arXiv:2603.00896v1 [math.CT] 1 Mar 2026

Lean 中对称幺半范畴的去偏

Robin Carlier*

2026年2月28日

摘要

我们在数学库 `Mathlib` 的框架内，使用 Lean 4 对对称幺半范畴的去偏过程进行了形式化。具体做法是，将对称幺半范畴的数据扩展为一个从有限集跨度的 (2,1)-范畴到 Cat 的伪函子，从而编码更高元数的张量积及其凝聚。该构造依托于 Mac Lane 相干定理的形式化；这一形式化采用了 Piceghello 将自由对称幺半范畴表示为对称列表的方式，并通过适当的 Kleisli 双范畴对普遍公式进行编码。

1 引言

在为数学建立基本代数层级时，人们将交换幺半群定义为一个集合 M，它配备了单位元 0_M，以及满足交换律和结合律的复合运算 $\cdot +_M \cdot : M \times M \to M$。给定此交换幺半群 M 中的元素 a, b, c, d，这些数据使我们能够对诸如 $(a +_M b) +_M (c +_M d)$、$(a +_M (c +_M b)) +_M d$ 或 $b +_M (a +_M (d +_M c))$ 这样的表达式赋予形式上的意义；而函数 $+_M$ 所满足的性质则让我们可以证明这些表达式彼此相等。在数学应用中，将二元运算扩展到更高元数的运算，很快就变得十分重要。例如，像下面这样的公式

$$\sum_{\rho \in \mathrm{Irr}(G)} (\dim(\rho))^2 = \mathrm{Card}(G)$$

对于有限群 G，其中涉及一个求和算子，它定义在底层幺半群中任意有限元素集上，而这些元素没有规范的顺序。仅仅是这个算子的存在，就需要证明它定义良好，也就是说，所得结果不会因底层加数列表的枚举顺序或结合方式不同而改变。

*作者感谢法国国家科研署项目“HQDIAG”（ANR-21-CE40-0015）提供的支持。

在软件证明助手的背景下，诸如Mathlib [19][12]这样的形式化数学库，必须经过这一过程，才能为用户提供使用此类有限和的框架。例如， Mathlib 有一个声明

```
def Multiset.sum {M : Type u} [AddCommMonoid M] : Multiset M → M := …
```

它通过将 定义为类型 模置换的商，来实现对 中多重集元素求和的功能。声明 则基于该定义，为用户提供了不依赖括号顺序的求和。[1]

从交换幺半群到对称幺半范畴

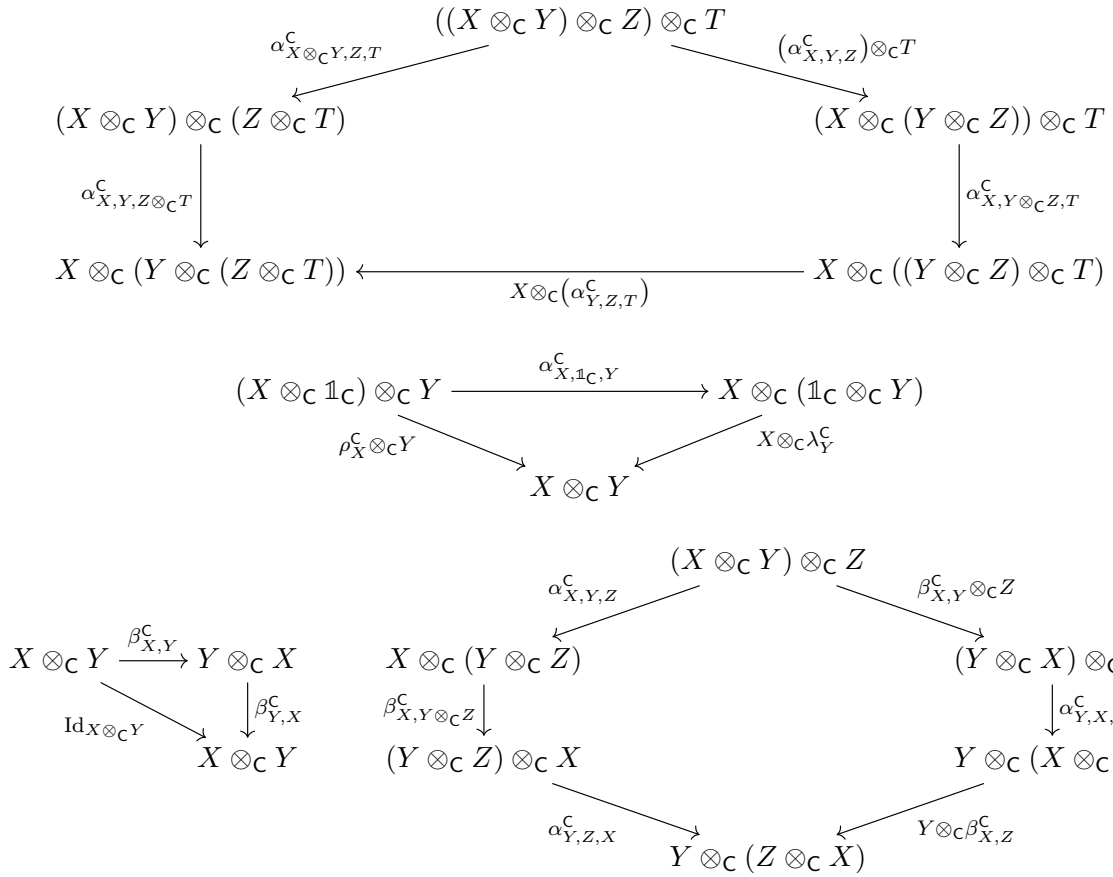
交换幺半群自然的范畴化形式是对称幺半范畴。对称幺半范畴由 Saunders Mac Lane 独立提出 [15]，旨在抽象出具有积的范畴中的范畴积、模上的张量积以及链复形的张量积所共有的结构。这一广为研究的概念如今被视为可赋予范畴的众多伪代数结构中的一个经典例子。

对称幺半范畴的经典定义（[15], [5, 定义 6.1.2]）由一个范畴 C 构成，该范畴配备了一个张量积双函子 $\otimes_{\mathsf{C}} : \mathsf{C} \times \mathsf{C} \to \mathsf{C}$、一个单位对象$\mathbb{1}_{\mathsf{C}}$，以及对 C 的所有对象 X, Y, Z，自然同构

$$\begin{aligned} \alpha^{\mathsf{C}}_{X,Y,Z} &: (X \otimes_{\mathsf{C}} Y) \otimes_{\mathsf{C}} Z \xrightarrow{\sim} X \otimes_{\mathsf{C}} (Y \otimes_{\mathsf{C}} Z) \\ \lambda^{\mathsf{C}}_{X} &: \mathbb{1}_{\mathsf{C}} \otimes_{\mathsf{C}} X \xrightarrow{\sim} X \\ \rho^{\mathsf{C}}_{X} &: X \otimes_{\mathsf{C}} \mathbb{1}_{\mathsf{C}} \xrightarrow{\sim} X \\ \beta^{\mathsf{C}}_{X,Y} &: X \otimes_{\mathsf{C}} Y \xrightarrow{\sim} Y \otimes_{\mathsf{C}} X \end{aligned}$$

分别称为结合子、左单位子、右单位子和编织，并满足以下图表可交换：

[1]这些求和仍然包含了对结合律的特定选择，这是因为 被定义为从左侧折叠求和运算。随后会在例如 中证明这种选择无关紧要。沿着证明思路追踪下去，问题归结为 ，最终会在这里用到结合律。



与交换幺半群的情形不同，对称幺半范畴中的结合律和交换律被表达为一种 结构 ，而不是一种 性质。这是范畴论的标准理念之一：范畴中对象的等式不是范畴等价下的不变量，因此应以选定同构这一结构取代相等这一性质。这些性质在高一层的范畴论中体现为涉及这些同构的图表的可交换性。

上述表述提供了一种相当简单直接的方式来定义对称幺半范畴：只需提供低元数据和凝聚。另一方面，与交换幺半群的情形类似，有些应用需要从无偏观点看待对称幺半范畴，并要求每种可能的元数下都有张量积和凝聚。举例来说，给定一个具有可数余积的对称幺半范畴 C，且 C 的张量积在每个变量上都保持适当的余极限，则对象 $c \in \mathsf{C}$ 上的自由交换幺半群可以识别为以自然数集为指标的对象余积 $\mathrm{Sym}^{\mathsf{C}}_n(c)$，其中 $\mathrm{Sym}^{\mathsf{C}}_n$ 将对象 a 映射为 C 中以 $\mathrm{B}\Sigma_n$为指标的图表的余极限，该图表对应于带有其自然 Σ_n 作用的对象 $a^{\otimes n}$ 。

除非将 C 的二元张量积及其对称性扩展到更高元数，否则甚至无法陈述这一事实。

这种形式上的问题通常通过调用 Mac Lane 著名的相干定理来解决[15, 定理 4：2] 粗略地说，该定理指出，在两个仅括号位置不同的迭代张量积之间，由结合子和单位子构成的“典范”同构只有一个；对于迭代张量积变量的给定置换，所对应的“典范”态射也只有一个。这里我们有意没有精确陈述该定理，因为本文会给出更精确的表述（见定理 1.3）。

对称幺半高范畴

无偏对称幺半范畴的另一个非常重要的应用，体现在从高范畴观点看待对称幺半范畴。我们在上一段给出的对称幺半范畴定义，在范畴层级提升时并不容易扩展：三角形、五边形、六边形关系以及对称性恒等式，都是范畴 C 中（1-）态射之间的等式；如果将 C 换成双范畴，那么所有这些等式都需要替换为范畴 C 中适当的 2-同构数据，并且这些 2-同构还必须满足新的相干恒等式（例如，参见[18, 附录 C] ，其中给出了对称幺半双范畴的代数描述）。若要进一步确信这种代数方法在实践中很难扩展，我们可以看看 [10]中将（非对称）幺半三范畴定义为单对象四范畴的方式。

在 $(\infty,1)$-范畴理论的背景下，人们已经研究了对称幺半 $(\infty,1)$-范畴的若干等价概念：Lurie 的 ∞-算畴 [13], Moerdijk 与 Weiss 的树状集合 [16], Cranch 的 Lawvere 对称幺半 $(\infty,1)$-范畴 [6],仅举几例。

上述所有模型的相似之处在于，对称幺半 $(\infty,1)$-范畴的数据被（直接地，或经由适当的比较定理）解释为一个取值于 $(\infty,1)$-范畴之 $(\infty,1)$-范畴（或等价地，纤维化）的伪函子，其源为一个已经包含更高元数数据与对称性的图表范畴：对于 ∞-算畴，是带基点有限集的范畴（亦即 Segal 的范畴 Γ）；在树状集合的设定中，是有根树的范畴 Ω；而对于 Cranch 的 Lawvere 对称幺半 $(\infty,1)$-范畴，则是有限集的跨的 (2,1)-范畴。在所有这些方法中，这些源范畴编码了所得对称幺半 $(\infty,1)$-范畴中预期应当“交换”的全部图表，而将数据封装为一个到 Cat_∞的 ∞-函子则实现了这些图表。

$(\infty,1)$ 范畴论的一个重要特点是，当某个概念应用于来自普通范畴的对象时，几乎总能归结为普通范畴论中的相应概念：例如，将普通范畴 C 视为 $(\infty,1)$-范畴时，从 x 到 y 的态射 1-群胚与态

射集合 x 到 y 在 C 中同构；将集合视为离散 ∞-群胚也是如此。类似地，当图表的源和目标是普通范畴时，$(\infty,1)$-范畴论中的极限和余极限概念会归结为普通的极限和余极限。对称（以及非对称）幺半范畴也不例外：将对称幺半普通范畴视为一个$(\infty,1)$-范畴时，其数据等价于对称幺半$(\infty,1)$-范畴的数据。这样的论述离不开“去偏”这一步，即将二元张量积扩展到具有适当内部对称性的任意有限对象族。通常会在这一步提及 Mac Lane 相干定理，不过对此的援引未必总是十分明确（[13, 构造 2.0.0.1，（ii）和（iii）]）。关于文献中少数几处实际详细阐述了对称幺半范畴去偏完整过程的例子（尽管并非以高阶范畴的术语表述），请参阅 [11]。

Lean 中当前形式化的进展

在 Lean 4 证明助手中，更具体地说，在数学库 `Mathlib`中，对称幺半范畴一直采用经典的有偏方法进行形式化。这非常合理，因为这是最基础的方法（`Mathlib`的定义只需在一个文件中完成，该文件仅导入范畴和同构的定义），而且这种方法构造出的构造子所需的数据和证明义务最少。不过，定义上的简便也有代价：虽然可以优雅地进行初等操作，但一些更复杂的构造却不易完成，例如前面提到的自由交换幺半群对象的描述。寻求无偏定义的另一个原因，是为前文提到的高阶范畴视角提供必要的“衔接”。看来 `Mathlib` 最终很可能会纳入高阶范畴的基础性结果，因此建立这种联系至关重要。这一看法基于这样的观察：`Mathlib` 已经包含与高阶范畴理论的拟范畴基础相关的定义，例如拟范畴、单纯范畴的单纯神经、不断扩展的单纯同伦论内容，以及一个正在进行的项目，该项目旨在将 Riehl-Verity 的 ∞-宇宙理论和形式范畴论形式化到一个∞-宇宙中。随着这套理论的发展，将日益需要把 `Mathlib`中已有的普通范畴论形式化所产生的对象，重新诠释为相应的高阶范畴对象。

（非对称）幺半范畴的相干定理已在 `Mathlib` 中形式化，但即便是非对称幺半范畴，也尚无形式化的去偏。

结果

借用 [14, 定义 00AL], 中的术语，我们将双范畴 B 的髓 称为通过舍弃 B 中不可逆的 2-态射而得到的双范畴；我们将其记作

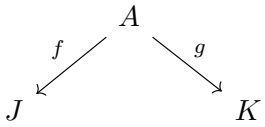
Pith(B)。给定一个具有拉回的范畴 C，我们将 C 中的跨度双范畴记作 Span(C)。

我们的项目所形式化的主要结果如下：

定理 1.1. 设 $\left(\text{范畴 } \mathsf{C}, \otimes_{\mathsf{C}}, \mathbb{1}_{\mathsf{C}}, \alpha^{\mathsf{C}}, \lambda^{\mathsf{C}}, \rho^{\mathsf{C}}, \beta^{\mathsf{C}}\right)$ 为一个对称幺半范畴。存在一个伪函子

$$\mathsf{C}^{\otimes} : \mathrm{Pith}\left(\mathrm{Span}\left(\mathrm{Fin}\right)\right) \to \mathrm{Cat}$$

它将有限集 J 映到积范畴范畴 C^J，并将跨度



映到函子 $g_! f^*$。这些函子定义如下：

- 函子 f^* : 范畴 $\mathsf{C}^J \to$ 范畴 C^A 是沿 f的前复合。
- 函子$g_!$: 范畴 $\mathsf{C}^A \to$ 范畴 C^K 将一个以 A为指标的族$(x_a)_{a\in A}$ 映到以 K为指标的族

$$\left(\bigotimes_{a\in g^{-1}(\{k\})} x_a\right)_{k\in K}.$$

该定理表明，按照通常的有偏版本定义的对称幺半范畴，在 $(\infty,1)$-范畴的术语中定义了一个 Lawvere 对称幺半[6]。

注 1.2. 有些读者可能会对我们选择用 Cranch 的 Lawvere 对称幺半范畴来表述这一结果，而非采用 Lurie 的∞-运算子感到意外；后者是（$\infty,1$）-范畴文献中最常用的模型。要完整证明 Lawvere 对称幺半（$\infty,1$）-范畴的（$\infty,1$）-范畴与 Lurie 风格的对称幺半（$\infty,1$）-范畴等价，需要做一些工作（参见 [2, 命题 C.1] ，其中给出了相当简洁的证明）。幸运的是，比较结果中较容易的方向，是将 Lawvere 对称幺半（$\infty,1$）-范畴转化为 Lurie 风格的对称幺半（$\infty,1$）-范畴：具体做法是沿着一个显式定义的伪函子进行前复合，该伪函子从 Fin_*（视为局部离散双范畴）映至 Pith (Span (Fin))，然后取 Grothendieck 构造（这是直化/反直化等价中较容易的方向）。因此，我们的结果很容易转化为从带基点有限集范畴构造一个伪函子。另一方面，根据一个从带基点有限集出发的合适伪函子的 数据，要从有限集跨度双范畴的髓构造出某种对象则更具挑战性，因为这需要沿另一个伪函子对伪函子作右 Kan 延拓，而非对其作限制。

