；值得注意的是，它完全决定了 ϕ [Gek91，定理 3。5]在我们的软件包中，我们实现了三种不同的算法来计算这一不变量，分别是：

- 基于上文讨论的 Anderson 动机的 `motive`算法，
- 基于 `crystalline`算法 [MS23], Frobenius 在晶体上同调上的作用，
- 基于 `CSA` 算法 [CL23], 将 Frobenius 的特征多项式重新解释为某个中心单代数中的约化范数。

图1 （见第 4 页）比较了我们的三种算法在不同 Drinfeld 模秩及扩张 [2] 次数下的运行时间 $K/\mathbb{F}_q$（在我们的示例中，$q = 5$）。我们观察到， `CSA` 算法在秩较大时表现更好，而 `crystalline`算法在[K： $\mathbb{F}_q$] 较大时表现最佳。我们软件包中用于此任务的入口方法 `frobenius_charpoly`经过调整，能够根据输入自行选择最佳可用算法；不过，用户仍可通过传入关键字 `algorithm`来指定使用某个特定算法。

作为这项计算的副产品，我们实现了一种方法 `is_isogenous`，用于检查给定的两个 Drinfeld 模是否同源。

2.5 指数函数与对数。 理解 Drinfeld 模的一个重要视角是解析观点。为说明这一点，我们回到椭圆曲线的情形：我们知道，椭圆曲线E在 $\mathbb{C}$ 上由一个自由 $\mathbb{Z}$-子模在 $\mathbb{C}$中一致化，其秩为 2，即 $E(\mathbb{C}) \cong \mathbb{C}/\Lambda$ 作为复李群 [Sil09，VI §5]。在 Drinfeld 情形中，将

[2]这里仍有一些优化空间。事实上，这三种算法最终都依赖于计算一个实际矩阵的特征多项式，该矩阵的系数属于 $K[T]$。为完成这项任务，我们只调用了 `charpoly` SageMath 中的函数；遗憾的是，该函数实现了一种四次复杂度的缓慢通用算法。尽管如此，我们认为图 1 仍有参考意义，因为其中的计时结果具有可比性。

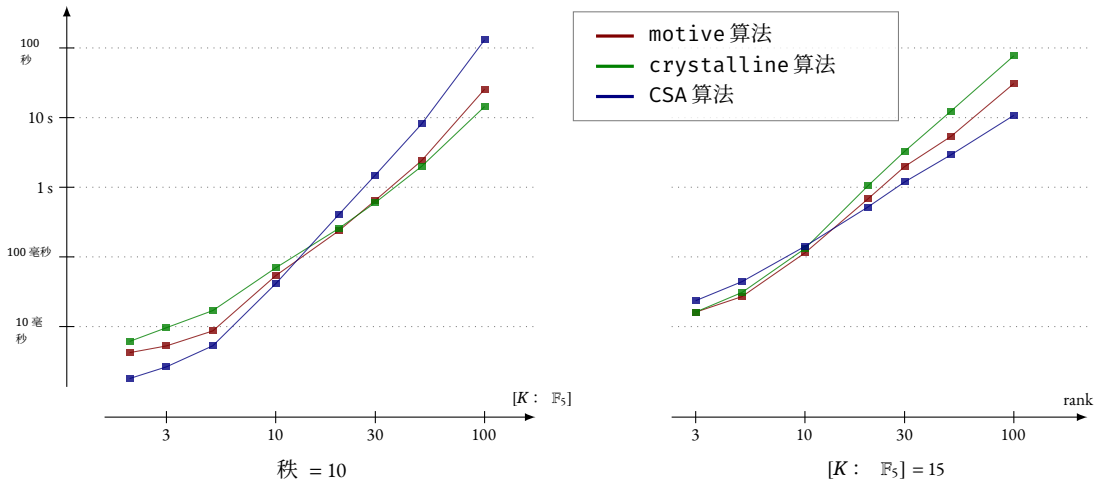

图1：计算 Frobenius 自同态特征多项式的耗时（CPU：Intel Core i5-8250U，1.60GHz；操作系统：Ubuntu 22.04.1）

域 $\mathbb{C}$ 替换为 $\mathbb{C}_\infty$，即与 $\frac{1}{T}$相关赋值下的代数闭包的完备化 $\mathbb{F}_q((\frac{1}{T}))$[Gos96，定理 4.6。9]。在这种情况下，一致化可通过一个 借助 一个 $\mathbb{F}_q$-线性、满射且非恒定的函数 $e_\phi\colon \mathbb{C}_\infty \to \mathbb{C}_\infty$ 来实现，该函数称为*Drinfeld* 模 ϕ 的指数函数。指数函数可表示为一个幂级数

$$e_\phi(z) = z + \sum_{i\geqslant 1} \alpha_i z^{q^i}$$

对于 $\alpha_i \in \mathbb{C}_\infty$ 和 $z \in \mathbb{C}_\infty$。该对数 ϕ，记作 $\log_\phi$ 是指数函数的复合逆。有关更多细节，请参阅 [Gos96] 第4章。在我们的实现中，任何 Drinfeld 模都具有以下方法 `exponential`和 `logarithm`，它们分别计算e_ϕ 和 $\log_\phi$ 的幂级数近似。代码采用惰性方式计算幂级数，即按需计算各项系数，用户无需输入任何精度参数。

致谢

我们感谢 Pierre-Jean Spaenlehauer 和 Emmanuel Thomé 的指导。David Ayotte 获得了 FRQNT 博士奖学金的资助。本研究还受益于 ANR 项目 CLap-CLap（ANR-18-CE40-0026-01）和 PadLEfAn（ANR-22-CE40-0013）的资助。

参考文献

[And86] Greg W. Anderson。《动机。t》，《杜克数学杂志》，53(2):457 – 502，1986。[Car35] Leonard Carlitz。关于与伽罗瓦域中多项式相关的某些函数。《杜克数学杂志》，1(2):137–168，1935。[Car18] Perlas Caranay。秩为二的 *Drinfeld* 模的同源火山计算。博士论文，卡尔加里大学，2018。[CGS20] Perlas Caranay、Mathew Greenberg 和 Renate Scheidler。有限域上秩为二的 Drinfeld 模的模多项式与同源计算。当代数学，754:293–314，2020。

