arxiv:2509.15116v1 中译本

科学的机械化通过 Lean 形式化多级 Proj 构造来说明

Arnaud Mayeux and Jujian Zhang

努力将科学研究的某些方面机械化与科学发展从其最早的日子起就交织在一起。在 1961 年的论文《"科学机械化"》中，汉明讨论了几步：

C1 "当我们遇到一个长而复杂的代数问题时，而且这种问题在当今越来越常见，我们可以至少让机器检查一下代数是否正确，然后再继续进行进一步的推导。有些人目前正在开发这样的程序来检查代数问题。"

C2 "现在我离开了已知过程的区域，进入了推测的领域。我相信我们可以合理地期望一个代数检查程序不会存在太久，就会有人将目前正在开发的启发式方法适应到以更具创造性的方法进行代数运算的问题上。机器可以一次提供多个步骤，并且只需要一条证明的引导线索。启发式的成功程度越高，我们需要提供的步骤就越少。"

C3 "我们能走多远取决于在定理证明、模式识别和一般启发式方法等领域工作的人们的成功。已经取得的成就非常显著，但我确实觉得我们离能够给机器一个我们不知道任何证明的重要定理，并期望从它那里得到这个证明还相去甚远。"

C4 "假设，我们确实有一个如此强大的程序，当我们给机器提供黎曼假设时，它最终会输出大约 *500* 页紧密印刷的细节，声称这是一个证明。进一步假设，我们有一个独立的程序称为"定理证明检查器"。这样的程序比第一个更容易编写。并且假设所声称的证明通过了第二次测试。这是否构成该定理的证明？*[...]* 我倾向于认为实际结果是有些人会在此基础上进行扩展，并因此将其用作一个证明，所以对他们而言它就是一个证明，无论任何人类是否跟随并理解它。"

Arnaud Mayeux
Einstein Institute of Mathematics, The Hebrew University of Jerusalem
e-mail: arnaud.mayeux@mail.huji.ac.il

Jujian Zhang
Department of Mathematics, Imperial College London
e-mail: jujian.zhang19@imperial.ac.uk

C5 “我一直在谈论数学。虽然其他科学的细节会有所不同，但我认为可以做很多工作来使它们机械化。我可以想象一个常规程序，它可以接受现代物理学中的一个猜想，并推导出许多逻辑结果，还可以与已知结果进行多项检查。因此，在某些领域，物理学家可以花大量时间制定新模型，并让机器给出结果。”

这些引用直接摘自 [Ha61]。类似的思想可以在许多先前和后续的文献中找到。进一步的后果和愿景也很多。形式化定理证明器如 Lean（参见 [dMal15, dMU21, Av24, Bu24]）满足 C1 条件。Lean 足够易于使用，可以称为一个 “定理-验证检查器”（参见 C4）。在这份笔记中，我们讨论了多级 Proj 构造的 Lean 形式化 [MZ25]。多级 Proj 构造位于代数几何和 Lie 理论 [BS03, MR24] 的交汇处，这两个领域在理论物理学中都被广泛使用。多级 Proj 构造的形式化的 Lean4 代码大约有 8000 行，对应于 400 行的 LaTeX。这大量利用了 mathlib[Z23, Bual22, WZ22, Mathlib] 中可获得的模式论形式化的材料。看到在 AI 自动化形式化方面的进步和一个更直观的形式化界面（例如 LaTeX 友好型）将是很有用的。截至 2025 年春季，Copilot（Lean 2025 AI）仍然不足以帮助代数几何的形式化工作。Copilot 无法自行完成证明在某个非平凡分级环中 $0 \times 1 = 0$ 的事实，因为简单的计算涉及到关于次数的证明。与上述 C2 和 C3 相关，本项工作 [MZ25] 作为新的形式化数据，将有助于这些已经强大的工具的培训。我们现在介绍这份笔记的代数部分。受射影几何的启发，Grothendieck 定义了 $\boldsymbol{N}$-分次环 A [Gr61] 的 Proj 方案。这一构造在数千篇出版物中被使用和呈现。除了 Proj 的定义之外，大多数处理方式还研究了 Proj 上的拟凝聚层和扭曲层，并使用 Proj 来定义爆破。Grothendieck 的 Proj 构造已被 Zhang [Z23] 在 Lean 中形式化。因此，这个 Proj 构造似乎已经被广泛研究。然而，可以提出以下问题。为什么 Grothendieck 假设 A 是 $\boldsymbol{N}$ 分次的而不是由一个一般的交换幺半群或群分次? 我们的回答是: 从方案理论的角度来看没有理由。事实上，Brenner-Schroer 定义了一个任意有限生成阿贝尔群 [BS03] 分次环的 Proj。据我们所知，截至 2025 年，教科书从未提及 Brenner 和 Schroer 的工作。多分次 Proj 构造的非正式版本在数学中的各种构造中隐含或显式地出现（但没有提及 Brenner-Schroer）（参见。[MR24]). 本笔记的第一部分解释了 Brenner-Schroer 构造和新结果。第二部分讨论了形式化。本笔记基于同一时间段在不同地点发表的两个类似报告，其中一个于 2025 年 6 月 19 日在*数学的形式化与交互式定理证明器，剑*