我们认为，这项工作首次对定理 1.1中的陈述给出了机械化证明。

正如本引言前文所述，证明上述定理的一个主要要素是对称幺半范畴的相干定理。我们实现了该定理的以下形式：

定理 1.3。设 J 是一个集合。 J 上的自由对称幺半范畴（作为对称幺半范畴）等价于有限集及其上的双射所构成的范畴 $\mathrm{Core}\left(\mathrm{Fin}_{/J}\right)$，该范畴赋有由集合上的余笛卡尔对称幺半结构所诱导的对称幺半结构。

S. Piceghello 此前在 Ho- motopy 类型论的框架下，将某个类型上的自由对称幺半范畴与对称列表群胚相识别，从而形式化了相干定理的一部分，见 [17]；但对称列表的态射与置换群之间的最终联系，在他的工作中仅得到部分形式化。我们将 Piceghello 的工作从同伦类型论框架移植并调整到 Lean/ 中范畴这一非 HoTT 框架，`Mathlib`；我们在此的主要贡献，是通过用适当考克斯特群的元素为态射赋予标签，形式化地将对称列表的态射与置换相识别（定理 2.8和推论 2.9）。

我们还认为，本工作对于阐释如何精确运用相干定理来得到定理1.1并厘清这一过程也颇具价值。这些一般性思想在数学界当然已为人所知，但实际细节往往被留给读者自行补充。

我们的形式化成果已在线公开[2]。完整代码库约含 12.5kLoC（不计注释）。我们的代码使用 Lean工具链 `leanprover/lean4:v4.28.0`编译，并依赖于 `Mathlib` 的提交 `8f9d9cff6bd728b17a24e163c9402775d9e6a365`。

大纲

在第 2 节中，我们聚焦于相干定理 1.3。首先，我们在 2.1 中介绍对称列表范畴，并将 [17] 中的工作改编到 Lean/`Mathlib` 的环境中。该范畴将作为集合上自由对称幺半范畴的便捷模型，在本文后续部分发挥核心作用。在 2.2 中，我们研究对称列表范畴中的态射：首先简要回顾并形式化 A 型考克斯特群$_n$ 与置换之间的联系，见 2.2.1；随后在 2.2.2 中利用这些内容，通过证明对称列表的态射可以忠实地标记为 A 型考克斯特群的元素，形式化说明它们与置换一一对应（定理 2.8）。在 2.3 中，我们遵循 Piceghello 方法证明对称列表确实构成自由对称幺半范畴，从而完成定理 1.3 的证明。

[2] https://github.com/robin-carlier/SymmMonCoherence/tree/2070e1d536854a49d4c4f98cd73857ba40c4cb92

在第 3 节中，我们回顾跨度双范畴的基础知识，并说明其形式化过程。随后，我们在定义 3.2 中介绍精确定义，以刻画如何从具有拉回的范畴的跨度双范畴的髓中定义伪函子。

在第 4 节中，我们构造必要的数据，以建立定理 1.1 的伪函子，所用机制来自第 3 节。在 4.1 中，我们首先将目标双范畴从 Cat 抽象为合适的 Kleisli 双范畴 $\mathcal{K}^{\mathrm{op}}{}_{\mathrm{SList}}$，以处理对称幺半范畴理论，并证明可以将一个对称幺半范畴解释为取值于 Cat 的伪函子，其定义域为该 Kleisli 双范畴的反范畴（命题 4.2）。在 4.3 中，我们说明如何利用相干定理 1.3 来证明：在良好情形下，Kleisli 双范畴中的计算表现得如同带有多重集的矩阵计算。最后，我们构造伪函子 $\Lambda : \text{髓}\,(\text{跨度}(\mathrm{Fin})) \to \mathcal{K}^{\mathrm{op}}{}_{\mathrm{SList}}$，见命题 4.12，从而完成定理 1.1 中所述伪函子的构造。

2 相干定理

2.1 对称列表：通过生成元与关系呈现范畴

（非对称）幺半范畴的相干定理早已由 Markus Himmel 在`Mathlib` 中完成形式化。他遵循的是 Beylin 和 Dybjer 提出的经典归一化证明[3]；该证明适用于基于类型理论的证明助手，最初是在证明助手 ALF 中形式化的。

该证明的一个关键思想，是将定理表述为关于类型上的自由幺半范畴的命题：这类自由幺半范畴是群胚，且任意一对对象之间至多有一个态射。根据构造，类型 C 上的自由幺半范畴中的态射，恰好是由结合子、左或右单位子，以及其他态射的张量积构造而成的。因此，“每个图表都可交换”这一经典说法便由此表述编码。为了证明自由幺半范畴中态射的这一性质，我们对类型 T 上的自由幺半范畴中的对象进行归一化，从而实质上证明该范畴等价于以列表集合为对象的离散范畴。

对称幺半范畴的情形更为复杂，因为我们不能指望自由对称幺半范畴中任意两个对象之间至多只有一个态射：存在一些对称幺半范畴 C，其中恒等式 $\beta^{\mathsf{C}}_{c,\,c} = \mathrm{Id}_{c\otimes c}$ 并不成立，尽管这些态射的源与目标相同。我们仍可合理地期望，自由对称幺半范畴的结合部分和含单位元部分能够规范化为类似列表的结构（就像遗忘编织后所能做到的那样），但由于存在编织同构，所得的规范对象范畴并非离散范畴，而会在列表上形成非平凡的范畴结构。Piceghello 将这种范畴结构 [17, Def. 4.11] 称为对称列表，它是多重集的一种直接范畴化形式，我们将在此进行介绍：

定义 2.1。 *[17, 定义 4。11]* 设 J 是一个集合。其上的对称列表范畴J记作$\mathrm{SList}(J)$定义为由以下生成元与关系给出的范畴：

- $\mathrm{SList}(J)$的对象是由 J 中元素组成的列表。
- 态射由以下（归纳）规则生成：

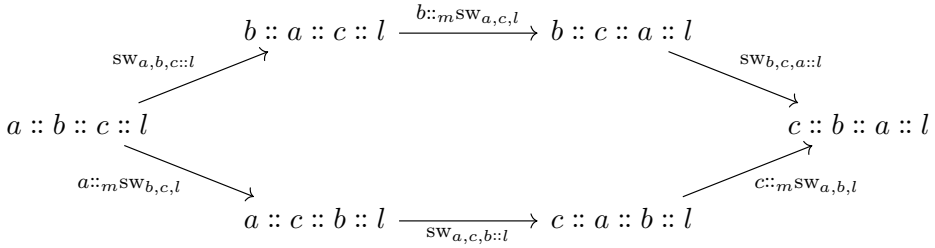
  1. 对任意 $a, b \in J$ 和 $l : \mathrm{SList}(J)$，存在一个态射

  $$\mathrm{sw}_{a,b,l} : (a :: b :: l) \to (b :: a :: l).$$

  2. 若 $f : a \to b$ 是一个态射，则存在一个态射 $(x ::_m f) : (x :: a) \to (x :: b)$.

- 态射满足以下关系：

  1. 构造 $f \mapsto x ::_m f$ 保持复合和恒等态射。*2.* 态射 $\mathrm{sw}_{a,b,l}$ 关于 l 是自然的（将 $x \mapsto (x :: -)$ 解释为函子时，使用前述关系）。*3.* 对称关系 $\mathrm{sw}_{b,a,l}\mathrm{sw}_{a,b,l} = \mathrm{Id}_{a::b::l}$ 成立。*4.* 图表



对每个 $a, b, c \in J$和 $l \in \mathrm{SList}(J)$均成立。

在同伦类型论的语境中，[17, 定义 4.11] 正是在该语境下进行的，这一定义可以立即实现为一个（1-截断的）高阶归纳类型。在Mathlib的语境中，我们无法直接作出这样的定义，而必须手动将 HoTT 中的 1-截断定义翻译并解释到由Mathlib提供的群胚模型中。

因此，形式化上述定义需要若干中间步骤，这些步骤共同构成了通过生成元与关系给出范畴的标准构造。在 Mathlib提供的基础设施中，这些步骤的形式如下

1. 定义一个类型 作为对象的生成元，并引入一个 该生成元类型上的实例，为其赋予生成箭头。

2. 取上一步所得箭图上的路径范畴 ，从而得到一个自由范畴，其中箭头是生成态射的形式复合（路径）。

3. 定义一个关系 ，作用于 中的态射，与最终范畴中的关系相对应。在 Mathlib中，这类关系的类型是 。

4. 取 出 关于上一点中关系 的商范畴 nt.

我们在本例中按此方案进行：

```
inductive SListQuiv (C : Type u) where
  | nil : SListQuiv C
  | cons (head : C) (tail : SListQuiv C) : SListQuiv C

infixr:67 " ::… " ⇒ SListQuiv.cons

inductive Hom {C : Type u} : SListQuiv C → SListQuiv C → Type u
  | swap (x y : C) (l : SListQuiv C) :
      Hom (x ::… (y ::… l)) (y ::… (x ::… l))
  | cons (z : C) {l l' : SListQuiv C} :
      Hom l l' → Hom (z ::… l) (z ::… l')

instance : Quiver (SListQuiv C) where Hom := Hom
structure FreeSListQuiv where p : Paths (SListQuiv C)
instance : Category (FreeSListQuiv C) :=
  inferInstanceAs (Category ◁ InducedCategory _ FreeSListQuiv.p)
```

在 HoTT 环境中，构造子 是隐式的：由于它在 HoTT 中是一个函数，因此会自动赋予构造子cons 对路径的作用。在非 HoTT 环境中解释此定义时，必须添加额外的构造子，并且必须手动将其扩展为路径范畴上的自函子，以表示它对路径的作用。在下面的代码块中， 是从 到 的包含预函子， 是生成交换态射的记号，而 ::_ （分别是::_m）表示将 构造子cons 扩展到

```
inductive HomEquiv : HomRel (FreeSListQuiv C)
  | swap_naturality (X Y : C) {l l' : SListQuiv C} (f : l ⟶ l') :
      HomEquiv
        (β₁_ X Y ((ι C).obj l) ≫
          (Y ::_m (X ::_m ((ι C).map f))))
        ((X ::_m (Y ::_m ((ι C).map f))) ≫
          (β₁_ X Y ((ι C).obj l')))
  | swap_swap (X Y : C) (l : FreeSListQuiv C) :
      HomEquiv (β₁_ X Y l ≫ β₁_ Y X l) (𝟙 _)
  | triple (X Y Z : C) (l : FreeSListQuiv C) :
      HomEquiv
        (β₁_ X Y (Z ::_ l) ≫ (Y ::_m (β₁_ X Z l)) ≫ β₁_ Y Z (X ::_ l))
        ((X ::_m (β₁_ Y Z l)) ≫ β₁_ X Z (Y ::_ l) ≫ Z ::_m (β₁_ X Y l))
```

```
  | cons (X : C) {l l' : FreeSListQuiv C} (f f' : l ⟶ l') :
      HomEquiv f f' → HomEquiv (X ::_m f) (X ::_m f')

def SList := CategoryTheory.Quotient (FreeSListQuiv.HomEquiv C)
  deriving Category
```

同样，由于 `Mathlib` 的框架具有非 HoTT 性质，为使定义健全，必须添加额外的关系：[redacted] 和 [redacted] 在 HoTT 表述中并未明确出现，因为 HoTT 中的高阶归纳类型保证所有构造子都是自然的，并且关于路径具有函子性。

注 2.2. 除了将 HoTT 环境调整为 Lean 的非 HoTT 环境中的解释之外，我们的构造与 Piceghello 的构造还有两处细微差异。

1. 如果要严格地在 `Mathlib` 的群胚所提供的“群胚模型”中解释所有内容，我们就需要用 [redacted] 上的自由群胚替换 [redacted] 上的自由范畴，以确保始终处于群胚的范畴之内。这样一来，就会为生成态射的形式逆元添加额外的生成元，还会添加额外的关系，以确保这些形式逆元在商范畴中确实定义了逆元。对于对称列表，很容易证明根据我们的定义得到的范畴是一个群胚，因此我们更倾向于采用范畴环境中更直接的构造。

2. 我们的定义不那么一般：如果我们严格按照 [17] 中的构造，在群胚模型中进行诠释，所构造的将是定义在一个群胚上的对称列表，而不是定义在一个集合上的对称列表（后者对应于离散群胚的情形）。接下来，我们还需要添加额外的关系，以确保生成态射相对于 C中的每个参数都满足自然性，这会使范畴的表示以及对其态射的研究变得更加复杂。为了陈述我们版本的相干定理并对称幺半范畴的去偏，这种额外的一般性并非必需，因此我们不予实现。

对称列表范畴作为由生成元与关系表示的范畴所具有的泛性质，可以明确表述如下：

引理 2.3. 设D 是一个范畴。

(i) 假设给定以下数据：

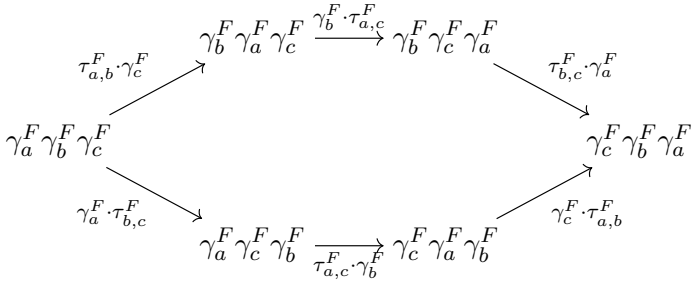
- 一个对象 F_{nil} ∈D。
- 对于每个 $c \in C$，一个自函子 γ_c^F: D →D。

对于所有 $a, b \in C$，一个自然同构

$$\tau_{a,b}^F : \gamma_a^F \gamma_b^F \xrightarrow{\sim} \gamma_b^F \gamma_a^F .$$

假设数据满足以下条件：

- 对所有 $a, b \in C$，等式 $\tau^F_{a,b}\tau^F_{b,a} = \mathrm{Id}$ 成立。
- 对所有 $a, b, c \in C$，图表



交换。

该数据定义了一个函子 $F : \mathrm{SList}(C) \to \mathsf{D}$，并配备同构

$$\begin{aligned} \upsilon_{\mathrm{nil},F} : &\quad F([\,]) &&\simeq F_{\mathrm{nil}}, \\ \upsilon_{\mathrm{cons},F,c} : &\quad F \circ (c :: \cdot) &&\simeq \gamma^F_c \circ F, \end{aligned}$$

并且使得 $F(\mathrm{sw}_{a,b,l})$ 通过这些同构与 $\left(\tau^F_{a,b}\right)_{F^{(l)}}$ 等同。

(ii) 设 $F, G : \mathrm{SList}(C) \to \mathsf{D}$ 为函子。假设给定如下数据 ：

- 态射 $\phi_{\mathrm{nil}} : F([\,]) \to G([\,])$。
- 对于每个 $c \in C$, $l \in \mathrm{SList}(C)$ 以及态射 $\mathrm{ind} : F(l) \to G(l)$，都有一个态射 $\phi_{c,l}(\mathrm{ind}) : F(c :: l) \to G(c :: l)$。