[CL23] Xavier Caruso 和 Antoine Leudière。一般 Drinfeld 模上的范数与特征多项式的计算。准备中的手稿，2023年。[CLRS09] Keith Conrad、Franz Lemmermeyer、Peter J. Roquette 和 Jean-Pierre Serre。类域论史。2009年。[Dev] SageMath 开发者。SageMath 参考手册，第 10.0 版。将随 SageMath 10.0 版发布。[Dri74] Vladimir G. Drinfeld。椭圆模。《苏联数学文集》，23(4):561–592，1974年。[Gek83] Ernst-Ulrich Gekeler。关于 Drinfeld 模的算术。《数学年刊》，262(2):167–182，1983年。[Gek91] Ernst-Ulrich Gekeler。有限 Drinfeld 模。《代数学杂志》，1(141):187–203，1991年。[Gos96] David Goss。函数域算术的基本结构，数学及其相关领域成果（第 3 辑）第 35 卷 /数学及相关领域成果（第 3 辑）/。Springer-Verlag，柏林，1996年。[Ham03] Yoshinori Hamahata。Drinfeld 模的J不变量值。《数学手稿》，112(1):93–108，2003年。[Laf02] Laurent Lafforgue。Drinfeld 的 Chtoucas、Arthur–Selberg 迹公式与 Langlands 对应。载于《国际数学家大会论文集》，第 I 卷（北京，2002 年），第 383–400 页。高等教育出版社，北京，2002年。[Lor96] Dino Lorenzini。《算术几何导论》。美国数学学会，1996年。[LS22] Antoine Leudière 和 Pierre-Jean Spaenlehauer。由虚双曲函数域的类域论计算群作用。2022年。arXiv:2203.06970。[MS19] Yossef Musleh 和 Éric Schost。计算有限秩二 Drinfeld 模的特征多项式。载于《2019 ACM 国际符号与代数计算研讨会论文集》，第 307–314 页。ACM，2019年。[MS23] Yossef Musleh 和 Éric Schost。使用晶体上同调计算有限 Drinfeld 模的自同态的特征多项式。2023年。arXiv:2302.08611。[Pot98] Igor Yu. Potemine。Drinfeld 粗模空间的最小终端 Q-阶乘模型。《数学、物理与分析几何》，1(2):171–191，1998年。[Sil94] Joseph H. Silverman。椭圆曲线算术中的高级主题，数学研究生教材第 151 卷。Springer-Verlag，纽约，1994年。

[Sil09] Joseph H. Silverman。椭圆曲线的算术，数学研究生教材第 106 卷。Springer，Dordrecht，第二版，2009年。[Wes22] Benjamin Wesolowski。计算有限 Drinfeld 模之间的同源。Cryptology ePrint Archive，论文 2022 438，2022年 / 。

arXiv:2305.00422v1 [cs.SC] 30 Apr 2023

教程：SageMath 中的 Drinfeld 模

David Ayotte、Xavier Caruso、Antoine Leudière、Joseph Musleh在

线试用：https://caruso.perso.math.cnrs.fr/notebook/drinfeld-modules

目录

1 构造与基本性质 12 **态射与同源 5**3 j**-不变量 11**4 **范数和特征多项式 15**5 **指数函数和对数 21**

---

1 构造与基本性质

1.1 定义

设 $\mathbb{F}_q$ 表示一个含有 q 个元素的有限域。我们考虑一个 $\mathbb{F}_q$-代数 A 和一个 A-代数 K。我们用 $\gamma : A \to K$ 表示 K的定义态射。

另设 $K\{\tau\}$ 表示 K 上以 τ为变量的斜多项式环，也就是乘法按以下规则扭曲的通常多项式环

$$\tau a = a^q \tau, \quad \forall a \in K.$$

根据定义， A-Drinfeld 模是满足以下条件的环同态 $\phi : A \to K\{\tau\}$

- 对所有 $a \in A$， $\phi(a)$的常数项都是 $\gamma(a)$,
- 对所有 $a \in A, a \notin \mathbb{F}_q$， 斜多项式 $\phi(a)$的次数均为正。

我们通常用 ϕ_a 代替 $\phi(a)$。

下文中，我们只考虑 $A = \mathbb{F}_q[T]$ 的情形（目前我们的实现仅限于此），并将简称为 *Drinfeld* 模，而不称作 $\mathbb{F}_q[T]$-Drinfeld 模。值得指出的是，一个 Drinfeld 模 ϕ 完全由 ϕ_T 的数据确定。

1.2 在 SageMath 中的构造

我们首先定义 $\mathbb{F}_q$, A和 K。

[1]:
```
有限域 = GF(5)A。<T> =有
限域[]K。<z> =有限域。扩张(3)
```

现在，我们可以使用构造函数 DrinfeldModule 创建一个 Drinfeld 模：

[2]:
```
phi = DrinfeldModule(A, [z, 0, 1, 1])
phi
```

[2]: 由 T 定义的 Drinfeld 模 |--> t^3 + t^2 + z

构造函数的第一个参数是基环 A。第二个参数可以是以下任一项：

- 斜多项式 ϕ_T，或
- ϕ_T 的系数列表（如上例所示）。

在这两种情况下，都可以使用 ore_polring 方法恢复底层的斜多项式环：

[3]:
```
φ。ore_polring()
```

[3]: 关于 z 的大小为 5^3 的有限域上的 t 中的 Ore 多项式环，其基域通过 Frob 进行扭曲

多项式 $a \in A$ 的像只需调用 phi(a) 即可获得：

[4]:
```
phi(T)
```

[4]:
```
t^3 + t^2 + z
```

[5]:
```
phi(T^2 + 1)
```

[5]:
```
t^6 + 2*t^5 + t^4 + 2*z*t^3 + (3*z^2 + z  + 1)*t^2 + z^2 + 1
```

[6]:
```
phi(T^2 + 1)  == phi(T)^2 + 1 # 基本检查：phi 是一个环同态
```

[6]: 正确

1.3 基本性质

可以计算附属于 Drinfeld 模的许多标准不变量。

1.3.1 特征

Drinfeld 模的特征 ϕ 按定义是 γ 的核（的一个生成元）。

[7]:
```
phi。特征()
```

[7]:
```
T^3 + 3*T  + 3
```

1.3.2 秩

Drinfeld 模的秩，按定义即为斜多项式 ϕ_T 的次数。更一般地，它满足以下关系：

$$\forall a \in \mathbb{F}_q[T], \quad \deg \phi_a = \operatorname{rank}(\phi) \cdot \deg a.$$

[8]:
```
phi。秩()
```

[8]: 3

[9]:
```
a = T^2 + 1phi(a)。degree() == phi。
rank() * a。degree()
```

[9]: 真

1.3.3 高度

$\mathfrak{p}$ 是 ϕ的特征。根据定义， ϕ 的高度等于 $\phi_{\mathfrak{p}}$ 的 τ赋值除以 $\mathfrak{p}$的次数；它始终是一个正整数。