桥大学等机构半研讨会上进行，另一个则在一个会议上第十六章。李理论及其在物理学中的应用，*2025* 年 *6* 月 *16* 日至 *22* 日，瓦尔纳进行。

1. 投影的定义及其行为

本节基于 [MR24]。参考文献 [BS03] 引入了主要定义，并且 [MR24] 后续进一步发展了该理论，提供了额外的重新表述和结果。如果给定的交换环 A 是对某个阿贝尔群 M 的 M-分级，我们将说 A 的一个子幺半群 S 是齐次的，如果 S 中的每个元素都是齐次的。在这种情况下，A_S 对于 A 关于 S 的局部化是典范的 M-分次。给定一个分级 A-模 Q，也可以考虑局部化模块 Q_S，它自然具有分级 A_S-模的结构。给定一个齐次子幺半群 $S \subset A$，我们将用 $\underline{S}$ 表示由 S 中元素的齐次除数组成的齐次子幺半群。注意我们有一个分级环的典范同构 $A_{\underline{S}} \cong A_S$. 如果 Q 是一个分级 A-模，我们也有一种识别 $Q_{\underline{S}} \cong Q_S$ 与 $A_{\underline{S}} = A_S$ 的作用相容。考虑一个阿贝尔群 M，以及一个交换的 M-分次环 A。

定义： 令 S 是 A 的一个齐次子集。我们记 $\deg(S)$ 为由 M 定义的子集 $\deg(S) := \{m \in M : \exists s \in S, s \in A_m\}$.

请注意，$\deg(\{0\}) = M$。更一般地说，如果 S 是 A 的齐次子幺半群，那么 $\deg(S)$ 是 M 的子幺半群，也表示为 $M[S\rangle$。

定义： 令 S 是 A 的一个齐次子半群。我们将由 $\deg(S)$ 生成的 M 的子群记为 $M[S] = M[S\rangle^{\mathrm{gp}} = M[S\rangle - M[S\rangle$。

定义： 若对于任意的 m 在 M 中都存在 $n \in \boldsymbol{Z}_{>0}$ 使得 nm 属于 $M[\underline{S}]$，即，同质子幺半群 S 被称为与 M 相关（如果 M 在上下文中是明确的，则只需称作相关）A 的。如果 $M/(M[\underline{S}])$ 是一个挠阿贝尔群。一组在 A 中的齐次元素 $\{a_i : i \in I\}$ 被称为 M-相关，如果由 a_i 生成的子幺半群是相关的。一个齐次元素 $a \in A$ 被称为 M-相关，如果该族 $\{a\}$ 是 M-相关的。