进一步假设以下条件成立：

- 对于所有 $a, b \in C$，对于所有 $l \in \mathrm{SList}(C)$ 以及对于所有态射 $\mathrm{ind} : F(l) \to G(l)$，图表

$$\begin{array}{ccc} F(a :: b :: l) & \xrightarrow{F(\mathrm{sw}_{a,b,l})} & F(b :: a :: l) \\ {\scriptstyle \phi_{a,b::l}(\phi_{b,l}(\mathrm{ind}))}\downarrow & & \downarrow{\scriptstyle \phi_{b,a::l}(\phi_{a,l}(\mathrm{ind}))} \\ G(a :: b :: l) & \xrightarrow[G(\mathrm{sw}_{a,b,l})]{} & G(b :: a :: l) \end{array}$$

交换。

- 给定 $c \in C$，一个态射 $f : l \to l' \in \mathit{SList}(C)$，以及态射 $\mathit{ind}_l : F(l) \to G(l)$、$\mathit{ind}_{l'} : F(l') \to G(l')$，它们满足 $\mathit{ind}_{l'} \circ F(f) = G(f) \circ \mathit{ind}_l$，则图表

$$\begin{array}{ccc} F(c :: l) & \xrightarrow{F(c ::_m f)} & F(c :: l') \\ {\scriptstyle \phi_{c,l}(\mathrm{ind}_l)}\downarrow & & \downarrow{\scriptstyle \phi_{c,l'}(\mathrm{ind}_{l'})} \\ G(c :: l) & \xrightarrow[G(c ::_m f)]{} & G(c :: l') \end{array}$$

可交换。

这些数据定义了唯一的自然变换 $\phi : F \to G$，使得 $\phi_{[\,]} = \phi_{\mathrm{nil}}$，并且使得 $\phi_{c::l} = \phi_{c,\ l}\ (\phi_l)$。

我们使用一个专门的结构来实现上述引理的第一点，该结构封装了第一点中定义的数据，并产生一个声明，提供与这些数据相关联的函子。第二点则通过声明来实现，该声明直接将必要的数据作为参数。

备注 2.4。引理 2.3 的表述是双范畴意义下的，即它刻画了从对称列表范畴出发的函子，并且仅在唯一同构意义下确定这些函子。事实上，第一点中的数据唯一地定义了函子，而刻画所得函子的同构 υ_{nil} 和 υ_{cons} 可以（逐分量地）成为定义相等。我们有意避免采用这种方式表述，原因将在下文解释。

在 Lean 这类依赖类型论的环境中形式化范畴论时，一个反复出现的主题是：一般而言，应尽可能避免范畴对象之间的等式。其中一个原因是，态射的类型以及诸如态射复合之类的函数都直接依赖于对象。给定一个范畴 和项`x`、`y`、`z : C`，在上下文中存在等式 时，态射 无法直接作为态射 `y → z` 进行类型检查，而必须对 执行类型转换操作。当等式 是上下文中自由变量之间的等式时，执行类型转换很容易。如果 是一个更复杂的表达式，通常无法直接进行替换；尽管理论上可以通过对等式使用精心构造归纳谓词的归纳原理来执行替换，但在实践中这样做极其繁琐，尤其是因为每当这种情况出现时，都必须在每个证明中反复进行。此类情况在 Lean 社区中通常被戏称为“DTT Hell”。

为了让对象等式的情况稍微容易处理一些，`Mathlib` 选择只为恒等态射提供强制类型转换，这些转换称为：给定，就是等式 作为态射。将 用作校正项，以组合源对象和目标对象仅在命题相等意义下匹配的态射，是 `Mathlib`中的标准方法，但这只能部分缓解

处理对象等式时固有的困难；人们仍然认为，最好尽量避免一开始就陷入需要用到这些转换的情形。

当对象之间的等式 是 定义相等时，情况会好一些，类型检查器会接受 作为有效态射 `y → z`，但仍存在一个重要的技术细节：在 Lean 中，出于性能考虑，定义相等的检查取决于一个名为 透明度的设置，它控制在检查项的定义相等时允许展开哪些定义。Lean 中的大多数自动化策略（ ， ， `rw`等）都在“可约”透明度级别下运行，此时大多数定义都不会展开。如果在可约透明度下定义相等 不成立（例如，它被一个 所阻挡），策略可能无法使用依赖于它的项。过度依赖不可约的定义相等，通常被 Lean 社区俗称为“定义相等滥用”。

在我们的情形中，涉及 的定义相等可能成立，但在可约透明度下并不成立；依赖它们可能会在尝试自动化涉及这类递归定义函子的证明和计算时引发问题。因此，我们采取一种明确的立场，将这些定义相等视为实现细节，并在引理 2.3 的表述中体现这一点。

在实践中，在我们的实现里，我们利用 `v4.26.0` 中引入的 Lean 模块系统，使得 的主体不会被“公开”：这样可以确保“公共”作用域中的任何声明都无法使用支撑 v_{nil} 以及 v_{cons}各组成部分的定义相等。我们仍会在公共作用域中导出关于等式存在性的引理，因为它们可以方便地用于证明某些完全由 v_{nil} 和 v_{cons} 形式的态射构成的图表是交换的；但它们的定义性本质已被封装起来，从而不可能被滥用。

2.2 研究对称列表的态射

Mac Lane 最初证明相干定理的关键思想是：六边形关系和对称关系对应于给出 $n+1$ 字母上的对称群、使其成为 A_n型考克斯特群的关系 Σ_{n+1}：这一表示是同构

$$\left\langle (s_i)_{1\le i\le n} \;\middle|\; \begin{array}{rcll} s_i^2 & = & e & \\ s_i s_{i+1} s_i & = & s_{i+1} s_i s_{i+1} & \\ s_i s_j & = & s_j s_i & \text{if } |i-j|>1 \end{array} \right\rangle \cong \Sigma_{n+1}$$

它将生成元 s_i 映射为置换 $(i\ i+1)$（在枚举完字母集合之后）。我们确实可以看出，这个表示与我们的对称列表范畴之间有着明显的联系：给定一个对称列表 $[x_0,\dots,x_n]$，我们可以将 s_0 标记为如下形式的态射

$$\mathrm{sw}_{x,y,[x_2,\dots,x_n]} : [x,y,\dots,x_n] \to [y,x,\dots,x_n]$$

它恰好实现了列表指标上的置换 (0 1)，因此对称关系就可以写成 $s_0^2 = 1$。我们还可以将 s_1 标记为任意如下形式的态射： $x :: \mathrm{sw}_{y,z,[x_3,\ldots,x_n]}$，这样六边形关系就可以方便地写成 $s_0 s_1 s_0 = s_1 s_0 s_1$。最后，例如将 s_2 标记为如下形式的态射： $x :: y :: \mathrm{sw}_{z,t,[x_4,\ldots,x_n]}$，那么对称列表的态射中断言交换的自然性的关系便可写成等式 $s_0 s_2 = s_2 s_0$。

上文所述的过程为建立对称列表的态射与置换之间的对应关系提供了核心思路。不过，在证明助手能够接受之前，还需要进一步完善这一思路，使其足够形式化。

2.2.1 A_n 型考克斯特群与置换群

在尝试将上述过程形式化之前，必须先将 A_n 型考克斯特群与集合 $\{0, \ldots, n\}$ 上的置换之间的联系形式化。库Mathlib 将考克斯特系统定义为考克斯特系统：它是群与由考克斯特矩阵所确定的生成元与关系所表示的群之间的同构结构。对于 Mathlib，考克斯特矩阵是集合 M 上的一个（可能是无限的）方阵 I，其元素为正整数，且矩阵对称；所有对角线元素均为 1，所有非对角线元素均不等于 1。

集合 I 上的每个考克斯特矩阵 M 都通过将集合 I 上的自由群对 考克斯特关系 $\left((s_i s_j)^{M_{ij}} = e\right)_{i\,j \in I^2}$ 取商，定义一个群 M；其中 s_i 是与 $i \in I$ 对应的生成元。

在本文中，我们主要研究集合$\{0, \cdots, n-1\}$上的 Coxeter矩阵 A_n ，其表示为

$$\mathrm{A}_n = \begin{pmatrix} 1 & 3 & 2 & \cdots & 2 \\ 3 & 1 & 3 & \ddots & \vdots \\ 2 & 3 & 1 & \ddots & 2 \\ \vdots & \ddots & \ddots & \ddots & 3 \\ 2 & \cdots & 2 & 3 & 1 \end{pmatrix}$$

以及它的无限版本 A_∞ （作为自然数集 N 上的 Coxeter矩阵）。

截至撰写本文时，尽管已有关于抽象考克斯特系统的一些理论， Mathlib 仍无法生成任何非平凡的 类型项（此处的“非平凡”是指G 并非按定义等同于由 Coxeter矩阵给出的群）。我们遵循 [4, §1.5], 中的教科书证明，将一个（经典的）判据形式化，以确定群 G 中满足正确关系的一组二次生成元何时可扩展为 G 的考克斯特群表示：我们引入一种结构

```
/- Below, M.simple i is the element of the Coxeter group attached to M that
```

```
structure PreCoxeterSystem {B : Type*} (M : CoxeterMatrix B) (G : Type*) [Group G] where
  hom : M.Group →* G
  surjective_hom : Function.Surjective hom
  orderOf_eq (i j : B) : M i j = orderOf (hom (M.simple i) * hom (M.simple j))
  hom_simple_ne_one (i : B) :  hom (M.simple i) ≠ 1
```

将一个次数为 2 生成元集合在 G 中满足适当方程，视为从相应考克斯特群到 G的满射群同态。这个判据称为 交换性质 [4, 第 18], 页。下面的代码清单中，我们将用 Lean 代码对此进行说明：

```
variable {B : Type*} (M : CoxeterMatrix B) (G : Type*) [Group G] (S : PreCoxeterSystem M G)
/-- The length of an element `g` is the minimal length (in the sense of Coxeter systems)
of the preimages of `g` in `M.Group`. This corresponds to the minimal length of
a word of generators needed to express `g`. -/
noncomputable def length (g : G) : ℕ :=
  Nat.find ◁ show (M.toCoxeterSystem.length '' (S.hom⁻¹' {g})).Nonempty by …

local prefix:100 "ℓ " ⇒ S.length

/-- For ω a list of generators, π ω is the product in the Coxeter group attached to M
of the word ω. -/
local prefix:100 "π " ⇒ M.toCoxeterSystem.wordProd

/-- For ω a list of generators, φ ω is the product in G of the generators, via
the morphism S.hom :  -/
local notation "φ " x:max ⇒ S.hom (π x)

/-- A word is reduced if its length in the Coxeter group M.Group
(which is the minimal number of generators required to express it)
is equal to its length in G. -/
abbrev IsReduced (ω : List B) : Prop := ℓ (S.hom ◁ π ω ) = ω.length

/-- The "exchange property" for a pre-Coxeter system `S`: if a word of generators
`w = s₁⋯sₖ` in `G` is reduced and reduces further when multiplying on the left by a
generator `s`, then there exists `1 ≤ i ≤ k` such that
`sw = s₁⋯ŝᵢ⋯sₖ`. -/
def ExchangeProperty : Prop :=
  ∀ (ω : List B) (_ : S.IsReduced ω) (b : B) (_ : ℓ (φ (b :: ω)) ≤ ω.length),
    ∃ (i : ℕ) (_ : i < ω.length),  φ (b :: ω) = φ (ω.eraseIdx i)
```

按照 [4, 定理 1.5。1], 中的完全相同证明，我们可以 形式化 预Coxeter系统 满足交换性质时，其群同态 是单射。我们的形式化紧密遵循原文所述，更多细节请参阅我们的实现。

因此，这个判据适用于对称群：我们可以定义一个预Coxeter系统 ，定义在群 Σ_{n+1} 上，对应矩阵 A_n 的方法是将生成元映射到

i 置换 $(i\ i+1)$[3]。再次沿用 [4, 定理 1.51], 中的内容，我们可以形式化证明 对所有 n都满足交换性质。这需要将该预Coxeter系统中的长度概念与置换的逆序数联系起来。

最后，我们记录一个将在后续应用中用到的引理。

引理 2.5。设 M 是集合 I上的 *Coxeter* 矩阵，则群 M 也可由 *Coxeter* 关系表示为一个幺半群；也就是说，幺半群同态自由幺半群$(I) \to M$ 的核是自由幺半群(I) 上包含词 $\left((s_i s_j)^{M_{ij}}\right)_{i,j}$ 的最小乘法同余关系。

该引理的证明归结为以下事实：对于每个 i， $s_i^{-1} = s_i$ 都属于 M 。

2.2.2 对称列表的态射与置换

现在我们可以开始研究对称列表的态射了。首先需要注意的是，我们为态射指定标签的方式会忽略它们的源对象和目标对象。我们将范畴的态射由幺半群元素进行乘法标记这一想法抽象如下：

定义 2.6。设范畴 C 是一个范畴，且 M 是一个幺半群。范畴C 上的 M-权 是一个从C 到范畴 BM 的反范畴的函子，其中该范畴只有一个对象，并且幺半群 M 是这个对象的自同态。

这个定义在范畴中的态射与幺半群的元素之间建立了一种接口，使复合转化为乘法，并将恒等态射映射到幺半群的单位元。

本节开篇段落中描述的态射标记，应当被视为对称列表范畴上的一个取值于考克斯特群的权。事实上，由于我们想要将定义对称列表的关系与将置换群表示为考克斯特群的关系联系起来，因此先在自由对象的层面定义权会更方便。

命题-定义 2.7。由以下公式刻画的、在SListQuiv(C)的箭头上递归定义的赋值 w_0

$$w_0(\mathrm{sw}_{a,b,l}) = 0$$
$$w_0(x ::_m f) = 1 + w_0(f)$$

[3]在拉取请求#35218 中，独立于我们的工作，Kim Morrison 也定义了同一个态射，并证明了它的满射性。 Mathlib在[4]中，范畴中态射的复合顺序与纸笔数学中通常的写法相反，也就是说，它“沿着箭头进行”；这就是此处取反范畴的原因。 Mathlib

扩展为FreeSListQuiv(C) 上的权 w_0 ，其取值属于以$\mathbb{N}$为底集的自由幺半群。权 w_0 下降为SList(C) 上的权 w ，其取值属于考克斯特群A$_\infty$ ，并满足图表

$$\begin{array}{ccc} \mathrm{FreeSListQuiv}(C) & \xrightarrow{w_0} & (\mathrm{BFreeMonoid}(\mathbb{N}))^{\mathrm{op}} \\ {\scriptstyle \pi_C}\downarrow & & \downarrow \\ \mathrm{SList}(C) & \xrightarrow[w]{} & \left(\mathrm{B}\overline{\mathrm{A}_\infty}\right)^{\mathrm{op}} \end{array}$$

可交换。

直观地说，权 w_0 记录了与 FreeSListQuiv 中某个态射对应的交换序列发生作用的指标(C)，而权 w 则将对称列表的态射实现为A_∞中的一个元素。后者也可以看作一个置换群：考克斯特群 A$_\infty$ 存在一个到自然数集 $\mathbb{N}$ 上的置换群的群同态，将生成元 s_k 映射到置换 $(k\ k+1)$。利用对每个 n ，A Σ_{n+1} 与 n 的同一性，以及 A$_\infty$ 是群 A$_n$ 的滤过余极限这一事实，可以证明这个同态是单射。