备注：高度是一个重要的不变量，因为它与 Drinfeld 模的 $\mathfrak{p}$-挠点的秩有关。

[10]:
```
phi。高度()
```

[10]: 1

高度为 1 是一般情形。不过，也存在高度更高的 Drinfeld 模。下面给出一个秩2的例子：

[11]:
```
ψ = DrinfeldModule(A, [z, 0, 1])
ψ。高度()
```

[11]: 2

1.4 Drinfeld 模的范畴

创建 Drinfeld 模时，SageMath 会同时自动创建它所属的范畴。该范畴会记录基环 A 和 K 以及定义态射 $\gamma : A \to K$。

[12]:
```
C = phi.phi 模块范畴
() C
```

[12]: 基域上的有限域 z 中的 Drinfeld 模范畴，大小为 5^3

在 SageMath 中将这个范畴作为一个实际对象十分重要，因为这有助于处理 Drinfeld 模之间的态射（稍后我们会看到）。此外，若干重要的不变量是在范畴层面定义的：

[13]:
```
C。base()                    # 域 K：作为 A 上的代数
```

[13]: 以其基域为基的、大小为 5^3 的 z 上的有限域

[14]:
```
C.基域_态射() # 定义态射 gamma # 也可以通过 C.base().defining_态射()
```

[14]: 环态射：从：底为有限域、大小为 5 的一元多项式环 T 到：以 z 为生成元、大小为 5^3 的有限域，其基域为底域 定义：T |--> z

[15]:
```
C.ore_多项式环() # 环 K{t}
```

[15]: 以 t 为变量、底域为以 z 为生成元且大小为 5^3 的有限域（其基域为底域）的 Ore 多项式环，由 Frob 扭曲

[16]:
```
C.特征()
```

[16]:
```
T^3 + 3*T  + 3
```

2 态射与同源

2.1 定义

设 ϕ, ψ： $\mathbb{F}_q[T] \to K\{\tau\}$ 是同一范畴中的两个 Drinfeld 模。

根据定义，态射 $\phi \to \psi$ 是满足 $u\phi_T = \psi_T u$ 的 $u \in K\{\tau\}$。（此关系更一般地意味着，对于所有 $a \in \mathbb{F}_q[T]$， $u\phi_a = \psi_a u$ 。）

根据定义，同源 是非零态射。我们注意到，如果存在同源 $\phi \to \psi$，那么 ϕ 和 ψ 必定具有相同的秩。

2.2 构造

[1]:
```
Fq = GF(5)
A.<T> =Fq[]K.<z> =Fq.扩张(3)phi  =
DrinfeldModule(A, [z, 0, 1, z])
```

2.2.1 构造函数同态集

创建态射最简单的方法是使用 hom 方法，我们可以直接向其传入定义斜多项式：

[2]:
```
t = phi。ore_变量() f  =
phi。hom(t + 1)f
```

[2]: Drinfeld 模态射：来自：由 T 定义的Drinfeld模 |--> z*t^3 + t^2 + z 至：由 T 定义的Drinfeld模 |--> (2*z^2 + 4*z + 4)*t^3 +(3*z^2 +␣↪2*z +2)*t^2 + (2*z^2 + 3*z + 4)*t + 定义：t + 1

从上面的例子中，我们可以看到，同源的陪域（不是 ϕ）是自动确定的：

[3]:
```
psi = f。psi 的陪域
()
```

[3]: 由.....定义的Drinfeld模，T |--> (2*z^2 + 4*z + 4)*t^3 + (3*z^2 + 2*z +␣↪2)*t^2

+ (2*z^2 + 3*z + 4)*t + z

给定Drinfeld模 ϕ 的一类重要自同态是标量乘法：它们是与斜多项式 ϕ_a 对应的自同态，其中 $a \in \mathbb{F}_q[T]$。这些自同态可按如下方式直接实例化：

[4]:
```
g = phi。
phi(T) 的同态集 g
```

[4]: 由 T 定义的Drinfeld模的自同态 |--> z*t^3 + t^2 + z 定义：z*t^3 + t^2 +z

2.2.2 两个 Drinfeld 模之间的态射集

When 我们知道定义域 ϕ 和陪域 ψ，就可以构造同态集 Hom(ϕ,ψ):

[5]:
```
同态集 = Hom(phi,
psi)H
```

[5]: 从(gen) z*t^到(gen) (2*z^3 + t^2 + z的Drinfeld 模态射集2 +4*z + 4)*t^3 + (3*z^2 + 2*z + 2)
*t^2 + (2*z^2 + 3*z + 4)*t + z

作为一种简便写法，当 $\phi = \psi$ 时，我们可以使用 End 原语:

[6]:
```
E = 自同态环
(phi) E
```

[6]: 从(gen) z*t^到(gen) z*t^3 + t^2 + 的Drinfeld 模态射集3 +↪t^2

+ z

请注意，这些同态集就是 f 和 g（我们之前创建的）的自然归属父对象:

[7]:
```
f。父对象()是H
```

[7]: True

[8]:
```
g.parent()是 E
```

[8]: True

定义同态集后，我们可以调用它来创建其中的态射（这与 SageMath 中的常见做法一致）。这为定义态射提供了 hom 构造函数之外的另一种方式。例如:

[9]:
```
Frob = E(t^3)
Frob
```

[9]: 由 T 定义的Drinfeld模自同态 |--> z*t^3 + t^2 + z 定义: t^3

后者是 Frobenius 自同态。也可以通过 frobenius_endomorphism 方法获取:

[10]:
```
phi。frobenius_自同态()
```

[10]: 由 T 定义的Drinfeld模自同态 |--> z*t^3 + t^2 + z 定义: t^3

2.3 关于同态集(ϕ, ψ) 和自同态环(ϕ)的结构

首先，同态集上当然有复合律：当 f 的定义域等于 g 的陪域时，可以复合 f 和 g，得到一个新的态射 $f \circ g$。在 Drinfeld 模的情形下，这一运算就对应于底层扭多项式的乘法。

在 SageMath 中，可以使用乘法运算符 * 或乘方算子 ^（或 **）来进行此运算：

[11]:
```
f* g
```

[11]: Drinfeld 模同态：来自：由.....定义的Drinfeld模 T |-> z*t^3 + t^2 + z 至：由.....定义的Drinfeld模T |-> (2*z^2 + 4*z + 4)*t^3 +(3*z^2 +␣↪2*z +2)*t^2 + (2*z^2 + 3*z + 4)*t + z 定义：(2*z^2 + 4*z + 4)*t^4 + (z + 1)*t^3 + t^2 + (2*z^2 + 4*z + 4)*t + z