令 S 为 A 的齐次子幺半群。

定义： 局部化 A_S 的 0 阶部分 $(A_S)_0$ 记为 $A_{(S)}$，被称为关于 S 的 A 的药水。如果 Q 是一个分级的 A 模，我们也将用 $Q_{(S)}$ 表示 Q_S 的度为 0 部分 $(Q_S)_0$；它具有一个作为 $A_{(S)}$ 模的标准结构。

我们有典范同构 $A_{(\underline{S})} \cong A_{(S)}$ 和 $Q_{(\underline{S})} \cong Q_{(S)}$。如果 $\{a_i : i \in I\}$ 是 A 的同质元素的族，我们将用 $A_{(\{a_i : i \in I\})}$ 表示由 $\{a_i : i \in I\}$ 生成的 A 的子单 oid 相关的药剂；在情况 $\#I = 1$ 中，我们将用 $A_{(a)}$ 表示 $A_{(\{a\})}$。如果

S 和 T 是 A 的子幺半群，我们将用 ST 表示由 $S\cup T$ 生成的 A 的子幺半群，即 $ST=\{st : s\in S, t\in T\}$。当然，如果 S 和 T 是齐次的，那么 ST 也是齐次的。以下是使 Proj 构造起作用的关键结果。

命题（药剂的魔力）: 设 S 和 T 是 A 的齐次子幺半群。

我们有一个魔药环的典范同态

1. $A_{(S)} \to A_{(ST)}$
2. 假设 S 是相关的。固定一个子集 $T' \subset T$，它作为 $(A,\times)$ 的一个子半群生成 T，并且对于任何 t 在 T' 中，固定 $n_t \in \boldsymbol{N}_{>0}$ 和 $s_t, s'_t \in \underline{S}$ 使得 $\deg(t^{n_t}) = \deg(s_t) - \deg(s'_t)$。那么 $\frac{t^{n_t}s'_t}{s_t}$ 属于 $A_{(\underline{S})} \cong A_{(S)}$。此外，我们有一个 $A_{(S)}$-代数之间的规范同构，介于 $A_{(ST)}$ 和 $A_{(S)}$ 关于由 $\{\frac{t^{n_t}s'_t}{s_t} : t\in T'\}$ 生成的 $A_{(S)}$ 的子单 oid 的局部化之间。
3. 假设 S 是相关的，并且 T 作为 $(A,\times)$ 的子幺半群是有限生成的。由（1）中的环同态诱导的概形 morphism $\mathrm{Spec}(A_{(ST)}) \to \mathrm{Spec}(A_{(S)})$ 是一个概形的开浸没。
4. 设 $f_1,\ldots,f_n \in A$ 是相同次数的非零相关齐次元素。则我们有一个典范开浸入

$$\mathrm{Spec}(A_{(f_1+\cdots+f_n)}) \to \mathrm{Spec}(A_{(f_1)}) \cup \cdots \cup \mathrm{Spec}(A_{(f_n)})$$

其中右侧定义为仿射概形 $\mathrm{Spec}(A_{(f_i)})$ 沿着开子概形 $\mathrm{Spec}(A_{(f_i\cdot f_j)}) \subset \mathrm{Spec}(A_{(f_i)})$ 的粘合。

从现在起我们假设 M 是有限生成，并固定一个交换的 M-分次环 A。我们将所有相关的作为 $(A,\times)$ 的子幺半群有限生成的 A 的同次子幺半群集合记为 $\mathcal{F}_A$。