我们可以通过这个权以及将 A$_\infty$ 中的元素解释为置换，证明对称列表的态射会以预期的方式置换源列表和目标列表中的元素

```
def toPerm : weight (SList C) (Equiv.Perm ℕ) := …

lemma SList.toPerm_app_lt_of_lt {L₁ L₂ : SList C} (f : L₁ → L₂) (k : ℕ) (hk : k < L₂.length) :
    (toPerm.app f) k < L₁.length := …

theorem SList.getElem_toList_toPerm {L₁ L₂ : SList C} (f : L₁ → L₂) (i : ℕ) (hi : i < L₂.length) :
    L₂.toList[i] = L₁.toList[toPerm.app f i]'(toPerm_app_lt_of_lt f i hi) := …
```

对范畴 FreeSListQuiv(C) 中的态射作归纳论证，还可以进一步证明w_0 是忠实的（ ），即作为函子，FreeSListQuiv(C)中标签相同的态射彼此相等。现在，主定理的一部分可以表述如下:

定理 2.8。函子 w 是忠实的，也就是说，两个对称列表的态射相等，当且仅当它们在范畴 C$_\infty$ 中具有相同的标签。

在我们的形式化中，这个定理称为 ，而这本身直接源自下方代码清单中的声明 ；该声明是将态射适当地提升到 后所得定理的一个版本。

```
/- Here, w₂ is the A∞-valued weight on FreeSListQuiv C induced by w₀
```

我们将证明 。

证明。在此证明中，我们将使用与上方 Lean 代码清单相同的记号：L 和 L' 是列表，视为对称列表的生成态射所构成的箭图上的自由范畴中的元素； f, g 是该范畴中的态射。令 toAinf 为典范投影 FreeMonoid(自然数集) $\to$A$_\infty$。首先需要稍作技术处理，将假设 转化为较小幺半群中的一个等式。这里我们用到引理 2.5 中的事实：考克斯特群作为幺半群的表示，与其作为群的表示具有相同的生成元与关系。回想一下， 是 。通过归纳法可知，底层对象列表的长度在 （因此也在 和 ）中的态射作用下保持不变，并且附加在长度为 n 的对称列表之间的态射上的形式字 $w_0(f)$ 不可能包含大于或等于 $n-1$的标签。因此，在我们的情形中，字 $w_0(f)$ 和$w_0(g)$ 可通过单态射提升

$$\iota_{n-1} : \mathrm{FreeMonoid}\,(\mathrm{Fin}(n-1)) \hookrightarrow \mathrm{FreeMonoid}\,(\mathbb{N})$$

由单射 Fin($n-1$) $\to$ N 诱导，其中 n 是源对象 L 底层列表的长度。此外还有一个交换图

$$\begin{array}{ccc} \mathrm{FreeMonoid}(\mathrm{Fin}(n)) & \xhookrightarrow{\iota_n} & \mathrm{FreeMonoid}(\mathbb{N}) \\ {\scriptstyle \mathrm{toA_n}}\big\downarrow & & \big\downarrow{\scriptstyle \mathrm{toAinf}} \\ \mathrm{A}_n & \longrightarrow & \mathrm{A}_\infty \end{array}$$

且底部的水平映射是单射，这可由 A$_n$ 与 Σ_{n+1} 之间的显式等价，以及 A$_\infty$ 是群 A$_n$ 的滤过余极限这一事实看出；该余极限与这些单射相容。

因此，等式 可以进一步细化为群 A$_{n-1}$ 的生成元所成字之间的等式。事实上，我们可以假设，在 FreeMonoid(Fin($n-1$)) 中提升 $w_0(f)$ 和 $w_0(g)$ 的字，分别是 FreeSListQuiv(C) 上某个权 w'_{n-1} 在 f 和 g 处的取值，其值落在 FreeMonoid(Fin($n-1$)) 中。可以通过将 Fin($n-1$) $\to$ N 的任意收缩扩展到自由幺半群上，由 w_0 构造满足此要求的权 w'_{n-1}。在我们的实现中，我们使用函数 $k \mapsto k$ %($n-1$) 作为这样的收缩，并将所得权命名为 。

因此，toAinf($w_0(f)$) = toAinf($w_0(g)$) 的等式等价于以下等式

$$\mathrm{toA_{n-1}}(w'_{n-1}(f)) = \mathrm{toA_{n-1}}(w'_{n-1}(g)) \tag{1}$$

我们可以利用商幺半群的刻画性质：这个等式意味着，$w'_{n-1}(f)$ 和 $w'_{n-1}(g)$ 是 Fin ($n-1$) 上自由幺半群中的字，并且它们在由定义考克斯特群 A$_{n-1}$ 的关系所生成的最小同余关系下相关。在 `Mathlib` 中，由一个关系生成的最小同余关系是以归纳方式定义的；我们回顾其定义以固定记号

```
inductive ConGen.Rel [Mul M] (r : M → M → Prop) : M → M → Prop| of : ∀ x y,
```

在我们的形式化中，对于考克斯特幺半群，施加在字上的关系同样是归纳生成的

```
variable {B : Type*} (M : CoxeterMatrix B) in
inductive CoxeterMatrix.monoidRelations : FreeMonoid B → FreeMonoid B → Prop
  | intro (i j : B) : monoidRelations ((.of i * .of j) ^ M i j) 1
```

证明通过对等式 (1) 进行归纳来进行，使用上文刻画 的递归子，并在此过程中对两个列表 L 和 L' 加以推广，以便在某些情形中可能出现的归纳假设能够应用于源和目标可能不同的态射。情形 和 没有技术上的困难。情形 和 稍为复杂，也更清楚地说明了我们为何决定先化归到群 A_{n-1}的情形：由于这里我们是对字而非态射进行归纳，这些情形中的归纳假设实际上会涉及自由幺半群 (Fin $(n-1)$) 中的额外字，而这些字先验地可能并不附着于 FreeSListQuiv (C) 中的任何态射。例如， 情形会给我们一个额外的字 z，它同时与 $\mathrm{toA}_{n-1}(w'_{n-1}(f))$ 和 $\mathrm{toA}_{n-1}(w'_{n-1}(g))$ 相关，而归纳假设只有在我们能找到一个实际的态射 h，其指向或来自 L，使得 $\mathrm{toA}_{n-1}(w'_{n-1}(h)) = z$ 时才能应用。幸运的是，情况确实如此，我们可以证明

```
-- Here, `w₀Fin n` is the weight called w'ₙ₋₁ in the text above.
```

注意，由上述方式构造的对象 j 实际上是唯一的，正如我们之前所观察到的（通过 ）：态射的目标由其源以及该态射的底层置换所限定。如果我们在 A_∞ 中工作，而不是在 A_{n-1}中，那么当词 z 包含大于或等于源和目标长度的字母时，f 与 g之间就不存在这样的态射。情形 类似，因为归纳假设中会出现额外的词；更多细节请参阅我们的形式化。

最后，情形 对所有可能的关系都遵循同一模式：在此情形下，toA 的词$_{n-1}$ $(w'_{n-1}(f))$ 和 $\mathrm{toA}_{n-1}(w'_{n-1}(g))$ 将是由生成元构成的显式短词，形式为 1，s_i^2、$(s_i s_{i+1})^3$ 或 $(s_i s_j)^2$。在这些情形中，我们可以从 L出发，使用构造子显式构造一个态射，使其在 w'_{n-1}下具有预期的标签。由于我们已经知道，在自由幺半群和 FreeSListQuiv(C)这一层面上，标签决定态射，因此这些手动构造的新态射

实际上必须等于 f 和 g，随后我们可以通过计算发现，这些第二态射作为对称列表的态射彼此相等。正如本小节开头一段所提示的，关系 $s_i^2 = 1$ 可化简为关系 `swap_swap`，关系 $(s_i s_{i+1})^3 = 1$ 本质上就是关系 ，而关系 $(s_i s_j)^2 = 1$ (针对$|i-j| > 1$) 对应于 。 □

由于 A_∞ 作为具有有限支撑的置换单射到 Perm (N)，我们可以将 SList(C) 中的态射完全解释为实际的置换。

```
def toEquiv {x y : SList C} (f : x → y) : Fin y.length ≃ Fin x.length where
```

该函数保持复合与逆。

受此构造启发，给定一个对称列表 L，我们将指标集（对应于 L）称为集合 $\{0, \ldots, \text{长度}(L) - 1\}$。在 Lean 一侧，我们将 称为指标类型 （对应于 ）。得益于定理 2.8，我们得到

```
theorem SList.getElem_toList_toEquiv {x y : SList C} (f : x → y) (i : Fin y.length) :
    y.toList[i] = x.toList[(toEquiv f) i] := …

theorem SList.hom_eq_iff_toEquiv_eq {x y : SList C} (f g : x → y) :
    f = g ↔ (toEquiv f) = (toEquiv g) where …
```

并且，我们可以通过归纳法证明以下内容，进一步巩固置换（或者更准确地说，指标集之间的双射）与对称列表之间的联系：

```
theorem SList.exists_lift_equiv {x y : SList C} (φ : Fin y.length ≃ Fin x.length)
```

最后这一批声明，连同前面列表中的 ，实际上提供了我们处理对称列表所需的全部 API。总而言之，这形式化了以下结果：

推论 2.9. 设 L_1 与 L_2 为 $\mathrm{SList}(C)$ 的对象。将态射$L_1 \to L_2$ 送到相应指标集之间对应双射的函数，实现了态射 $L_1 \to L_2$ 与双射之间的一个双射

$$\phi : \{0, \ldots, \mathrm{length}(L_2) - 1\} \xrightarrow{\sim} \{0, \ldots, \mathrm{length}(L_1) - 1\}$$

使得 $L_1[\phi(i)] = L_2[i]$。

综上，本小节中的定义构造了一个函子

$$\mathrm{I}_C : \mathrm{SList}\,(C) \to \mathrm{Core}\left(\mathrm{Fin}_{/C}\right)$$

它将对称列表 L 映射到其指标集合以及映射 $i \mapsto L[i]$，并将映射 f 映射到由 诱导的指标类型等价的逆映射。上述结果表明，此函子是全忠实的。利用有限集 J 等价于 $\mathrm{Fin}(k)$，其中 k 是J的基数，可以轻易证明其本质满。因此，函子 I_C 是范畴等价，不过目前我们还无法对这一等价的对称幺半结构作出任何论断。

2.3 从对称列表到相干定理

要证明一个与定理 1.3 有几分相似的陈述，剩下的步骤是将对称列表与集合上的自由幺半范畴联系起来。这是 S. Piceghello 的学位论文 [17], 中的部分内容；他在 HoTT 库中使用 Rocq 证明助手对其进行了形式化。

定理 2.10。 *[17, 推论 4.49]* 设 C 是一个集合，则定义在 C 上的对称列表范畴 $\mathrm{SList}(C)$ 与定义在 C上的自由对称幺半范畴等价。

通过这一等价，范畴 $\mathrm{SList}(C)$ 具有本质上唯一的对称幺半结构，使得定理 2.10 中的等价成为范畴的对称幺半等价。事实上，定义这一幺半结构本身就是证明该等价的一部分。我们基本上是在当前语境中翻译[17]中的相关构造。

自由对称幺半范畴 集合上的自由幺半范畴已在 `Mathlib` 中形式化为一个由生成元与关系表示的范畴。在这种表示中，对象由单位对象、来自基类型的对象以及二元张量积运算归纳生成。随后，态射由形式上的结合子、单位子和态射的张量积归纳生成。最后，加入关系以确保结合子和单位子具有自然性、张量积具有函子性，并满足三角恒等式和五边形恒等式。采用这种定义，所得对象是输入类型上的自由幺半范畴这一事实基本上是显然的。

在我们的形式化中，我们只需通过加入相关的额外生成元和关系，将这个定义调整为对称幺半情形：对象的类型仍以相同方式构造，并为编织同构加入生成元，同时为六边形关系和对称性恒等式加入额外关系。

与非对称情形一样，可以直接证明，这一构造具有预期的泛性质：从基类型到任意对称幺半范畴的每个函数都能提升为一个对称幺半函子，其定义域是我们构造的自由对称幺半范畴。类似地，从这个自由对称幺半范畴出发的对称幺半函子之间的幺半自然变换（或者更一般地，从对称幺半函子到松辫函子的幺半自然变换[5]）可以根据它们在来自基范畴的对象上的分量来定义，并且由这些分量唯一确定。

对称列表上的对称幺半结构证明定理2.10 的主要步骤之一，是在对称列表范畴上定义幺半结构。为了定义张量积双函子，需要将列表的拼接运算提升为 SList 上的双函子运算(C)。这通过引理2.3 实现，并借助我们在小节 中介绍的结构2.1来完成；该结构允许以归纳方式定义从对称列表出发的函子。

```
abbrev SList.appendRecData : RecursiveFunctorData C (SList C ⥤ SList C) where
  nilObj := 1 (SList C)
  consF c := Functor.whiskeringRight _ _ _▷.obj (c>~) -- c>~ is the functor c ::~ -
  swapIso x y := NatIso.ofComponents
    (fun F ↦
      (Functor.associator ..) ≪≫
        (Functor.isoWhiskerLeft F ◁ swapNatIso _ _) ≪≫
        (Functor.associator ..).symm)
    (fun {x y} f ↦ by ext; simp [swap_natural])
  …

def SList.appendFunctor : SList C ⥤ (SList C ⥤ SList C) := appendRecData.functor
```

请注意，这个定义是通常将两个列表接在一起的归纳定义的直接范畴化，这表明 是底层列表上的通常运算。

将此运算扩展到 SList 上一个（尚未对称的）幺结构中(C)很容易，因为此运算是严格结合且含单位元的[6]。其定义

[5]事实上，我们只需要源函子是余松编织的，也就是说，是一个与编织相容的余松幺半函子，但 Mathlib 目前尚无用于余松幺半函子编织结构的类型类。[6]这是少数几个会用到如下事实的地方之一：同构 υ_{nil} 和 υ_{cons} 引理 2.3 中出现的同构都由等式给出。利用这一点可以省略一些计算，这只是一个捷径，并非必不可少：我们也可以进行与 [17] 相同的计算，从而避免使用对象之间的任何等式。

编织的定义稍微复杂一些，我们完全沿用 [17,定义 425]中的构造。首先用归纳法定义一个同构

$$Q_{x,l_1,l_2} : (x :: l_1) \otimes l_2 \simeq l_1 \otimes (x :: l_2)\,,$$

关于 l_1，通过对 l_1 应用对称列表范畴中出发的自然变换的归纳原理（引理 2.3）。基础情形是同构

$$\begin{aligned} (x :: [\,]) \otimes l_2 &\rightharpoonup x :: ([\,] \otimes l_2) && \otimes\text{的刻画} \\ &\rightharpoonup x :: l_2 && \text{右单位元.} \end{aligned}$$

然后，给定某个同构 $(x :: l_1) \otimes l_2 \rightharpoonup l_1 \otimes (x :: l_2)$ 和 $y \in C$，可构造$Q_{x,\ y::l_1,\ l_2}$ 作为复合

$$\begin{aligned} (x :: y :: l_1) \otimes l_2 &\rightharpoonup x :: ((y :: l_1) \otimes l_2) && \otimes\text{的刻画} \\ &\rightharpoonup x :: y :: (l_1 \otimes l_2) && \otimes\text{的刻画} \\ &\rightharpoonup y :: x :: (l_1 \otimes l_2) && \text{使用 } \mathsf{sw}_{x\ \ y,l_1\otimes l_2} \\ &\rightharpoonup y :: ((x :: l_1) \otimes l_2) && \otimes \text{ 的刻画} \\ &\rightharpoonup y :: (l_1 \otimes (x :: l_2)) && \text{归纳法} \\ &\rightharpoonup (y :: l_1) \otimes (x :: l_2) && \otimes \text{ 的刻画。} \end{aligned}$$