[12]:
```
Frob^5
```

[12]: Drinfeld 模的自同态，定义为 T |--> z*t^3 + t^2 + z 定义：t^15

我们注意到，两个态射之和 $\phi \to \psi$ 是定义良好的。事实上，如果 u 和 v 满足关系 $\phi_T u = u\psi_T$和 $\phi_T v = v\psi_T$，那么可以立即推出 $\phi_T(u+v) = (u+v)\psi_T$。因此，Hom$(\phi, \psi)$ 在加法运算下自然构成一个交换群。

此外，鉴于 $\mathbb{F}_q[T]$ 中的元素会定义任意 Drinfeld 模的自同态：

- 同态集 Hom(ϕ, ψ) 继承了左 $\mathbb{F}_q[T]$-模结构（通过在左侧与 ψ_a of ψ 这一自同态复合），
- 自同态环 End(ϕ) 继承了 $\mathbb{F}_q[T]$-代数结构。

我们的实现能够以非常直观的方式处理这些结构。例如：

[13]:
```
T* f
```

[13]: Drinfeld 模同态：来源：由 T |-> z*t^3 + t^2 + z 定义的 Drinfeld 模 目标：由 T |-> (2*z^2 + 4*z + 4)*t^3 +(3*z^2 +␣↪2*z +2)*t^2 + (2*z^2 + 3*z + 4)*t + z 定义的 Drinfeld 模 定义式：(2*z^2 + 4*z + 4)*t^4 + (z + 1)*t^3 + t^2 + (2*z^2 + 4*z + 4)*t + z

[14]:
```
Frob + g
```

[14]: 由 T 定义的Drinfeld 模的自同态 |--> z*t^3 + t^2 + z 定义：(z + 1)*t^3 +t^2 + z

[15]:
```
Frob* g  == g * Frob
```

[15]: 真

[16]:
```
(Frob  + g)^5 == Frob^5 + g^5
```

[16]: 真

2.4 识别同构

根据定义很容易得出，态射 $f\colon \phi \to \psi$ 当且仅当定义 f 的斜多项式为常数时是同构。

方法 is_同构 可用于检查这一事实。

[17]:
```
g。是_同构()
```

[17]: 假

[18]:
```
h = phi。phi的同态集(z)
h
```

[18]: Drinfeld 模态射：出发自：由 T 定义的Drinfeld模 `|--> z*t^3 + t^2 + z` 到：由 T 定义的Drinfeld模 `|--> z*t^3 + (4*z^2 + z  + 4)*t^2 + z` 定义：`z`

[19]:
```
h.is_同构()
```

[19]: 是

inverse 方法用于计算同构的逆：

[20]:
```
h.逆()
```

[20]: Drinfeld 模态射：来源：由……定义的Drinfeld模 T `|--> z*t^3 + (4*z^2 + z  + 4)*t^2 + z` 目标：由……定义的Drinfeld模 T `|--> z*t^3 + t^2 + z` 定义：`3*z^2 + 4`

也可以使用 is_isomorphic 方法检查给定的两个 Drinfeld 模是否同构：

[21]:
```
phi2 = h。codomain()phi。是否_isomorphic(phi2)
```

[21]: 是

[22]:
```
phi。是_同构(psi)
```

[22]: 否

在最后一种情况下，ϕ 和 ψ 是同源的（通过 f），但它们不同构。

2.5 有限域的情况

当 K 是一个有限域时，Hom(ϕ, ψ) 中由次数至多为 d 的斜多项式所定义的态射构成的子空间（其中 d 为给定的正整数）是一个有限维的 $\mathbb{F}_q$-向量空间。

同态集上的 basis 方法会返回该向量空间的一组基：

[23]:
```
H。基底(次数=5)
```

[23]: [Drinfeld 模同态：来自：由……定义的Drinfeld模 T ⊢→ z*t^3 + t^2 + z 目标：由……定义的Drinfeld模 T ⊢→ (2*z^2 + 4*z + 4)*t^3 + (3*z^2 + 2*z+ 2)*t^2 + (2*z^2 + 3*z + 4)*t + z 定义：t + 1，Drinfeld 模同态：来自：由……定义的Drinfeld模 T ⊢→ z*t^3 + t^2 + z 目标：由……定义的Drinfeld模 T ⊢→ (2*z^2 + 4*z + 4)*t^3 + (3*z^2 + 2*z+ 2)*t^2 + (2*z^2 + 3*z + 4)*t + z 定义：(2*z^2 + 4*z + 3)*t^4 + z*t^3 + t^2 + (2*z^2 + 4*z + 4)*t + z，Drinfeld 模同态：来自：由……定义的Drinfeld模 T ⊢→ z*t^3 + t^2 + z 目标：由……定义的Drinfeld模 T ⊢→ (2*z^2 + 4*z + 4)*t^3 + (3*z^2 + 2*z+ 2)*t^2 + (2*z^2 + 3*z + 4)*t + z 定义：(3*z^2 + 3*z + 1)*t^4 + (3*z^2 + 4*z)*t^3 + (3*z^2 + z + 1)*t^2 + (2*z+ 3)*t + z^2，Drinfeld 模同态：来自：由……定义的Drinfeld模 T ⊢→ z*t^3 + t^2 + z 目标：由……定义的Drinfeld模 T ⊢→ (2*z^2 + 4*z + 4)*t^3 + (3*z^2 + 2*z+ 2)*t^2 + (2*z^2 + 3*z + 4)*t + z 定义：t^4 + t^3]

在有限 域中，我们也有一种方法可以检查 Hom 空间是否缩减为零 o.

[24]:
```
H。是_zero()
```

[24]: 假

[25]:
```
psi2 = DrinfeldModule(A, [z, 0, 1, z^2])
同态集2 = Hom(phi, psi2) H2。是_zero()</style}
```

[25]: 真

此外，当同态集非零时，我们可以使用方法 an_同源来生成其中一个非零态射：

[26]:
```
H。an_同源()
```

[26]: Drinfeld 模同态：源：由……定义的Drinfeld模 T ⊢--> z*t^3 + t^2 + z 目标：由……定义的Drinfeld模 T ⊢--> (2*z^2 + 4*z + 4)*t^3 +(3*z^2 + 2*z +2)*t^2 + (2*z^2 + 3*z + 4)*t + z

定义：t + 1

3 j-不变量

3.1 定义

Drinfeld 模的 j-不变量构成一族元素，用于刻画该 Drinfeld 模在代数闭包上的同构类。

更准确地说，设 $\phi\colon \mathbb{F}_q[T] \to K\{\tau\}$ 是一个 Drinfeld 模，并记

$$\phi_T = g_0 + g_1\tau + \cdots + g_r\tau^r$$

其中 r 表示 ϕ 的秩。设 $(k_1, \ldots, k_n)$、$(d_1, \ldots, d_n)$ 是两个 n 元整数元组，满足 $1 \le k_1 < k_2 < \ldots < k_n < r$，且对所有 i 都有 $d_i > 0$。我们再取一个非负整数 d，并假设满足所谓的权重-0 条件：

$$d_1(q^{k_1} - 1) + d_2(q^{k_2} - 1) + \cdots + d_n(q^{k_n} - 1) = d(q^r - 1).$$