构造（Proj 作为粘合部分） 设 $\mathcal{F} \subset \mathcal{F}_A$ 是一个子集。对于每个 $S\in\mathcal{F}$，令 $D_\dagger(S)$ 为药剂 $A_{(S)}$ 的光谱。如果 $S,T\in\mathcal{F}$，仿射概形 $D_\dagger(ST)$ 可以规范地识别为 $D_\dagger(S)$ 的一个开子概形。对于每个 $S,T\in\mathcal{F}$，我们有等式 $D_\dagger(SS)=D_\dagger(S)$ 和 $D_\dagger(ST)=D_\dagger(TS)$.。此外，对于每个三元组 $S,T,U\in\mathcal{F}$，我们有 $D_\dagger(ST)\cap D_\dagger(SU) = D_\dagger(TS)\cap D_\dagger(TU)$.。现在，通过粘合，从这些数据中我们得到一个概形 $\mathrm{Proj}^M_{\mathcal{F}}(A)$，并且对于每个 $S\in\mathcal{F}$，有一个开浸入 $\varphi_S : D_\dagger(S) \to \mathrm{Proj}^M_{\mathcal{F}}(A)$，使得 $\mathrm{Proj}^M_{\mathcal{F}}(A) = \bigcup_{S\in\mathcal{F}} \varphi_S(D_\dagger(S))$.。实际上，我们将经常识别 $D_\dagger(S)$ 和 $\varphi_S(D_\dagger(S))$。在 $\mathcal{F}=\mathcal{F}_A$ 的情况下，方

案 $\mathrm{Proj}^M_{\mathcal{F}_A}(A)$ 将被表示为 $\mathrm{Proj}^M(A)$，或者当 M 从上下文中明确时，简写为 $\mathrm{Proj}(A)$。

命题： 方案 $\mathrm{Proj}^M(A)$ 是拟分离的。

命题（Proj 的函子性）： 设 $\Psi : A \to B$ 是一个 M-分次环同态。对于任意 $\mathcal{F} \subset \mathcal{F}_A$ 我们有一个模式的典范态射 $\mathrm{Proj}^M_{\Psi(\mathcal{F})}(B) \to \mathrm{Proj}^M_{\mathcal{F}}(A)$。

命题： 假设 $\Psi : A \to B$ 是满射。那么我们有 $\mathrm{Proj}^M_{\Psi(\mathcal{F}_A)}(B) = \mathrm{Proj}^M(B)$，并且典范态射 $\mathrm{Proj}^M(B) \to \mathrm{Proj}^M(A)$ 是一个闭浸。

命题： 令 M 和 M' 为两个有限生成的阿贝尔群。令 R 是一个交换环，并且令 A（分别地 A'）是一个交换的 M-分次（分别地 M'-分次）R-代数。然后对于自然的 $(M \times M')$ 分级在 $A \otimes_R A'$ 上，我们有一个典范同构

$$\mathrm{Proj}^{M\times M'}(A \otimes_R A') \cong \mathrm{Proj}^M(A) \times_{\mathrm{Spec}(R)} \mathrm{Proj}^{M'}(A').$$

令 Q 是一个 M-阶的 A-模。对于任意齐次子幺半群 $S \subset A$，我们已经考虑了 $A_{(S)}$-模 $Q_{(S)}$。

定义： 存在一个唯一的准相干 $\mathcal{O}_{\mathrm{Proj}^M(A)}$-模 $\widetilde{Q}$，使得 $\Gamma\big(D_\dagger(S), \widetilde{Q}\big) = Q_{(S)}$ 对于每一个 $S \in \mathcal{F}_A$。

一个 M-分级的 A-模 Q 将被称为*可忽略的*如果 $\widetilde{Q} = 0$。如果 Q 是一个分级的 A 模，并且 $\alpha \in M$，我们将用 $Q(\alpha)$ 表示与作为 A 模时与 Q 相同的 M 分级的 A 模，但其 M 分级由 $(Q(\alpha))_\beta = Q_{\alpha+\beta}$ 对 $\beta \in M$ 定义。