接着，通过归纳法构造完整的编织，所依据的是 Q。基础情形同样很容易给出；假设某个同构 $l_1 \otimes l_2 \rightharpoonup l_2 \otimes l_1$ 已经构造出来，并且 $x \in C$，则构造一个同构 $(x :: l_1) \otimes l_2 \rightharpoonup l_2 \otimes (x :: l_1)$，其形式为复合

$$\begin{aligned} (x :: l_1) \otimes l_2 &\rightharpoonup x :: (l_1 \otimes l_2) && \otimes \text{ 的刻画} \\ &\rightharpoonup x :: (l_2 \otimes l_1) && \text{归纳法} \\ &\rightharpoonup (x :: l_2) \otimes l_1 && \otimes \text{ 的刻画} \\ &\rightharpoonup l_2 \otimes (x :: l_1) && \text{通过 } Q_{x,l_2,l_1}. \end{aligned}$$

包含函子与等价 根据一个类型上的自由对称幺半范畴的泛性质，上一段中的对称幺半结构诱导出一个对称幺半函子 N，它从 C 上的自由对称幺半范畴映到对称列表；在相差唯一幺半同构的意义下，该函子由如下事实刻画： $N(c)$ 同构于单元素对称列表 $[c]$。[7]

[7]事实上，我们对自由对称幺半范畴的实现可以确保同构 $N(c) \rightharpoonup [c]$ 是（定义上的）等式。但为了便于陈述，我们更倾向于尽可能避免

在另一个方向上，我们定义一个函子 J，从对称列表到 C 上的自由对称幺半范畴，并使用从对称列表出发的函子的归纳原理，即引理 2.3 以及我们先前引入的结构 。基础情形将空列表送到对称幺半结构的单位元，并将函子 $L \mapsto c :: L$ 送到函子 $x \mapsto c \otimes x$。通过直接计算（并多次使用 ，或者更确切地说是其自然同构变体 ），我们也可以在 J 上赋予幺半结构。与编织的相容性可以说是计算量最大的部分，因为它最终依赖于对称列表中编织关于上文所提及的部分辫结 Q 的递归定义。

这个单位同构同样通过归纳法构造，并利用函子 J 和 N的特征性质：空列表的情形很简单；假设同构 $N(J(L)) \nrightarrow L$ 已经构造好，那么它在形如 $c :: L$ 的对称列表上的分量由以下复合同构刻画

$$
\begin{aligned}
N\left(J\left(c :: L\right)\right) &\nrightarrow N\left(c \otimes J\left(L\right)\right) && \text{根据 } J \text{ 的刻画} \\
&\nrightarrow N\left(c\right) \otimes N\left(J\left(L\right)\right) && \text{根据 } N \text{ 的幺半性} \\
&\nrightarrow [c] \otimes N\left(J\left(L\right)\right) && \text{根据 } N \text{ 的刻画} \\
&\nrightarrow [c] \otimes L && \text{通过归纳法} \\
&\nrightarrow c :: L && \text{根据定义.}
\end{aligned}
$$

这一构造与交换相容这一事实，归结为直接计算。此外，还可以证明，以这种方式构造的单位同构是幺半的。

该等价的余单位同构很容易定义，因为它应当是从以 C 为基础的自由对称幺半范畴出发的对称幺半函子之间的幺半自然同构：这种自然同构由其在 C 的元素处的分量决定，也就是说，只需给出一族同构 $(J(N(c)) \nrightarrow c)_{c \in C}$，而这些同构可由 J 和 N 的定义给出。接着可以用一个小技巧证明，以这种方式构造的等价构成一个伴随等价：

引理2.11. 设 $F, G :$ 范畴 $\mathsf{C} \to \mathsf{D}$ 是幺半范畴之间的松幺半函子，并给定幺半同构 η: $F \circ G \simeq \mathrm{Id}$ 和 $\epsilon \colon G \circ F \simeq \mathrm{Id}$，使得$(F, G, \eta, \epsilon)$ 构成一个（不一定是伴随的）范畴等价。伴随化(F, G, η, ϵ') 的余单位(F, G, η, ϵ) 是幺半的。

尽可能使用一种不受范畴等价保持不变的语言。用同构来表述确实带来了一点额外的微妙之处：同构族 $(N(c) \nrightarrow [c])_{c \in C}$ 应被视为刻画对称幺半函子与幺半自然变换范畴中 N 的附加数据。此类同构族先验地可能具有非平凡的自同构，从而产生 N 的非平凡幺半自同构。在这个特定情形下，不存在这样的自同构，因为对象 $[c] \in \mathrm{SList}(C)$ 没有任何非平凡自同构，因此，仅凭这类同构的存在， N 就在唯一幺半同构的意义下被刻画。

证明。 伴随化过程可以在任意双范畴中进行；特别地，它可以在由幺半范畴、松幺半函子和幺半自然变换构成的双范畴中进行。

□

在 Lean 中，上述引理的证明本质上相同：幺半自然变换被表述为一个类型类 ，并为松幺半函子对幺半自然变换进行左侧和右侧粘合，以及结合子和单位子提供了该类型类的实例。伴随化过程只涉及此类构造，因此，伴随化保持幺半性的 Lean 证明完全由类型类推断系统自动完成。

对由 J 和 N构成的等价进行伴随化后，我们得到一个伴随等价，其余单位是幺半自然同构；只需证明这个新的余单位与我们上面构造的余单位相同。由于伴随等价的新余单位是幺半的，它由其在对象 $c \in C$处的分量 $J(N(c)) \nrightarrow c$唯一确定。仅利用 N 是全忠实的这一事实（这不需要任何伴随性），以及 $N(c) \simeq [c]$这一性质，我们就能证明这样的同构族必定唯一，因为形如 $[c]$ 的对称列表没有自同构。

2.3.1 对称列表和有限类型

为了引导自动化并简化与对称列表上的幺半结构相关的证明，定义代理等价很有用

```
def SList.Ψ (x y : SList C) : Fin x.length ⊕ Fin y.length ≃ Fin (x ⊗ y).length := …

def SList.Φ (x : C) (l : SList C) : Unit ⊕ Fin l.length ≃ Fin (x ::~ l).length := …
```

从底层来看，等价 是沿着等式的强制类型转换

$$\mathrm{length}(x \otimes y) = \mathrm{length}(x) + \mathrm{length}(y)$$

再接上等价（来自 `Mathlib`） ，；对于 也类似。通过定义抽象这些等价，而不是利用它们作为强制类型转换的底层本质，可以非常系统地证明从对称列表张量积出发的态射之间的等式。要证明等式 ，其中f和 是对称列表的态射，首先使用 ，然后调用 tactic，在上下文中引入一个项 ，接着利用 的满射性，得到一个项 ，并对其使用和类型的归纳原理，从而归结为证明以下两个等式

```
∀ i : Fin x.length, (toEquiv f) (Ψ x y (.inl i)) = (toEquiv g) (Ψ x y (.inl i))
```

and

```
∀ i : Fin y.length, (toEquiv f) (Ψ x y (.inr i)) = (toEquiv g) (Ψ x y (.inr i))
```

若能用 和 系统地刻画幺半结构的各种基本运算（例如态射的张量积），这种设计就能非常高效地自动化涉及对称列表上幺半结构的证明。例如，我们可以证明如下关键公式：

```
lemma whiskerRight_apply_right {x y : SList C} (f : x → y) (z : SList C)
    (i : Fin y.length) :
    toEquiv (f ▷ z) (Ψ y z ◁ .inl i) = Ψ _ _ (.inl (toEquiv f i)) := by …
```

并使用类似形状的引理标记属性，例如 可使自动化策略使用这些引理。所有这类公式都通过对对称列表应用归纳法来证明，使用对称列表张量积的归纳定义，以及 的定义性质：它由自然数集 上的置换诱导而来，该置换对应于态射的标签，并且在生成态射上的作用完全符合预期。

上面代码清单中的公式只是众多此类公式之一。这些公式表明，通过等价 ，由态射的张量积诱导的指标类型之间的等价会以预期的方式作用于各个因子；事实上，它还证明，通过对称列表的态射与其类型之间的等价之间的对应关系，我们用 Piceghello 的方法构造的编织确实对应于和类型因子的交换（即类型上的余笛卡尔幺半结构中的编织）。

定理 1.3 剩余部分由以下事实得出：从 C 上的对称列表到 C 上的有限类型群胚的函子是对称幺半函子。由于我们可以将等价 打包为一个位于 C之上的等价，等价 本身就是将对称列表映射到其指标类型的函子上的幺半结构同构。该等价的自然性及其与结合子、单位子和编织的相容性，正是我们上文提到的关于这些对象的特征引理的内容。因此，关于该等价Lean 形式化的事实

$$\mathrm{I}_C : \mathrm{SList}(C) \to \mathrm{Core}(\mathrm{Fin}_{/C})$$

是范畴的对称幺半等价这一事实，主要可以交由自动化策略来证明。至此，定理 1.3 的证明完成。

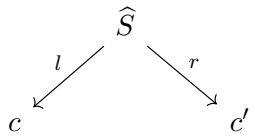
3 P来自跨度双范畴髓部的伪函子

现在，我们转向工作的第二个方面：运用相干定理，构造定理 1.1 中所述的无偏对称幺半范畴。

3.1 跨度的双范畴

定理 1.1 的陈述涉及跨度双范畴。我们简要回顾其定义，并说明如何实现它们，因为目前 `Mathlib` 中尚未提供这些定义。

给定范畴 C中的两个态射类 W_l 和 W_r，可以考虑左腿属于 W_l 且右腿属于 W_r的跨度：根据定义，这些跨度是 C中的图表



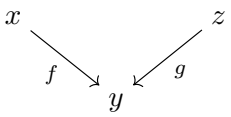
满足 $l \in W_l$ 和 $r \in W_r$。该定义可以形式化如下：

```
structure Span {C : Type*} [Category* C]
    (Wₗ : MorphismProperty C) (Wᵣ : MorphismProperty C)
    (c c' : C) where
  apex : C
  l : apex → c
  r : apex → c'
  wl : Wₗ l
  wr : Wᵣ r
```

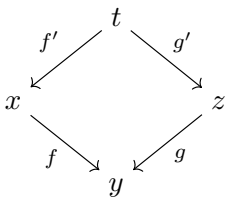
跨度具有范畴结构，其中态射 $S \to S'$ 是跨度顶点之间的态射，并且与投影相容。

如果态射类 W_l 和 W_r 满足以下条件，我们称它们是适当的：

- 它们都包含恒等态射，并且在复合下稳定。
- W_l 相对于 W_r有拉回，即对于每个图表

  使得 $f \in W_r$ 和 $g \in W_l$，存在 C中沿 f 对 g 的一个拉回。
- W_l 和 W_r 彼此在基变换下稳定，即如果

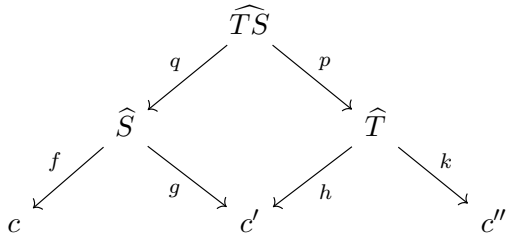
  

  ¡是 C 中的一个拉回方块，其中 $f \in W_r$ 和 $g \in W_l$，那么 $f' \in W_l$ 和 $g' \in W_r$。

当态射类 W_l 和 W_r 足够良好时，可以定义一个跨度双范畴：

定义 3.1. 设 C 是一个范畴，且 W_l, W_r 是态射的适当类。关于 W_l和 W_r的 C的跨度双范畴记作 $\mathrm{Span}(C, W_l, W_r)$，它是由以下数据定义的双范畴：

- 对象是 C的对象。
- 从 c 到 c' 的态射是跨度 $c \xleftarrow{f} \widehat{S} \xrightarrow{g} c'$。
- 2-态射是跨度之间的态射。
- c 上的恒等态射是跨度 $c \xleftarrow{\mathrm{Id}_c} c \xrightarrow{\mathrm{Id}_c} c$。
- 两个跨度 $c \xleftarrow{f} \widehat{S} \xrightarrow{g} c'$ 和 $c' \xleftarrow{h} \widehat{T} \xrightarrow{k} c''$ 的复合定义为以下图表中的总跨度



其中该方块是一个拉回方块。

- 水平复合由拉回的泛性质诱导。
- 结合子和单位同构定义为拉回之间的典范态射。

当 W_l 和 W_r 由所有态射构成时，充分性意味着 C 有拉回；此时，我们将相应的跨度双范畴记为 Span(C)。

最值得注意的一点是，在实现这一双范畴结构时存在一个特殊之处：该实现使用由选择公理提供的任意拉回来定义跨度的复合。为了使 Span 中复合所基于的实际拉回方块尽可能无关紧要，并减少从 Mathlib的 API（它通过选择公理提供拉回）可能泄漏的信息量，我们引入若干定义

```
variable {X Y Z : Spans C Wₗ Wᵣ} (S₁ : X → Y) (S₂ : Y → Z)
def Spans.πₗ : (S₁ ≫ S₂).apex → S₁.apex := Limits.pullback.fst _ _
def Spans.πᵣ : (S₁ ≫ S₂).apex → S₂.apex := Limits.pullback.snd _ _
```

```
def Spans.compPullbackCone : Limits.PullbackCone S₁.r S₂.l :=
  Limits.PullbackCone.mk (πₗ _ _) (πᵣ _ _) (comp_comm _ _)

def Spans.isLimitCompPullbackCone : Limits.IsLimit (compPullbackCone S₁ S₂) := …

def Spans.compLiftApex {c : C} (fₗ : c ⟶ S₁.apex) (fᵣ : c ⟶ S₂.apex) (hₘ : fₗ ≫ S₁.r = fᵣ ≫ S₂.l) :
    c ⟶ (S₁ ≫ S₂).apex :=
  Limits.PullbackCone.IsLimit.lift …
```

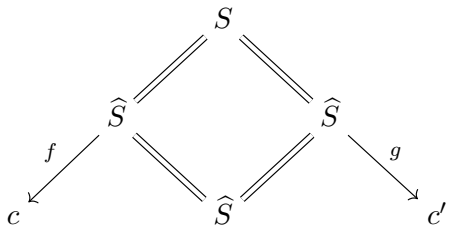
并将这些对象作为处理跨度复合的主要 API。例如，我们不使用 Mathlib现有的 ，而是使用 重新定义双范畴结构的结合子和单位子，并通过 和 来刻画它们，例如：

```
lemma associator_hom_hom_πₗ {W X Y Z : Spans C Wₗ Wᵣ} (S₁ : W ⟶ X) (S₂ : X ⟶ Y) (S₃ : Y ⟶ Z) :
    (α_ S₁ S₂ S₃).hom.hom ≫ πₗ .. = πₗ .. ≫ πₗ .. := …
```

这样的设计能尽量减少为改用打包的选定拉回族（而非选择公理所提供的拉回族）所需进行的重构。通过 和 系统地刻画结合子和单位子，也营造了一种有助于引导 和 等自动化策略的环境。

3.2 从跨度的髓构造伪函子

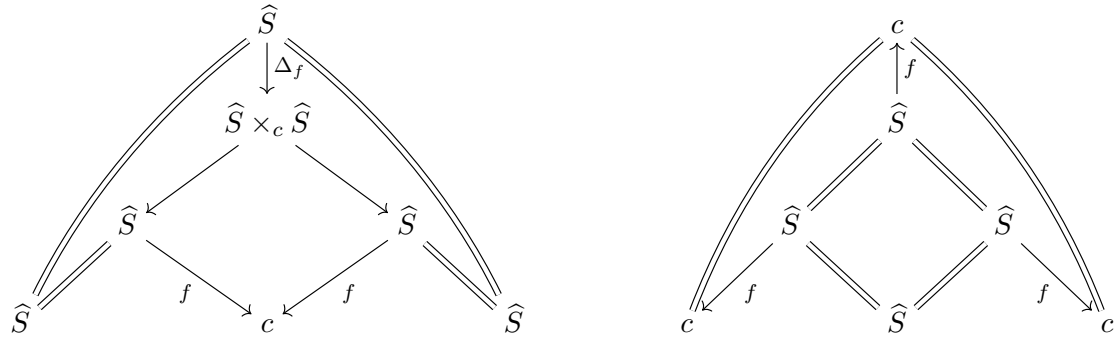
下文中，我们假设 C 是一个具有拉回的范畴，并假设跨度中的左类和右类都由所有态射构成（因此，充分性归结为 C 具有拉回这一事实）。给定 C 中的一个跨度 S：$c \xleftarrow{f} \widehat{S} \xrightarrow{g} c'$，图表