对于这些数据，我们关联如下量，称为 ϕ 的 $((k_1, \ldots, k_n),(d_1, \ldots, d_n, d))$-$j$- 不变量：

$$j^{d_1,\ldots,d_n,d}_{k_1,\ldots,k_n}(\phi) := \frac{1}{g_r^d}\prod_{i=1}^{n} g_{k_i}^{d_i}$$

若 $\gcd(d_1, \ldots, d_n, d) = 1$，则称 j-不变量为基本 不变量。基本j-不变量只有有限多个。此外，若 $((k_1, \ldots, k_n), (d_1, \ldots, d_n, d))$ 是任意参数，只需将 d_i 和 d 除以它们的 gcd，即可构造出一个基本不变量。与初始参数对应的 j-不变量，等于基本 j-不变量的 $\gcd(d_1, \ldots, d_n, d)$ 次幂。

Potemine 的一项经典结果指出，两个 Drinfeld 模 ϕ 和 ψ 在 K 上同构¯，当且仅当它们的基本 j-不变量都相等。

3.2 j-不变量的计算

[1]:
```
Fq = GF(5)A。<T> =Fq[]K。<z> =Fq。扩张
(3)phi  = DrinfeldModule(A, [z, 0, 1, z, z^2])
```

首先需要了解的是，能够得到基本 j-不变量的（有限）参数列表 $((k_1, \ldots, k_n),(d_1, \ldots, d_n, d))$。可通过方法 basic_j_invariant_parameters 访问此列表。

[2]:
```
参数 = phi。basic_j_不变量_parameters() 参数
```

[2]:
```
[((1, 2, 3), (26, 1, 4, 1)),
 ((1, 2, 3), (57, 1, 3, 1)),
 ((1, 2, 3), (88, 1, 2, 1)),
 ...
 ((1, 2, 3), (93, 26, 153, 32)),
 ((1, 2, 3), (155, 26, 151, 32)),
 ((1, 2, 3), (156, 26, 156, 33))]
```

对于秩为 5 的 Drinfeld 模，这个列表相当长：

[3]:
```
长度(参数)
```

[3]:
```
3402
```

然而，当 ϕ_T 的某些系数为零时，一些 j-不变量也会随之为零，因为后者是由前者相乘得到的。在这种情况下，了解会产生非零基本 j-不变量的参数列表很有用。为此，只需传入关键字 nonzero=True 即可。

[4]:
```
非零_参数 = phi。基本_j_不变量_参数(nonzero=True) 非零_参数
```

[4]:
```
[((2, 3), (1, 30, 6)),
 ((2, 3), (6, 24, 5)),
 ((2, 3), (7, 54, 11)),
 ((2, 3), (8, 84, 17)),
 ((2, 3), (9, 114, 23)),
 ((2, 3), (10, 144, 29)),
 ((2, 3), (11, 18, 4)),
 ((2, 3), (13, 78, 16)),
 ((2, 3), (15, 138, 28)),
 ((2, 3), (16, 12, 3)),
 ((2, 3), (17, 42, 9)),
 ((2, 3), (19, 102, 21)),
 ((2, 3), (20, 132, 27)),
 ((2, 3), (21, 6, 2)),
 ((2, 3), (23, 66, 14)),
 ((2, 3), (25, 126, 26))]
```

一旦我们知道一个有效参数，就可以使用 j_不变量方法计算 j-不变量：

[5]:
```
参数 = 非零_参数[0]phi。j_不变量
(parameter)
```

[5]:
```
4*z^2 + 2*z  + 1
```

也可以调用 basic_j_不变量方法，获取完整的 j-不变量列表：

[6]:
```
phi。basic_j_不变量(nonzero=True)
```

[6]:
```
{((2, 3), (1, 30, 6)): 4*z^2 + 2*z + 1,
 ((2, 3), (6, 24, 5)): 4*z + 3,
 ((2, 3), (7, 54, 11)): 2*z,
 ((2, 3), (8, 84, 17)): 4*z^2 + 3*z + 1,
 ((2, 3), (9, 114, 23)): z^2 + 2*z + 1,
 ((2, 3), (10, 144, 29)): 2*z^2 + 2*z + 1,
 ((2, 3), (11, 18, 4)): 2*z + 3,
 ((2, 3), (13, 78, 16)): 3*z^2 + z,
```

((2, 3), (15, 138, 28)): z^2 + 3*z + 1, ((2, 3), (16, 12, 3)): 4*z^2 + 3*z + 4, ((2, 3), (17, 42, 9)): 3*z^2 + 3*z + 4, ((2, 3), (19, 102, 21)): 3*z^2 + 2*z + 4, ((2, 3), (20, 132, 27)): 2*z^2 + 2*z + 2, ((2, 3), (21, 6, 2)): z^2, ((2, 3), (23, 66, 14)):z^2 + 4*z + 1, ((2, 3), (25, 126, 26)): 2*z^2 + z + 1}

3.2.1 j_k-不变量

与基本参数相关联的 j-不变量

$$\left(k, \left(\frac{q^r-1}{q^{\gcd(r,k)}-1}, \frac{q^k-1}{q^{\gcd(r,k)}-1}\right)\right)$$

称为 j_k-不变量。在这种情况下，方法 j_invariant 也接受整数 k 作为参数。

[7]:
```
phi。j_不变量(3)
```

[7]:
```
3*z
```

[8]:
```
phi。j_不变量([[3], [156, 31]])
```

[8]:
```
3*z
```

3.3 j 不变量（低秩情形）

3.3.1 秩为 1 的情形

[9]:
```
phi = DrinfeldModule(A, [z, 1])
phi
```

[9]: 由 T 定义的Drinfeld 模 |--> t + z

秩为 1 时，不存在基本的 j 不变量。

[10]:
```
phi。基本_j_不变量_parameters()
```

[10]: []

因此，计算 j-不变量并不重要。

此外，根据 Potemime 定理，我们可以得出结论：所有秩为 1 且具有相同定义态射 γ：$\mathbb{F}_q[T] \to K$ 的 Drinfeld 模，在代数闭包 K of K上彼此同构。然而，它们并不一定在 K上同构。

[11]:
```
psi = DrinfeldModule(A, [z, z])
phi。是_同构(psi)
```

[11]: 假

[12]:
```
phi。是_同构(psi, 绝对=真)
```

[12]: 是

3.3.2 秩为 2 的情形

[13]:
```
φ = DrinfeldModule(A, [z, z^2, z^3]) φ
```

[13]: 由.....定义的Drinfeld模 T |--> (2*z + 2)*t^2 + z^2*t + z对于秩为 2 的 Drinfeld模，只有一个基本 j不变量，即 $(1,(q+1,1))$