定义（扭曲层） 令 $\alpha \in M$。拟凝聚层 $\widetilde{A(\alpha)}$ 在 $\mathrm{Proj}^M(A)$ 上表示为 $\mathcal{O}_{\mathrm{Proj}^M(A)}(\alpha)$。如果 $\mathcal{Q}$ 是一个 $\mathcal{O}_{\mathrm{Proj}^M(A)}$-模层，我们设定 $\mathcal{Q}(\alpha) = \mathcal{O}_{\mathrm{Proj}^M(A)}(\alpha) \otimes_{\mathcal{O}_{\mathrm{Proj}^M(A)}} \mathcal{Q}$。

命题： 假设 A 是诺特环。对于任意的 $\alpha \in M$，拟凝聚层 $\mathcal{O}_{\mathrm{Proj}^M(A)}(\alpha)$ 是凝聚的。如果 Q 是一个由有限生成的 M-分级 A-模，那么 $\widetilde{Q}$ 是凝聚的。

一族 $S \in \mathcal{F}_A$ 将被称为*最大程度相关的*如果 $M[\underline{S}] = M$。我们将用 $\mathcal{F}^{\mathrm{m}}_A \subset \mathcal{F}_A$ 表示由最大程度相关的族组成的子集。我们现在讨论一下该条件的一些推论：$\mathrm{Proj}^M(A) = \bigcup_{S \in \mathcal{F}^{\mathrm{m}}_A} D_\dagger(S)$.

命题： 假设上述条件成立。对于任意的 $\alpha \in M$，拟凝聚 $\mathcal{O}_{\mathrm{Proj}^M(A)}$-模 $\mathcal{O}_{\mathrm{Proj}^M(A)}(\alpha)$ 是一个可逆层。

命题： 假设上述条件成立，并且 $\mathrm{Proj}^M(A)$ 是拟紧的。那么函子

$$\mathrm{L} : \mathrm{Mod}^M(A)/\mathrm{Mod}^M(A)_{\mathrm{neg}} \to \mathrm{QCoh}(\mathrm{Proj}^M(A))$$

是范畴的等价。

示例（旗流形作为射影空间） 设 k 是一个代数闭域，并且设 G 是在 $k.$ 上的连通约化群概形。设 T 是一个极大分裂环面，B 是一个 Borel 子群，使得 $B = TN$ 其中 N 是单根子群。然后 G/N 是准仿射的，环 $A := \Gamma(G/N, \mathcal{O}_{G/N})$ 可以规范地进行 $X^*(T)$-分级。我们有一个概形上的优势开浸入 $G/N \to \mathrm{Spec}(A)$。旗簇 G/B 与 $\mathrm{Proj}^{X^*(T)}(A)$ 相同。考虑一个有限维的 G 模 $\widetilde{V}$ 和一个 B 不变子空间 $V \subset \widetilde{V}$。然后我们可以考虑诱导的概形 $G \times^B V$, 即 乘积 $G \times V$ 除以由 $b \cdot (g, x) = (gb^{-1}, b \cdot x)$ 定义的（自由）作用 B 的商。它是一个在 G/B 上的向量丛。有一种规范的方式来将这样的概形视为 Proj。一个示例是当 $\widetilde{V}$ 是 G 的李代数，而 V 是 B 的单根子的李代数。然后 $G \times^B V$ 是 Springer 分解。

2. 形式化

与集合和子集相比，商类型是 Lean4 类型系统中的一个基本概念。因此，将分次环的次数-0 部分定义为商类型而不是分次环的子集更加自然和符合人体工程学。对于一个 ι-分次环 $A \cong \bigoplus_{i:\iota} \mathcal{A}_i$ 和子幺半群 $x \leq A$，我们定义局部化 A_x 的次数为 0 的部分为三元组 (i, n, d) 的等价类，其中 $i : \tau$ 且两者 n 和 d 都是元素 $\mathcal{A}_i$ 在等价关系下 $(i, n, d) \sim (i', n', d')$ 当且仅当等于 $\frac{n}{d} = \frac{n'}{d'}$ 在局部化环 A_x 中。

```
structure NumDenSameDeg where
  deg : $\iota$
  (num den :    deg)
  den_mem : (den : A)    x

def NumDenSameDeg.embedding (p : NumDenSameDeg    x) :=
  Localization.mk p.num  p.den, p.den_mem

def HomogeneousLocalization : Type _ :=
  Quotient (Setoid.ker <| NumDenSameDeg.embedding    x)
```