展示了 S同构于两个较小跨度的复合 f^*：$c \xleftarrow{f} \widehat{S}_{\mathrm{Id}} \xrightarrow{\widehat{S}} \widehat{S}$和$g_!$：$\widehat{S}_{\mathrm{Id}} \xleftarrow{\widehat{S}} \widehat{S} \xrightarrow{g} c'$。构造 $f \mapsto f^*$和 $f \mapsto f_!$都具有拟函子性，前者是逆变的，后者是协变的。

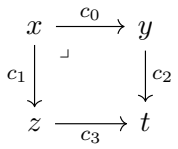
内部 地对应于跨度双范畴，其中 1-态射 $f_!$ 是该态射的左伴随 m

f^*：这一点可以通过考察图表看出

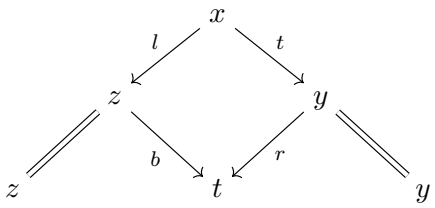


我们可以将其解释为定义伴随的单位元 $\mathrm{Id}_{\widehat{S}} \to f^* f_!$ 和余单位 $f_! f^* \to \mathrm{Id}_c$ 态射。这个伴随关于 f 也具有拟函子性，即该构造可提升为一个从 C 到 $\mathrm{Span}(C)$ 中某个适当的伴随双范畴的伪函子。

此外，给定一个拉回方块



在范畴 C 中，我们可以考察图表



作为 1-态射之间的一个同构

$$\left(x \xleftarrow{l} x \xrightarrow{t} y\right) \simeq (r^* b_!)$$

我们已经看到，第一个 1-态射也同构于 $t_! l^*$。因此，上面的拉回方块可以解释为一个同构

$$r^* b_! \simeq t_! l^*$$

看起来像许多“基变换公式”。[8] 尽管这些一般事实并非我们实现对称幺半范畴去偏这一目的所必需，但我们仍然实现了它们，以此验证跨度双范畴实现的稳健性。

[8] 在基变换公式的通常表述中，同构的一个方向是通过伴随演算得到的典范同构。在跨度这一情形下，我们可以验证，所构造的态射确实是这里所涉及的伴随和可伴随方块对应的那个态射。

由于双范畴内部的伴随可通过伪函子传递，所有这些关于伴随的约束以及基变换公式都必须在目标双范畴中成立。这对沿着来自基范畴的包含映射对伪函子的限制施加了相当强的约束；经典定理（例如参见[9, 定理 A2]）指出，这些约束实际上刻画了从跨度双范畴出发的伪函子。也就是说，给定一个双范畴 B，伪函子 $\mathbf{Span}(C) \to B$ 可以由两个伪函子的

$$U : C \to B$$
$$V : C^{\mathrm{op}} \to B$$

使得对于所有 x，$U(x) = V(x)$；使得在适当的伪函子意义下，U 是 V 的左伴随；并且使得由这些数据通过拉回方块诱导的基变换态射都是同构。

在定理 1.1的情形中，我们感兴趣的是定义从跨度双范畴的髓出发的伪函子，即仅保留跨度态射中顶点之间的态射为同构的那些跨度所构成的子双范畴。我们不能指望定理 1.1 中出现的伪函子是定义在整个跨度双范畴上的某个伪函子的限制，因为一般而言，定理中出现的函子 $f_!$和 f^* 并不互为伴随[9]。我们可以看到，先前描述的伴随 $f_! \dashv f^*$ 不能提升到髓中，除非态射 f 是同构。不过，拉回方块仍然定义了同构 $r^* b_! \simeq t_! l^*$；但这些同构不再能解释为通过双范畴髓内部的伴随演算得到的一般基变换态射。因此，要通过沿着 C的包含映射作限制来描述一个从 $\mathbf{Pith}\,(\mathrm{Span}(C))$ 出发的伪函子，就必须将拉回方块的基变换同构作为外部额外数据加入。因此，我们引入以下定义：

定义 3.2。设 C 是一个范畴，B 是一个双范畴。定义在 C 上的 B 值 髓-贝克-切瓦莱系统由以下数据组成：

- 一个函子 $U\colon\ C \to B$，以及 $V\colon\ C^{\mathrm{op}} \to B$，使得 U 和 V 在对象上相一致 $s.$
- 对于每个笛卡尔方块

$$\begin{array}{ccc} c_0 & \xrightarrow{t} & c_1 \\ {\scriptstyle l}\downarrow & \lrcorner\ S & \downarrow{\scriptstyle r} \\ c_2 & \xrightarrow[b]{} & c_3 \end{array}$$

在 C中，一个同构

$$\mathrm{BC(S)} : V(b)U(r) \nrightarrow U(l)V(t)$$

[9]例如，当 f 是唯一态射 $\{0,1\} \to \{*\}$时，函子 $f_!$ 在定理 1.1 中与张量积双函子相一致。一般而言，它作为双函子并不保持余极限（有时它分别在每个变量上保持余极限，但这并不相同），因此不能是左伴随。

在 B中。

数据必须满足以下条件：

- 同构 BC 与方块的水平粘贴相容：给定一个图表

$$\begin{array}{ccccc} a & \xrightarrow{t_0} & b & \xrightarrow{t_1} & c \\ {\scriptstyle v_0}\downarrow & \lrcorner\ L & \downarrow{\scriptstyle v_1} & \lrcorner\ R & \downarrow{\scriptstyle v_2} \\ d & \xrightarrow[b_0]{} & e & \xrightarrow[b_1]{} & f \end{array},$$

该图表

$$\begin{array}{ccccc} & & U(v_0)V(t_1t_0) & & \\ & {\scriptstyle \mathrm{BC}(\mathrm{hcomp}(L,R))}\nearrow & & \searrow{\scriptstyle U(v_0)\cdot V_{\mathrm{comp}}} & \\ V(b_1b_0)U(v_2) & & & & U(v_0)\,(V(t_0)V(t_1)) \\ {\scriptstyle V_{\mathrm{comp}}\cdot U(v_2)}\downarrow & & & & \uparrow{\scriptstyle \alpha} \\ (V(b_0)V(b_1))\,U(v_2) & & & & (U(v_0)V(t_0))\,V(t_1) \\ {\scriptstyle \alpha}\downarrow & & & & \uparrow{\scriptstyle \mathrm{BC}(L)\cdot V(t_1)} \\ V(b_0)\,(V(b_1)U(v_2)) & \xrightarrow[V(b_0)\cdot\mathrm{BC}(R)]{} & V(b_0)\,(U(v_1)V(t_1)) & \xrightarrow[\alpha^{-1}]{} & (V(b_0)U(v_1))\,V(t_1) \end{array}$$

是交换的。

- 同构 BC 与方块的垂直粘贴相容：给定一个图表

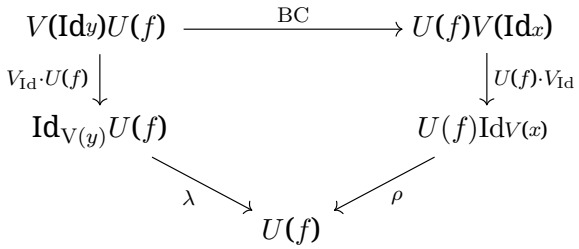
$$\begin{array}{ccc} a & \xrightarrow{h_0} & b \\ {\scriptstyle l_0}\downarrow & \lrcorner\ T & \downarrow{\scriptstyle r_0} \\ c & \xrightarrow[\lrcorner\ h_1]{} & d \\ {\scriptstyle l_1}\downarrow & B & \downarrow{\scriptstyle r_1} \\ e & \xrightarrow[h_2]{} & f \end{array},$$

图表

$$\begin{array}{ccccc} & & U(l_1l_0)V(h_0) & & \\ & {\scriptstyle \mathrm{BC}(\mathrm{vcomp}(T,B))}\nearrow & & \searrow{\scriptstyle U_{\mathrm{comp}}\cdot V(h_0)} & \\ V(h_2)U(r_1r_0) & & & & (U(l_1)U(l_0))\,V(h_0) \\ \downarrow{\scriptstyle V(h_2)\cdot U_{\mathrm{comp}}} & & & & \uparrow{\scriptstyle \alpha^{-1}} \\ V(h_2)\,(U(r_1)U(r_0)) & & & & U(l_1)\,(U(l_0)V(h_0)) \\ \downarrow{\scriptstyle \alpha^{-1}} & & & & \uparrow{\scriptstyle U(l_1)\cdot\mathrm{BC}(T)} \\ (V(h_2)U(r_1))\,U(r_0) & \xrightarrow{\mathrm{BC}(B)\cdot U(r_0)} & (U(l_1)V(h_1))\,U(r_0) & \xrightarrow{\alpha} & U(l_1)\,(V(h_1)U(r_0)) \end{array}$$

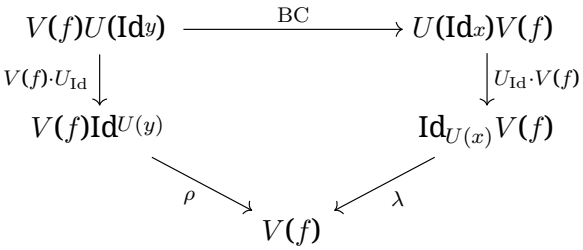
可交换。

- 同构 BC 保留水平单位方块：给定态射 $f\colon x \to y$在 C中，图表



交换。

- 同构 BC 尊重垂直单位方块：给定一个态射 $f\colon x \to y$ 在C中，该图表



交换。

我们将上述定义形式化如下。

```
structure PseudofunctorCore
    (C : Type u₁) [Category.{v₁} C] (B : Type u₂) [Bicategory.{w₁, v₂} B]
where
  obj : C → B
  u {x y : C} : (x ⟶ y) → (obj x ⟶ obj y)
  v {x y : C} : (x ⟶ y) → (obj y ⟶ obj x)
  uId' {x : C} (f : x ⟶ x) (hf : f = 𝟙 x := by cat_disch) : u f ≅ 𝟙 (obj x)
  vId' {x : C} (f : x ⟶ x) (hf : f = 𝟙 x := by cat_disch) : v f ≅ 𝟙 (obj x)
  uComp' {x y z : C} (f : x ⟶ y) (g : y ⟶ z) (h : x ⟶ z) (hfg : f ≫ g = h := by cat_disch) :
    u h ≅ u f ≫ u g
  vComp' {x y z : C} (f : x ⟶ y) (g : y ⟶ z) (h : x ⟶ z) (hfg : f ≫ g = h := by cat_disch) :
    v h ≅ v g ≫ v f
  -- pseudofunctoriality of u
  u_associator {c₀ c₁ c₂ c₃ : C} (f : c₀ ⟶ c₁) (g : c₁ ⟶ c₂) (h : c₂ ⟶ c₃) : …
  u_left_unitor {c₀ c₁ : C} (f : c₀ ⟶ c₁) : …
  u_right_unitor {c₀ c₁ : C} (f : c₀ ⟶ c₁) : …
  -- pseudofunctoriality of v
```

```
v_associator {c₀ c₁ c₂ c₃ : C} (f : c₁ ⟶ c₀) (g : c₂ ⟶ c₁) (h : c₃ ⟶ c₂) : …
v_left_unitor {c₀ c₁ : C} (f : c₁ ⟶ c₀) : …
v_right_unitor {c₀ c₁ : C} (f : c₁ ⟶ c₀) : …
-- base change isomorphisms and their compatibilities
baseChangeIso {c₀ c₁ c₂ c₃ : C} (t : c₀ ⟶ c₁) (l : c₀ ⟶ c₂) (r : c₁ ⟶ c₃) (b : c₂ ⟶ c₃)
  (S : IsPullback t l r b) :
  u r ≫ v b ≅ v t ≫ u l
baseChangeIso_unit_vert {x y : C} (f : x ⟶ y) :
  (baseChangeIso f (𝟙 x) (𝟙 y) f (IsPullback.of_vert_isIso .mk)).hom =
  (uId' (𝟙 y)).hom ▷ v f ≫ (λ_ _).hom ≫ (ρ_ _).inv ≫ v f ◁ (uId' (𝟙 x)).inv
baseChangeIso_unit_horiz {x y : C} (f : x ⟶ y) :
  (baseChangeIso (𝟙 x) f f (𝟙 y) (IsPullback.of_horiz_isIso .mk)).hom =
  u f ◁ (vId' (𝟙 y)).hom ≫ (ρ_ _).hom  ≫ (λ_ _).inv ≫ (vId' (𝟙 x)).inv ▷ u f
baseChangeIso_comp_horiz : …
baseChangeIso_comp_vert : …
```

我们可以观察到这一设计的若干技术要点：

1. 在注记 2.4 中，我们详述了处理范畴中对象的命题等式时所遇到的种种困难。这一讨论同样适用于双范畴中 1-态射的命题等式，因为它们是态射范畴的对象。不过，在我们的情形中，某些 1-态射的等式必然会出现：局部离散双范畴中的任何 2-态射都来自这样的等式。为避免表达式中大量出现 类型的态射，我们采用了`Mathlib` 中同样使用的技术来处理从局部离散双范畴出发的伪函子，即对 2-同构的返回类型给予定义控制：例如，这可见于字段 ，其中将复合 设为一个参数，并附带一个证明，即 。

2. 为了表述伪函子 U 与 V 在对象上一致这一事实，我们需要在该结构中展开它们的所有字段（ 、 等），从而使对象上的一致成为定义性的。试图通过目标双范畴中对象的命题等式来表述这种对象上的一致是不合理的，因为双范畴中对象的命题等式会进一步放大注记 2.4中所述的依赖类型问题。

> 另一种设计方案是将这部分结构封装为两个伪函子的数据，并附加一个关于它们的对象之间的等价族（在双范畴意义下）的额外数据。每当需要复合 1-态射时，都必须不断使用这些额外的等价来校正伪函子的源或目标，同时还需要进行额外的复合重新结合，这些新增的复杂性会使另一种可能的方案在实践中极难使用。

给定范畴 C 上的一个 Pith-Beck-Chevalley 系统 $\mathcal{S} := (U, V, \mathrm{BC})$，以及一个同构 ϕ：$x \to y$ in C，我们将用 $\eta_{\mathcal{S}}(\phi)$ 表示由以下复合得到的态射 $U(\phi)V(\phi) \to \mathrm{Id}_y$：

复合

$$U(\phi)V(\phi) \xrightarrow{\mathrm{BC}(S)^{-1}} V(\mathrm{Id})U(\mathrm{Id}) \xrightarrow{V_{\mathrm{Id}} \cdot U_{\mathrm{Id}}} \mathrm{Id} \circ \mathrm{Id} \xrightarrow{\lambda} \mathrm{Id}$$

其中 S 是笛卡尔方块

$$\begin{array}{ccc} x & \xrightarrow{\phi} & y \\ {\scriptstyle\phi}\downarrow & \lrcorner & \downarrow{\scriptstyle \mathrm{Id}\cdot} \\ y & \xrightarrow[\mathrm{Id}]{} & y \end{array}$$