[14]:
```
phi.基本_j_不变量_parameters()
```

[14]: [((1,),(6, 1))]

在这种情况下（且仅在这种情况下！），允许不带任何参数调用方法 j_不变量。

[15]:
```
phi。j_不变量()
```

[15]: 4*z^2 + 4

4 范数与特征多项式

4.1 定义

4.1.1 自同态的情形

设 ϕ: $\mathbb{F}_q[T] \to K\{\tau\}$ 是一个特征为 $\mathfrak{p}$ 的 Drinfeld 模，并设 $\mathfrak{q}$ 是不同于 $\mathfrak{p}$ 的 $\mathbb{F}_q[T]$ 的极大理想。

F对于每个正整数 n，考虑由下式定义的 ϕ 的 $\mathfrak{q}^n$-挠点集合 :

$$\phi[\mathfrak{q}^n] = \left\{x \in \bar{K} : \phi_a(x) = 0, \forall a \in \mathfrak{q}^n\right\}$$

这里约定，斜多项式 ϕ_a 中的变量 τ 作用为 Frobenius $x \mapsto x^q$。通过令 $a \in \mathbb{F}_q[T]$ 作用为 $x \mapsto \phi_a(x)$，空间 $\phi[\mathfrak{q}^n]$ 继承了 $\mathbb{F}_q[T]$-模结构。根据定义， $\phi[\mathfrak{q}^n]$ 被 $\mathfrak{q}^n$ 消去。

ϕ 的 Tate 模定义如下:

$$T_{\mathfrak{q}}(\phi) = \varprojlim_{n\to\infty} \phi[\mathfrak{q}^n]$$

它是环 $A_{\mathfrak{q}}$ 上的一个模，定义为 $\mathbb{F}_q[T]$ 在赋值处 $\mathfrak{q}$ 的完备化，即 $A_{\mathfrak{q}} = \varprojlim_{n\to\infty} \mathbb{F}_q[T]/\mathfrak{q}^n$。若 r 表示 ϕ 的秩，则一个标准结果表明， $T_{\mathfrak{q}}(\phi)$ 是 $A_{\mathfrak{q}}$ 上秩为 r 的自由模。

任意自同态 f: $\phi \to \phi$ 都诱导一个 $A_{\mathfrak{q}}$-线性映射 $T_{\mathfrak{q}}(f)$: $T_{\mathfrak{q}}(\phi) \to T_{\mathfrak{q}}(\phi)$。根据定义:

- f 的范数是 $T_{\mathfrak{q}}(f)$ 的行列式,
- f 的特征多项式是 $T_{\mathfrak{q}}(f)$ 的特征多项式。

可以证明，它们都不依赖于 $\mathfrak{q}$ 的选择，并且前者属于 $\mathbb{F}_q[T]$，而后者是一个系数属于 $\mathbb{F}_q[T]$ 的多项式。

4.1.2 一般态射的情形

现在考虑两个不同的 Drinfeld 模 ϕ 和 ψ之间的态射 $f: \phi \to \psi$ 。在这种情况下， f 的特征多项式不再有定义，但 f 的范数仍然有意义，它是 $\mathbb{F}_q[T]$的一个理想（而非实际元素）。

为定义它，我们考虑映射

$$\begin{array}{rccc} T(f): & \bar{K} & \to & \bar{K} \\ & x & \mapsto & u(x) \end{array}$$

其中， u是定义 f的斜多项式。注意，若分别赋予定义域和陪域由 ϕ 和 ψ 诱导的 $\mathbb{F}_q[T]$-模结构，则 $T(f)$是 $\mathbb{F}_q[T]$-线性的。若 f非零，则 $T(f)$的余核是一个有限生成的挠 $\mathbb{F}_q[T]$-模；因此，它具有如下形式

$$\operatorname{coker} T(f) \simeq \mathbb{F}_q[T]/\mathfrak{a}_1 \times \mathbb{F}_q[T]/\mathfrak{a}_2 \times \cdots \times \mathbb{F}_q[T]/\mathfrak{a}_m$$

对于 $\mathbb{F}_q[T]$ 的某些理想 $\mathfrak{a}_1$、.....、 $\mathfrak{a}_m$。 f 的范数定义为 $\nu(f) = \mathfrak{a}_1 \cdots \mathfrak{a}_m$（理想的乘积）。

4.2 在 SageMath 中计算范数和特征多项式 h

正如我们所见，范数和特征多项式的定义涉及一些复杂的中间对象（即 Tate 模 $T_q(f)$ 或映射 $T(f)$），而这些对象本身又涉及 K 的代数闭包，因此在保持良好性能的同时，实现起来似乎很困难。

不过，我们的实现提供了高效计算任意态射/自同态的范数和特征多项式的功能。这些功能基于 Anderson 动机理论中关于范数和特征多项式的替代定义。

4.2.1 自同态

[1]:
```
Fq = GF(5)A。<T> =Fq[]K。<z> =Fq。
extension(3)phi  = DrinfeldModule(A, [z,
0, 1, z])
```

我们首先计算 Frobenius 自同态的范数：

[2]:
```
Frob = phi。frobenius_自同态() Frob。范
数()
```

[2]: 主理想 (T^3 + 3*T + 3)，T 上大小为 5 的有限域上的一元多项式环

为保持一致，范数方法默认总是返回一个理想。传入参数 ideal=False 可以覆盖此行为：

[3]:
```
Frob。范数(ideal=False)
```

[3]:
```
3*T^3 + 4*T + 4
```

我们观察到，Frobenius 的范数与特征成 $\mathbb{F}_q$-共线，这是一个标准结论：

[4]:
```
phi。特征()
```

[4]:
```
T^3 + 3*T + 3
```

[5]:
```
3 *phi.特征()
```

[5]: 3*T^3 + 4*T + 4我们可以类似地计算自同态 ϕ_a 对于某个 $a \in \mathbb{F}_q[T]$

[6]:
```
f =
phi..hom(T) f.范数()
```

[6]: 有限域（大小为 5）上的 T 一元多项式环中的主理想 (T^3)

[7]:
```
g = phi。同态集
(T+1)g。范数()
```

[7]:
```
一元多项式环中 T^3 + 3*T^2 + 3*T + 1的主理想
↪上的
```

大小为 5 的有限域

在每种情况下，我们观察到，ϕ_a 的范数是由 a^3生成的理想；实际上，ϕ_a 的范数为 $a^r\mathbb{F}_q[T]$ 是一个普遍事实，其中 r 是底层 Drinfeld 模的秩。
类似地，可以借助 charpoly 方法（或 characteristic_polynomial）计算特征多项式：