商的方法的优点是两方面的：

- 分子 n, 分母 d, 次数 i 和事实 $n, d \in \mathcal{A}_i$ 以及 $d \in x$ 距离选择公理的应用仅一步之遥。与集合论方法 $A_{(x)} = \left\{ f \middle| \exists i \in \iota, n \in \mathcal{A}_i \cap x, d \in \mathcal{A}_i, f = \frac{n}{d} \right\}$ 相比，数据的提取需要反复应用 AC。
- 商类型自带一个通用属性，这已经使得从 $A_{(x)}$ 定义函数变得方便。

形式化的实践常常迫使表述得更加清晰。在魔药的第二部分中，我们可以定义以下数据类型：

```
structure PotionGen where
  (index : Type*)
  (elem : index → A)
  (elem_mem :   t, elem t   T)
  (gen : Submonoid.closure (Set.range elem) = T.toSubmonoid)
  (n : index →  +)
  (s s' : index → A)
  (s_mem_bar :   t, s t   S.bar)
  (s'_mem_bar :   t, s' t   S.bar)
  (i i' : index → $\iota$)
  (t_deg :   t : index, (elem t : A)^(n t :  )     (i t - i' t))
  (s_deg :   t, s t     (i t))
  (s'_deg :   t, s' t     (i' t))

def PotionGen.genSubmonoid (T' : PotionGen S T) : Submonoid S.Potion :=
  Submonoid.closure
    {x |   (t : T'.index), x =
      S.equivBarPotion.symm (.mk
        { deg := T'.i t,
          num := (T'.elem t) ^ (T'.n t :  ) * T'.s' t,
            by simpa using SetLike.mul_mem_graded (T'.t_deg t) (T'.s'_deg t) ,
          den := T'.s t, T'.s_deg t ,
          den_mem := T'.s_mem_bar t }) }
```

因此，魔药的第二部分陈述可以表述为：对于每一个有限生成的相关齐次子幺半群 S,T 的 A 和任意 PotionGen S T T'，我们有一个同构关系在魔药 $A_{(ST)}$ 和局部化环 A_S 在 genSubmonoid T' 上。

```
def localizationRingEquivPotion (T' : PotionGen S T) :
    Localization T'.genSubmonoid  +* (S * T).Potion := ...
```

当然，在笔纸证明中，可以定义辅助结构以达到同样的效果，但缺少这样的结构并不会给作者带来负担，因此，作者较少有动力去定义这些结构。然而，在形式化过程中，这种负担落到了作者身上——没有这些结构，不仅会让读者难以阅读证明过程，也会使作者的写作难上加难。

另一方面，形式化带来的清晰性和严谨性先验的需要形式化者付出额外的谨慎 — 在纸上笔算中看似不必要的步骤，突然需要向计算机解释。人们可以承认这是因为类型理论与集合论不同，不应期望逐字逐句地采用相同的做法，并且在许多地方确实缺少中间步骤。一个明显的区别是等号的概念 — 考虑以下示例，如果 S 和 T 是 A 的两个相等的齐次