我们形式化的主要结果是

定理—定义 3.3。设 C 是一个具有拉回的范畴， B 是一个双范畴，并设 $\mathcal{S} := (U, V, \mathrm{BC})$ 是定义在 C 上、取值于 B 的 ***Pith-Beck-Chevalley*** 系统。那么，存在一个伪函子

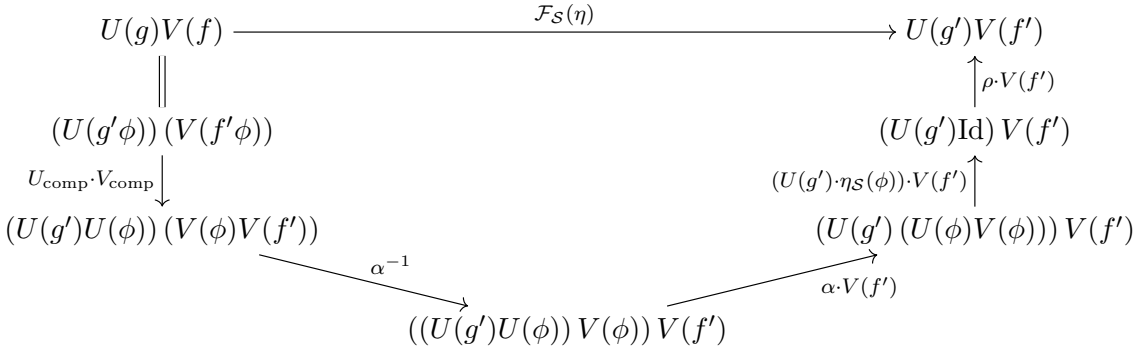
$$\mathcal{F}_{\mathcal{S}} : \mathrm{Pith}\,(\mathrm{Span}(C)) \to B$$

具有以下性质：

- 对于每个 $c \in C$， $\mathcal{F}_{\mathcal{S}}(c) = U(c)$。
- 给定 $c, c' \in C$ 和一个跨度 S： $c \xleftarrow{f} \widehat{S} \xrightarrow{g} c'$，属于 C，

$$\mathcal{F}_{\mathcal{S}}(S) = U(g)V(f).$$

- 给定 $c, c' \in C$，跨度 S： $c \xleftarrow{f} \widehat{S} \xrightarrow{g} c'$ 和 S'： $c \xleftarrow{f'} \widehat{S} \xrightarrow{g'} c'$，以及跨度 ϕ： $S \simeq S'$的态射 $\widehat{\ }$ $\widehat{\ }$，图表



可交换。

- 给定 $c, c', c'' \in C$，跨度 S： $c \xleftarrow{f} \widehat{S} \xrightarrow{g} c'$ 和 S'： $c' \xleftarrow{h} \widehat{S'} \xrightarrow{k} c''$ 在 C 中。将投影 $S'S \to S$（分别为 $S'S \to S'$）记作 $\widehat{S'S} \to \widehat{S}$（分别为 $\widehat{S'S} \to \widehat{S'}$），图表

$$\begin{array}{ccc}
U(k\pi_2)V(f\pi_1) & \xrightarrow{(\mathcal{F}_S)_{\mathrm{comp},S,S'}} & (U(k)V(h))\,(U(g)V(f)) \\
\downarrow{\scriptstyle U_{\mathrm{comp}}\cdot V_{\mathrm{comp}}} & & \uparrow{\scriptstyle \alpha} \\
(U(k)U(\pi_2))\,(V(\pi_1)V(f)) & & \\
\downarrow{\scriptstyle \alpha^{-1}} & & \\
((U(k)U(\pi_2))\,V(\pi_1))\,V(f) & & ((U(k)V(h))\,U(g))\,V(f) \\
\downarrow{\scriptstyle \alpha\cdot V(f)} & & \uparrow{\scriptstyle \alpha^{-1}\cdot V(f)} \\
(U(k)\,(U(\pi_2)V(\pi_1)))\,V(f) & \xrightarrow[(U(k)\cdot \mathrm{BC}(\mathrm{Cs})^{-1})\cdot V(f)]{} & (U(k)\,(V(h)U(g)))\,V(f)
\end{array}$$

可交换，其中 范畴 C 是笛卡尔方块

$$\begin{array}{ccc}
\widehat{S'S} & \xrightarrow{\pi_1} & \widehat{S} \\
{\scriptstyle \pi_2}\downarrow & \lrcorner & \downarrow{\scriptstyle g} \\
\widehat{S'} & \xrightarrow[g']{} & c'
\end{array}.$$

- 给定 $c \in C$，图表

$$\begin{array}{ccc}
U(\text{恒等}_c)V(\text{恒等}_c) & \xrightarrow{(\mathcal{F}_S)_{\mathrm{Id}\,c}} & \mathrm{Id} \\
& {\scriptstyle U_{\mathrm{Id}_c}\cdot V_{\mathrm{Id}_c}} \searrow \quad \nearrow {\scriptstyle \lambda} & \\
& \mathrm{Id}_c \circ \underset{\text{等}}{\text{恒}}\ c &
\end{array}$$

可交换。

请注意，定理中出现的伪函子必然是唯一的，因为伪函子的所有数据部分都已在此处得到完整描述。这些分量能够组装成伪函子的事实纯粹是命题性的，不包含任何额外数据。

定理 3.3 的形式化 是通过双范畴计算得到的。这些计算相当繁琐，因为图表涉及双范畴中的许多结构同构（结合子、单位子）。最繁琐的兼容性检查是跨度双范畴中复合与结合子的兼容性，这会产生许多充斥着双范畴符号的目标和表达式。 Mathlib的 ，结合 Mathlib的 策略 很有帮助，但还不足以让所有证明都能完全自动化。我们认为，开发专门的二维重写技术和策略，能够显著改善在 Mathlib 的双范畴框架中进行形式化的体验。有关计算细节，请参阅我们的形式化成果。

定理 3.3 肯定是众所周知的，但我们未能在双范畴文献中找到以这种方式精确陈述的版本（大多数相关内容都处理如何从完整的跨度双范畴定义伪函子，但未必是从其髓定义）。一个 $(\infty,1)$-范畴变体（以 $(\infty,1)$-跨度范畴的余笛卡尔纤维化来陈述）可见于 [14, 定理 04AE]。

4 将对称幺半范畴打包为伪函子

由于定理 (Span(Fin)) 已清楚地确定了从 Pith 3.3构造伪函子所需数据的形式，因此剩下的就是从对称幺半范畴 C 的数据中实际构造出这类数据，以证明定理 1.1。

4.1 对称列表的 Kleisli 双范畴

给定一个有限集 J 和一个对称幺半范畴 C，定理 1.1 中出现的积范畴 应解释为从 自由对称幺半范畴 J 到 C的对称幺半函子范畴；或者等价地，根据定理 2.10，解释为从 SList(J) $\to$ C 的对称幺半函子范畴。

设 J, K 为有限集，并设 g 为函数 $J \to K$。考察描述定理 $g_!$ 1.1 中出现的函子 C $^J \to$C^K 时，并牢记积范畴应解释为来自对称列表的对称幺半函子，合理的预期是，构造 $g \mapsto g_!$ 以及关于这类函子的公式，都是在 C 中的特例；这些公式在所有对称幺半范畴中普遍成立，也就是说，是关于对称列表的公式。

因此，为了形式化定理 1.1中的构造，我们需要能够将普遍层面上发生的事情分离出来，并抽象出这样一种思想：该构造仅在最后一步才将 范畴 C 中的普遍公式具体化。

对于普通代数理论，普遍公式由自由对象的子范畴编码。等价地，这类公式由相应单子的 Kleisli 范畴编码：根据定义，1-范畴 C 上单子 T 的 Kleisli 范畴具有与 C 相同的对象，而从 x 到 y 的态射就是范畴 C 中从 $x \to T$ 到 y 的态射。例如，交换环理论的一个普遍公式（其中 $T :=$ Z$[-]$）是具有整数系数的多项式族之间的一个等式。

对称幺半范畴无法直接纳入这一概念框架。这是因为，从本质上说，对称幺半范畴的结构是伪代数式的：结合律和幺元性只是在弱意义下成立，也就是说，它们是相对于指定的自然同构成立，而非相对于等式成立。将对称幺半范畴解释为某个伪代数 上的伪单子的伪代数要自然得多：这些概念是单子和代数这两个 1-范畴概念在双范畴中的类比。根据这种解释，

用于编码普遍公式的恰当对象应当是*Kleisli* 双范畴，即通常的 Kleisli 范畴在双范畴中的类比；它对应于对称幺半范畴理论所对应的伪单子。

这种解释最终证明颇有成效：构造 $I \mapsto \mathrm{SList}(I)$，连同其典范包含 $I \hookrightarrow \mathrm{SList}(I)$，以及对称列表的泛性质（它将任意函数 $J \to \mathrm{SList}(I)$ 延拓为对称幺半函子 $\mathrm{SList}(J) \to \mathrm{SList}(I)$），定义了一个相对伪单子 [8, 定义 3.1]，它位于包含函子 $\mathbf{Set} \to \mathbf{Cat}$[10]之上。相对伪单子的 Kleisli 双范畴也在 [8]中定义。由于陈述或形式化我们的结果并不需要关于相对伪单子的一般理论，相关概念的更多细节请读者参阅同上文献。

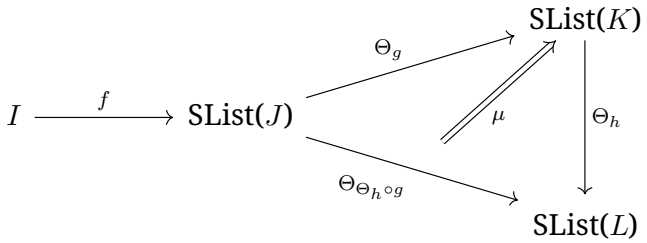
将对称幺半范畴解释为伪代数这一观点，引出了以下定义；它是 [8, 定理 4.1] 在我们情形下的特例：

命题—定义 4.1。对称列表的 Kleisli 双范畴 $\mathcal{K}_{\mathrm{SList}}$ 是由以下数据描述的双范畴：

- 对象是集合。
- 态射范畴 $I \rightsquigarrow J$ 是积范畴 $(\mathrm{SList}(J))^I$。
- 恒等态射 $I \rightsquigarrow I$ 由包含映射给出 $\iota_I : I \hookrightarrow \mathrm{SList}(I)$，它将元素 $i \in I$ 映射为单元素对称列表。
- $f : I \rightsquigarrow J$ 与 $g : J \rightsquigarrow K$ 的复合由如下复合给出，其中 Θ_g 是将 g 扩张为对称幺半函子的本质唯一延拓。
$$I \xrightarrow{f} \mathrm{SList}(J) \xrightarrow{\Theta_g} \mathrm{SList}(K)$$
- 给定 $f, f' : I \rightsquigarrow J$，$g, g' : J \rightsquigarrow K$，$\varphi : f \to f'$ 以及 $g \to g'$，φ 与 的水平复合是如下水平复合 $\Theta_\psi \cdot \varphi$，其中 Θ_ψ 是由 诱导的唯一幺半自然变换 $\Theta_g \to \Theta_{g'}$。
- 给定 $f : I \rightsquigarrow J$，右单位元 $f \circ \mathrm{Id}_I \nrightarrow f$ 是同构 $\Theta_f \circ \iota_I \nrightarrow f$，它刻画了 Θ_f 作为 f 在对称列表上的延拓。
- 给定 $f : I \rightsquigarrow J$，左单位子 $\mathrm{Id}_J \circ f \nrightarrow f$ 是由同构 $\Theta_{\iota_J} \simeq \mathrm{Id}_{\mathrm{SList}(J)}$ 所诱导的同构 $\Theta_{\iota_J} \circ f \nrightarrow f$，该同构由以下事实推出：两个函子均为幺半函子，且在限制到单元素对称列表时与恒等函子同构。

[10]在这里必须使用相对伪单子，在某种程度上是由我们的形式化设定人为造成的：将范畴C 映射到 C 上的自由对称幺半范畴这一操作，定义了 Cat 上一个真正的伪单子；但由于我们只对集合上的自由对称幺半范畴进行了形式化，也只在该设定下建立了它与对称列表之间的联系，因此我们只能使用相对于包含映射 Set → Cat 的伪单子。

- *Given* $f\colon\ I \rightsquigarrow J$，$g\colon\ J \rightsquigarrow K$，$h\colon\ K \rightsquigarrow L$，结合子由该图表描述



其中，μ 是唯一的幺半自然同构，其在单元素对称列表 $[j]$ 处的分量是同构

$$\Theta_{\Theta_h\circ g}([j]) \xrightarrow{\sim} \Theta_h(g(j)) \xrightarrow{\sim} \Theta_h(\Theta_g([j])).$$

证明命题 4.1 中的数据确实定义了一个双范畴结构，所需的所有证明都遵循完全相同的模式：所需的等式是对称列表态射族之间的等式，而这些态射族是从某个对称列表范畴出发的幺半函子之间的幺半自然变换的分量。因此，只需证明这些变换在由单元素对称列表组成的分量处相等。在这种情况下，等式归结为只涉及 Θ 的刻画同构的等式。图表中这些同构能够恰当地相互消去这一事实，属于 Lean 可通过 [redacted] 策略处理的记账工作。

在通常的 1-范畴论中，一个经典事实是，单子 T 的代数可解释为单子 $[7, \text{Th. } 1.2]$的 Kleisli 范畴上的适当预层。这种解释是通过将 T-代数范畴的 Yoneda 嵌入沿着从 Kleisli 范畴到 T-代数范畴的典范全忠实函子加以限制而得到的。

将对称幺半范畴解释为某个（相对）伪单子的伪代数，因此，可以合理地期待：对称幺半范畴可解释为从命题 4.1 的 Kleisli 双范畴的反范畴出发、取值于 Cat 的伪函子：

命题-定义4.2。设范畴 C 是一个对称幺半范畴。存在一个伪函子

$$\mathsf{C}_{\mathcal{K}} : (\mathcal{K}_{\mathrm{SList}})^{\mathrm{op}} \to \mathrm{Cat}$$

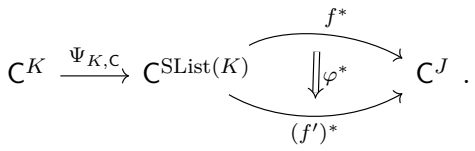
具有以下性质：

- 给定一个集合 J，范畴 $\mathsf{C}_{\mathcal{K}}(J)$ 是范畴 范畴 C^J。
- 给定一个态射 $f\colon\ J \rightsquigarrow K$ 在 $\mathcal{K}_{\mathrm{SList}}$ 中，函子 范畴 $\mathsf{C}_{\mathcal{K}}(f)$ 是复合

$$\mathsf{C}^K \xrightarrow{\Psi_{K,\mathsf{C}}} \mathsf{C}^{\mathrm{SList}(K)} \xrightarrow{f^*} \mathsf{C}^J$$

其中 $\Psi_{K,\mathsf{C}}$ 是将函数 $K \to$范畴 C 扩展为对称幺半函子 $\mathrm{SList}(K) \to$范畴 C的函子。

• 给定态射 $f, f' \colon J \rightsquigarrow K$（属于 $\mathcal{K}_{\mathrm{SList}}$）以及一个 2-态射 $\varphi \colon f \to f'$，范畴 $\mathsf{C}_{\mathcal{K}}$ 对 φ 的作用可由下图描述



• 给定集合 J，幺元同构

$$\mathsf{C}_{\mathcal{K}}(\mathrm{Id}_J) \overset{\sim}{\to} \mathrm{Id}_{C^J}$$