[8]:
```
Frob。特征多项式()
```

[8]:
```
X^3 + (T + 1)*X^2 + (2*T + 3)*X + 2*T^3 + T + 1
```

[9]:
```
f。charpoly()
```

[9]:
```
X^3 + 2*T*X^2 + 3*T^2*X + 4*T^3
```

[10]:
```
g。charpoly(var='Y') # 我们可以更改变量名
```

[10]:
```
Y^3 + (2*T + 2)*Y^2 +(3*T^2 + T + 3)*Y + 4*T^3 + 2*T^2 + 2*T
+ 4
```

再次，我们可以验证 ϕ_a 的特征多项式为 $(X-a)^r$：

[11]:
```
g。特征多项式()。因式分解()
```

[11]:
```
(X + 4*T + 4)^3
```

4.2.2 一般态射

对于一般态射，只有范数方法可用（因为在这种一般情形下未定义特征多项式）。

[12]:
```
t = phi。ore_变量() h
= phi。同态集(t + 1)h
```

[12]:
```
Drinfeld 模态射：源：由 T 定义的Drinfeld模 |--> z*t^3 + t^2 + z 目标：由 T 定义
的Drinfeld模 |--> (2*z^2 + 4*z + 4)*t^3 +(3*z^2 +
↪2*z +2)*t^2 + (2*z^2 + 3*z
+ 4)*t + 定义：t + 1
```

[13]:
```
h。范数()
```

[13]:
```
主理想 (T + 4) 的一元多项式环（T 上的有限域）
↪的
```

大小为 5

我们通过示例验证范数具有乘法性：

[14]:
```
(h * Frob) 。范数() == h。范数()* Frob。范数()
```

[14]: 真

[15]:
```
(h * f) 。范数() == h。范数()* f。范数()
```

[15]: 真

[16]:
```
(Frob * g) 。范数() == Frob。范数()* g。范数()
```

[16]: True

4.3 Frobenius 自同态的情形

当 K 是有限域时，Frobenius 自同态的特征多项式是一个至关重要的不变量。我们的实现提供了一种直接访问它的方法：

[17]:
```
phi。frobenius_charpoly()
```

[17]:
```
X^3 + (T + 1)*X^2 + (2*T + 3)*X + 2*T^3 + T + 1
```

实际上，此计算有多种不同的算法可用。可以通过传入关键字 algorithm 来指定，选项包括：

- gekeler：它尝试通过令特征多项式消去 Frobenius 自同态来确定系数；此算法在某些情况下可能会失败；
- 动机：该方法利用 Frobenius 在 Anderson 动机上的作用；
- 晶体：该方法利用 Frobenius 在晶体上同调上的作用；
- 中心单代数：该方法利用 Frac $K\{\tau\}$的中心单代数结构。

当然，所有这些算法都会给出相同的答案，但性能可能有所不同。默认情况下，当秩小于扩张次数时，选择晶体算法$K/\mathbb{F}_q$；否则选择中心单代数算法。

4.3.1 同源性检验

一个重要结果表明，两个 Drinfeld 模 ϕ 和 ψ 同源，当且仅当 ϕ 和 ψ 的 Frobenius 自同态的特征多项式相同。这提供了一种高效的同源性检验，可通过我们软件包中的方法 实现 is_isogenous。

[18]:
```
psi  = h.codomain() # 请记住，h 是一个同源映射，其定义域为 phiphi.is_
isogenous(psi)
```

[18]: 真

[19]:
```
phi2 = DrinfeldModule(A,  [z、0、1、z^2])
phi。是_同源(phi2)
```

[19]: 假

最后需要指出的是，当确实存在同源时，可以使用同态集上的 an_isogeny 方法找到它。

[20]:
```
同态集(phi, psi)。一个_isogeny()
```

[20]: Drinfeld 模态射：源：由……定义的Drinfeld模 T |--> z*t^3 + t^2 + z 目标：由……定义的Drinfeld模 T |--> (2*z^2 + 4*z + 4)*t^3 +(3*z^2 +↵2*z +2)*t^2 + (2*z^2 + 3*z + 4)*t + z 定义：t + 1

4.3.2 计时下面我们比较运行时间。当 $[K:\mathbb{F}_q]$ 较大时，晶体算法最快：

[21]:
```
Fq = GF(5) A.<T> =Fq[]K.<z> =Fq.extension(50) phi = DrinfeldModule(A, [z] + [K.random_元素() for _ in range(5)] + [1])
```

[22]:
```
%时间 _ = phi.frobenius_charpoly(算法="动机")
```

CPU 时间：用户 969 毫秒，系统 5.7 毫秒，总计 975 毫秒；实际耗时 981 毫秒

[23]:
```
%时间 _ = phi.frobenius_charpoly(算法="晶体")
```

CPU 时间：用户 623 毫秒，系统 2.98 毫秒，总计 626 毫秒；实际耗时 629 毫秒

[24]:
```
%时间 _ = phi.frobenius_charpoly(算法="中心单代数")
```

CPU 时间：用户 3.92 秒，系统 14.6 毫秒，总计 3.93 秒；实际耗时 3.96 秒

相反，当秩较大时，中心单代数算法最快：

[25]:
```
Fq = GF(5) A.<T> =Fq[]K.<z> =Fq.extension(5) phi = DrinfeldModule(A, [z] + [K.random_元素() for _ in range(50)] + [1])
```

[26]:
```
%时间 _ = phi.frobenius_charpoly(算法="动机")
```

CPU 时间：用户 1.15 秒，系统：12.9 毫秒，总计：1.16 秒 墙上时间：1.17 秒

[27]:
```
%时间 _ = phi.frobenius_特征多项式(算法="晶体")
```

CPU 时间：用户 2.08 s，系统：13.9 ms，总计：2.1 s；实际耗时：2.11 s

[28]:
```
%时间 _ = phi.frobenius_charpoly(算法="中心单代数")
```

CPU 时间：用户 40 毫秒，系统 1.97 毫秒，总计 42 毫秒；墙钟时间：41.8 毫秒

5 指数函数与对数

5.1 定义

考虑有理函数域 $K = \mathbb{F}_q(T)$，并设 $K_\infty = \mathbb{F}_q((\frac{1}{T}))$ 为该域在赋值处 $\frac{1}{T}$ 处的完备化。我们将 $\mathbb{C}_\infty$ 定义为 K_∞ 的代数闭包的完备化。域 $\mathbb{C}_\infty$ 在作用上类似于复数域。