子单群，我们应该能够在某种程度上将 $A_{(S)}$ 和 $A_{(T)}$ 视为同一个对象。在形式化之前，我们认为 $A_{(S)}$ 和 $A_{(T)}$ 是相等的，但是在集合论中，$A_{(S)}$ 和 $A_{(T)}$ 分别是 A_S 和 A_T 的子环，因此，字面上的等式 $A_{(S)} = A_{(T)}$ 简单来说就是不成立的。在类型理论中，虽然我们可以证明等式 $A_{(S)} = A_{(T)}$ 成立，但由于需要过多的 rewrite 等式 $S = T$，这个等式提供的函数作用不大。因此，$A_{(S)}$ 和 $A_{(T)}$ 是相同的，因为以下同构关系，注意该同构的两个方向都是使用商类型的基本性质而不是直接从等式 $S = T$ 构建的：

```
def potionEquiv {S T : HomogeneousSubmonoid } (eq : S = T) :
    S.Potion  +* T.Potion :=
  RingEquiv.ofHomInv
    (HomogeneousLocalization.map _ _ (RingHom.id _) ... :
      S.Potion →+* T.Potion)
    (HomogeneousLocalization.map _ _ (RingHom.id _) ... :
      T.Potion →+* S.Potion)
    ...
```

明确“平凡”同构的要求不仅是类型理论的产物，它也是严谨性所必需的：考虑在 $A = B$ 的情况下，笛卡尔积 $A \times B \cong B \times A$ 之间的同构，在没有更多细节 $A \times A \cong A \times A$ 的情况下是模棱两可的。

References

[Av24] J. Avigad, *Automated reasoning for mathematics,* in *Automated reasoning. Part I*, 3–20, Lecture Notes in Comput. Sci. Lecture Notes in Artificial Intelligence, 14739 , Springer, Cham, 2024.

[BS03] H. Brenner, S. Schroer, *Ample families, multihomogeneous spectra, and algebraization of formal schemes* Pacific J. Math. 208 (2003), no. 2, 209 – 230.

[Bual22] K. Buzzard, C. Hughes, K. Lau, A Livingston, R. F. Mir, and S. Morrison, *Schemes in Lean,* Exp. Math. **31** (2022), no. 2, 355–363.

[Bu24] K. Buzzard, *Mathematical reasoning and the computer,* Bull. Amer. Math. Soc. (N.S.) **61** (2024), no. 2, 211–224.

[Gr61] A. Grothendieck, *EGA : II. Etude globale elementaire de quelques classes de morphismes* Publications mathématiques de l'I.H.E.S., tome 8 (1961), p. 5-222

[Ha61] R. Hamming, *The mechanization of science* Proceedings of the 1961 16th ACM national meeting. Association for Computing Machinery, New York, United States

[Mathlib] Lean Community, *Mathlib*, https://leanprover-community.github.io/mathlib-overview.html

[MR24] A. Mayeux and S. Riche, *On multi-graded Proj schemes*, to appear in Publ. Res. Inst. Math. Sci.

[MZ25] A. Mayeux and J. Zhang, *Multi-graded Proj construction in Lean4*, https://github.com/ProjConstruction/Proj (2025)

[dMU21] L. de Moura and S. Ullrich, *The Lean 4 theorem prover and programming language,* in *Automated deduction—CADE 28*, 625–635, Lecture Notes in Comput. Sci., 12699, Springer, Cham, 2021.

[dMal15] L. de Moura, S. Kong, J. Avigad, F. van Doorn and J. von Raumer, *The lean theorem prover (system description)*, in *Automated deduction—CADE 25*, 378–388, Lecture Notes in Comput. Sci. Lecture Notes in Artificial Intelligence, 9195 , 2015 Springer, Cham.

[Z23] J. Zhang, *Formalising the Proj construction in Lean*, in *14th International Conference on Interactive Theorem Proving*, Art. No. 35, LIPIcs. Leibniz Int. Proc. Inform., 268 (2023).

[WZ22] E. Wieser and J. Zhang, *Graded rings in Lean's dependent type theory,* in *Intelligent computer mathematics*, 122–137, Lecture Notes in Comput. Sci. Lecture Notes in Artificial Intelligence, 13467 , Springer, Cham, 2022.