Drinfeld 模的一个重要方面是其解析观点。给定任意秩为 r 的 Drinfeld 模 ϕ: $\mathbb{F}_q[T] \to \mathbb{C}_\infty\{\tau\}$，可以证明存在一个秩为 r 的 $\mathbb{F}_q[T]$-格，即 一个秩为 r 的 $\mathbb{C}_\infty$ 的自由 $\mathbb{F}_q[T]$-子模，记作 Λ_ϕ，使得对于任意 $z \in \mathbb{C}_\infty$ 和 $a \in \mathbb{F}_q[T]$

$$\phi_a(z) = z \prod_{\substack{\lambda \in a^{-1}\Lambda_\phi / \Lambda_\phi \\ \lambda \neq 0}} \left(1 - \frac{z}{e_\phi(\lambda)}\right)$$

其中 e_ϕ : $\mathbb{C}_\infty \to \mathbb{C}_\infty$ 是一个 $\mathbb{F}_q$-线性、满射且非常值的函数。特别地，它可以写成幂级数

$$e_\phi(z) = z + \sum_{i \geq 1} \alpha_i z^{q^i}.$$

该函数称为 ϕ 的指数函数 。e_ϕ 的复合逆存在，称为 ϕ 的对数 ，记作 $\log_\phi$:

$$\log_\phi(z) = z + \sum_{i \geq 1} \beta_i z^{q^i}.$$

5.2 解释

这里我们说明代码的工作原理。简而言之，给定任意 Drinfeld 模

$$\phi : T \mapsto g_0 + g_1\tau + \cdots + g_r\tau^r,$$

我们首先通过递归过程计算对数，然后对所得幂级数求逆，从而得到指数函数。

更准确地说，我们使用以下函数方程:

$$e_\phi(az) = \phi_a(e_\phi(z)).$$

对函数方程两边取 $\log_\phi$ ，得到

$$a \log_\phi(z) = \log_\phi(\phi_a(z)).$$

然后，可以比较上述等式两边的系数，得到以下递归过程:

$$a\beta_i = \sum_{n+m=i} \beta_n g_m^{q^n}.$$

令 $\beta_0 := 1$，该过程会生成一个与对数函数对应的无穷序列 $(\beta_i)_{i\geq 0}$ 。

接下来，由于根据定义有 $e_\phi \circ \log_\phi(z) = z$，我们现在可以递归地计算 e_ϕ 的系数：

$$\alpha_i = -\sum_{n=0}^{i-1} \alpha_i \beta_{i-n}^{q^n}.$$

所得序列 $(\alpha_i)_{i\geq 0}$ 对应于 ϕ 的指数函数。

[1]:
```
q = 4Fq = GF(q)A  =Fq['T']K。<T> =Frac(A)phi
 = DrinfeldModule(A, [T, T+1, T^2 - T  + 1])
```

我们使用 exponential 方法计算 e_ϕ 的幂级数近似：

[2]:
```
exp = phi。指数函数()
```
exp

[2]:
```
z  + ((1/(T^3+T^2+T))*z^4) + O(z^8)
```

要计算对数，请使用 logarithm 方法：

[3]:
```
对数 = φ。指数函数() 对数
```

[3]:
```
z  + ((1/(T^3+T^2+T))*z^4) + O(z^8)
```

返回的幂级数是一个惰性幂级数。这意味着用户无需输入精度参数，代码会按需计算系数。

[4]:
```
exp[q**5]
```

[4]:
```
(T^2045 + T^2044 + ... + T^2 + 1)/(T^4827 + T^4825 + ... + T^342 + T^341)
```

[5]:
```
对数[q**5]
```

[5]:
```
(T^2045 + T^2044 + ...  + T^2 + 1)/(T^4827 + T^4825 + ...  + T^342 +T^
341)
```
对数函数和指数函数互为复合逆：

[6]:
```
对数.compose(exp)
```

[6]:
```
z  +O(z^8)
```

[7]:
```
exp。compose(log)
```

[7]:
```
z  +O(z^8)
```

为了获得更多系数，可以对级数进行切片：

[8]:
```
exp[0:q^2 + 1] # 从 0 到 q 的系数^2
```

[8]:
```
[0,
 1,
 0,
 0,
 1/(T^3 + T^2 + T),
 0,
 0,
 0,
 0,
 0,
 0,
 0,
 0,
 0,
 0,
 (T^14 + T^13 + T^12 + T^10 + T^9 + T^8 + T^6 + T^5 + T^4 + T + 1)/(T^28 +␣
↪T^24
+ T^20 + T^13 + T^9 +T^5)]
```

5.3 Carlitz 模的情形

当 $\phi = \rho$ 时，Carlitz 模的指数函数和对数的系数具有已知的闭式公式：

$$\rho : T \mapsto T + \tau.$$

更确切地说，对于任意 $i > 0$，我们定义以下量：

- $[i] := T^{q^i} - T$,
- $D_0 := 1$ 和 $D_i := [i][i-1]^q \cdots [1]^{q^{i-1}}$,
- $L_0 := 1$ 和 $L_i := [i][i-1] \cdots [1]$.

那么，指数函数可显式表示为：

$$e_\rho(z) = \sum_{i \geq 0} \frac{z^{q^i}}{D_i}$$

而对数表示为

$$\log_\rho(z) = \sum_{i \geq 0} (-1)^i \frac{z^{q^i}}{L_i}.$$

[9]:
```
rho  = DrinfeldModule(A, [T,  1]) # Carlitz 模exp =
rho.指数函数() 对数 = rho.对数()
```

[10]:
```
指数函数
```

[10]:
```
z  + ((1/(T^4+T))*z^4) + O(z^8)
```

[11]: 对数

[11]:
```
z  + ((1/(T^4+T))*z^4) + O(z^8)
```

[12]:
```
def sq(i): """ 返回 [i] = T^(q^i) - T。 """ return T^(q^i) - T def D(i): """ 如果 i = 0 and D_i = [i][i - 1]^q *... * [1]^(q^(i - 1)), 则返回 1；否则返回。 """ if i == 0: return 1 return prod(sq(n)**(q^(i - n)) for n in range(1, i + 1)) def L(i): """ 如果 i = 0and L_i = [i][i - 1] *...* [1] , 则返回 1；否则返回。 """ if i == 0: return 1 return prod(sq(n)for n in range(1, i + 1))
```

我们可以检查预期值：

[13]:
```
断言 exp[q] == 1/D(1) 断言 exp[q**2] == 1/D(2) 断言 exp[q**3] == 1/D(3) 断言 exp[q**4] == 1/D(4)
```

[14]:
```
断言 log[q] == -1/L(1) 断言 log[q**2] == 1/L(2) 断言 log[q**3] == -1/L(3) 断言 log[q**4] == 1/L(4)
```