在 x： $J \to$范畴 C 处的分量是同构

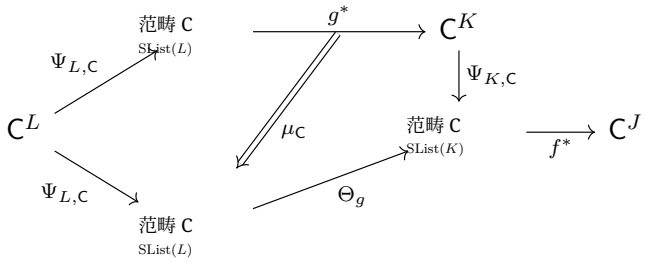
$$\iota_J^* \left(\Psi_{J,\mathsf{C}}(x) \right) \overset{\mathrm{def}}{=} \Theta_x \circ \iota_J \overset{\sim}{\to} x$$

它是将 x 沿着 ι_J 延拓的唯一对称幺半函子 Θ_x 的定义性刻画的一部分。

• 给定 在 $\mathcal{K}_{\mathrm{SList}}$ 中的态射 f： $J \rightsquigarrow K$ 和 g： $K \rightsquigarrow L$，复合同构

$$\mathsf{C}_{\mathcal{K}}(gf) \overset{\sim}{\to} \mathsf{C}_{\mathcal{K}}(g) \circ \mathsf{C}_{\mathcal{K}}(f)$$

由图表描述



其中， μ_{C} 是一个自然同构，其在 x： $L \to$范畴 C 处的分量是唯一的幺半同构

$$\Psi_{K,\mathsf{C}} \left(g^* \left(\Psi_{L,\mathsf{C}}(x) \right) \right) \overset{\mathrm{def}}{=} \Theta_{\Theta_x \circ g} \overset{\sim}{\to} \Theta_g \circ \Theta_x \overset{\mathrm{def}}{=} \Theta_g \circ \Psi_{L,\mathsf{C}}(x)$$

由两侧的对称幺半群结构以及一族同构所诱导

$$\left(\Theta_{\Theta_x \circ g}([l]) \overset{\sim}{\to} \Theta_x(g(l)) \overset{\sim}{\to} \Theta_x(\Theta_g([l])) \right)_{l \in L}$$

源于 Θ_h 作为从对称列表出发、延伸给定函数 h 的本质唯一幺半函子的刻画性质。

此外，对于 $f:\ J \rightsquigarrow K$ 属于 $\mathcal{K}_{\mathrm{SList}}$ 的每一种情形，函子 $\mathsf{C}_{\mathcal{K}}(f)$ 都带有对称幺半函子结构，该结构由 $\Psi_{K,\ \mathsf{C}}$ 上的对称幺半结构与 f^* 上的对称幺半结构复合而诱导。*2*-态射上的作用，以及复合和幺元性的同构，关于这一结构都是幺半的。

注4.3。命题 4.2 中所述的伪函子有一个特点：它完全依赖于对称列表作为自由对称幺半范畴的泛性质。我们可以通过用自由对称幺半范畴的其他模型（例如，我们在小节 $\mathcal{K}_{\mathrm{SList}}$ 中使用的"平凡"模型，它是证明对称列表构成自由对称幺半范畴的一部分）替换对称列表，来定义对应的双范畴版本，而类似的构造也是可行的。2.3

这可以由以下事实解释：命题 4.2 中所述的伪函子显然是对称幺半范畴、对称幺半函子和幺半自然变换所构成的双范畴 SymmMonCat 的双范畴 Yoneda 嵌入，经由一个伪函子 $\mathcal{K}_{\mathrm{SList}} \to$SymmMonCat 所得的限制；该伪函子将 J 映到范畴 SList(J)。如果我们愿意相信，SymmMonCat 与对称列表的相对伪单子（按 [1]的意义）所对应的伪代数双范畴双等价，那么上面定义中的伪函子恰好就是沿着从 Kleisli 双范畴出发的典范全忠实函子对（双范畴）Yoneda 嵌入所作的限制。关于伪单子的这些一般性结果，或许有助于从概念上理解命题 4.2 中的构造，并看出它如何平行于 [7, 定理 1。2], 所述的 1-范畴现象，但对我们的论述并非必需，因此我们不对其进行形式化。

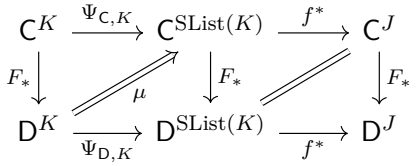
命题 4.2 的形式化主要包括：在幺半扩张函子 $\Psi_{K,C}$: 范畴 $\mathsf{C}^K \to$范畴 $\mathsf{C}^{\mathrm{SList}(K)}$ 上构造对称幺半结构，并证明其中出现的所有 2-态射（包括对 2-态射的作用，以及幺元性和复合同构）相对于该结构都是幺半的。扩张函子上的幺半结构，本质上源于这样一个事实：在对称幺半范畴 C 中，张量积双函子本身就是对称幺半的。得到这一幺半结构后，便可利用对称列表作为自由对称幺半范畴的泛性质，构造命题中出现的所有 2-同构。

其余证明义务（与结合子和单位子的相容性）都采用与命题 4.1中相同的方式进行形式化：展开定义并逐分量处理后，出现的每个公式都是某个幺半自然变换的分量等式；该自然变换连接从对称列表范畴到 范畴 C的幺半函子。同样，只要将问题化归到单元素对称列表，证明便可归结为整理细节（以及 Kleisli 双范畴的结合子和单位子的刻画性质），而 之类的自动化策略即可完成这些证明。

正如将其解释为一种合适的 Yoneda嵌入时所预期的那样，将对称幺半范畴编码为从 Kleisli

双范畴的反范畴出发的伪函子，与对称幺半范畴之间的态射相容：

引理4.4。 设范畴 C 和D 为对称幺半范畴，且 $F\colon\ \mathsf{C}\to\mathsf{D}$ 为松辫函子（即保持编织的松幺半函子）。对 F 进行后复合可扩展为伪函子$\mathsf{C}_{\mathcal{K}}\to\mathsf{D}_{\mathcal{K}}$之间的松自然变换。对于态射 $f\colon J\rightsquigarrow K$，其自然性方块由下图给出，



其中，μ 在 $X\colon\ K\to\mathsf{C}$ 处的分量是唯一的幺半自然变换$\Theta_{F_*(X)}\ \to\ F_*(\Theta_X)$，满足其在单元素对称列表 $[k]$ 处的分量为复合

$$\Theta_{F_*(X)}([k])\xrightarrow{\mathrm{def}}F(X(k))\xrightarrow{F(\mathrm{def})}F(\Theta_X([k]))$$

如果 F 是（强）幺半的，则 μ 是自然同构，因此该自然变换是伪函子之间的强自然变换。

我们在 形式化中实现了上述引理，但遗憾的是目前无法进一步证明，也尚未能形式化以下事实：松辫子幺半函子的幺半自然变换会诱导出相应松自然变换之间的修正：在撰写本文时，中目前仅定义了反松自然变换之间或强自然变换之间的修正。

作为伪函子 范畴 $\mathsf{C}_{\mathcal{K}}$ 能够完整编码在给定的对称幺半范畴中实例化对称列表普遍公式的过程 范畴 C，我们现在可以精确定义（从而使其可形式化）定理 1.1中所涉及的公式全都是普遍公式这一观点：我们试图构造的伪函子

$$\mathsf{C}^{\otimes}:\mathrm{Pith}\left(\mathrm{Span}\left(\mathrm{Fin}\right)\right)\to\mathrm{Cat}$$

w实际上是范畴 $\mathsf{C}_{\mathcal{K}}$ 与一个伪函子的复合 or

$$\Lambda:\mathrm{Pith}\left(\mathrm{Span}\left(\mathrm{Fin}\right)\right)\to\mathcal{K}^{\mathrm{op}}_{\mathrm{SList}}.$$

因此，构造这样的伪函子 Λ 就是我们必须实现的新主要目标，以便完成定理 1.1的证明。

4.2 对称列表的对偶

作为构建满足我们要求的伪函子 Λ 的重要一步，我们还需要再抽象出若干中间构造。正如我们从

非正式描述（关于范畴 $\mathbf{C}^{\otimes}$，见定理1.1）中可以看到，伪函子的协变部分涉及取由函数在一点处的纤维所索引的族的张量积。因此，在转向对称列表之前，先在位于给定集合之上的有限集的群胚这一层面抽象出相应运算是合理的。为此，我们抽象出以下

命题-定义 4.5. 设 J 是一个集合，且设 $j \in J$。将配备函数 $f\colon X \to J$ 的集合 X 映至 $f^{-1}(\{j\})$，并将 J 上有限集之间的双射映至其在 j 处纤维上的限制，这一构造扩展为对称幺半函子 $\mathrm{fib}^j \colon \mathrm{Core}\left(\mathrm{Fin}_{/J}\right) \to \mathrm{Core}(\mathrm{Fin})$.

在族的层面上，这一操作实现了底集合上的集合与集合族之间的经典对应：

引理 4.6. 设 J 是一个有限集，对称幺半函子

$$\mathrm{Fib}_J : \mathrm{Core}\left(\mathrm{Fin}_{/J}\right) \to \mathrm{Core}(\mathrm{Fin})^J$$

由对称幺半函子族 $(\mathrm{fib}^j)_{j\in J}$ 诱导的函子扩展为范畴的对称幺半等价，其中逆函子将集合族 $(X_j)_{j\in J}$ 映至它们的不交并，并配备投影之和。

最后两个构造的形式化相当直接。在我们的实现中，我们使用 [redacted] 而不是 $\mathbf{Core}(\mathrm{Fin})$，但这种差异纯属表面上的，因为相干定理表明它们作为对称幺半范畴是等价的；我们也将这一事实纳入了形式化中。

借助范畴的对称幺半等价 $\mathbf{I}_C$，上一个引理中的对称幺半函子揭示了一个关于对称列表的有趣构造：

定义 4.7。设 J 和 K 为有限集。我们定义对偶函子

$$\mathcal{D}_{J,K} : (\mathrm{SList}(K))^J \to (\mathrm{SList}(J))^K$$

为如下复合

$$\begin{aligned}
(\mathrm{SList}(K))^J &\xrightarrow{\sim} \left(\mathrm{Core}\left(\mathrm{Fin}_{/K}\right)\right)^J \\
&\xrightarrow{\sim} \left(\mathrm{Core}\left(\mathrm{Fin}\right)^K\right)^J \\
&\xrightarrow{\sim} \left(\mathrm{Core}\left(\mathrm{Fin}\right)^J\right)^K && \text{通过交换变量} \\
&\xrightarrow{\sim} \left(\mathrm{Core}\left(\mathrm{Fin}_{/J}\right)\right)^K \\
&\xrightarrow{\sim} (\mathrm{SList}(J))^K .
\end{aligned}$$

此函子是对称列表的一种对偶：复合等价

$$\mathrm{SList}(K)^J \rightleftarrows \left(\mathrm{Core}\,(\mathrm{Fin})^K\right)^J$$

这使我们能够将取值于 K 的对称列表向量（至少从直观上）视为有限集矩阵，而函子 $\mathcal{D}_{J,\ K}$ 在这种解释下本质上就是转置操作。注意，作为范畴对称幺半等价的复合，$\mathcal{D}_{J,\ K}$ 是范畴对称幺半等价。

4.3 对称列表与多重集

由于我们不应对经过 I_C 的逆映射的对称列表中元素的顺序作任何假设，因此用多重集的语言来表述 $\mathcal{D}_{J,\ K}$ 的某些性质是很自然的。

回顾一下，元素取自 的多重集 是指 K 的一个列表模去置换关系，即一个带重数的有限集。在 `Mathlib`中，给定一个类型 `α`, 类型 定义为类型 关于关系 的商类型。该关系在 Lean 的核心库中归纳定义，用来表示一个列表是另一个列表的置换。定义 非常直观地表明，对称列表是多重集的一种范畴化：更准确地说，元素取自 K 的对称列表的同构类集合，与元素取自 K的多重集之间存在双射，因此我们可以谈论对称列表的底层多重集。多重集构成一个交换幺半群，其加法由列表的拼接诱导，因此不难看出，对称列表张量积的底层多重集，等于各因子底层多重集之和。

我们将对多重集使用类似集合的记号，同时牢记其中允许有重数（因此， $\{i,i\}$ 是一个与 $\{i\}$不同的多重集）。给定由K和 $k \in K$的元素构成的多重集 M ，我们用 $\mathfrak{c}(k,M)$表示 k在 M中的重数。给定某个集合中的对称列表 L ，我们用 $||L||$ 表示其底层多重集。

继续类比范畴 C中的1-态射与有限集构成的矩阵，鉴于有限集的同构类可由自然数描述，我们可以预期，范畴 C中1-态射各分量的底层多重集表现得如同自然数矩阵；事实上，复合所得多重集有一个类似矩阵的公式：

引理 4.8。 设 J、 K 和 L 是视为范畴 C对象的集合，设 $f\colon\ J \rightsquigarrow K$、 $g\colon\ K \rightsquigarrow L$是范畴 C中的*1*-态射，并设 $j \in J$，那么如果 K 是有限的，

$$\|(g \circ f)(j)\| = \sum_{k \in K} \mathfrak{c}\left(k, \|f(j)\|\right) \cdot \|g(k)\|.$$

证明。 通过检查 Kleisli 双范畴中复合的定义，可以看出，只需证明对于每个函数 $f\colon\ K \to$ SList(L) 以及每个对称列表

$X \in$SList(K)，等式

$$||\Theta_f(X)|| = \sum_{k \in K} \mathfrak{c}\,(k, ||X||) \cdot ||f(k)||$$

成立。这个公式将对称列表的张量积转化为多重集的加法，因此只需证明当 X 是一个单元素对称列表 $[k]$ 时成立。在这种情况下，公式显然成立，因为根据 Θ 的定义， $\Theta_f([k])$ 和 $f(k)$ 同构，因此具有相同的底层多重集。

□

类似地，对偶函子 $\mathcal{D}_{J,\ K}$ 所对应的多重集可以用矩阵式公式计算，从而进一步印证了将 $\mathcal{D}_{J,\ K}$ 解释为转置算子的观点。

引理 4.9。设 J 和 K 是有限集，并设 $(X_j)^{j \in J}$ 是一个由 K 中元素的对称列表组成的族。设 $k \in K$，则

$$\|\mathcal{D}_{J,K}(X)(k)\| = \sum_{j \in J} \mathfrak{c}(k, X(j)) \cdot \{j\}.$$

证明。 两个多重集相等，当且仅当它们在每个元素处的重数都相同，因此该公式归结为证明

$$\mathfrak{c}(j, \|\mathcal{D}_{J,K}(X)(k)\|) = \mathfrak{c}(k, \|X(j)\|)$$

对每个 $j \in J$ 和每个 $k \in K$ 都成立。展开 $\mathcal{D}_{J,\ K}$ 的定义后，只需注意到，对于 J 上的每个有限集 S， $\mathfrak{c}(j, \mathrm{I}_J^{-1}(\mathrm{Fib}_J^{-1}(S)))$ 等于 S 中 j 的原像的基数；这一点可由定义直接得出。

□

由于多重集恰好是对称列表的同构类，因此证明底层多重集相等，是断言存在某个 对称列表之间同构的一种方便的计算工具。我们已经看到，在某些情况下，可能存在多个彼此不相等的此类同构：重复同一元素 n 次的对称列表的自同构集合是对称群 Σ_n，这可由推论 2.9得出。正如下文将要解释并形式化的，列表中存在重复元素，实际上是两个对称列表之间存在多个同构的唯一可能原因。

定义 4.10。设 J 是一个集合， L 是一个对称列表。若 L 的底层列表没有重复项，则称 L 是线性的 。

等价地，对称列表 L 是线性的，当且仅当其底层多重集中的所有元素的重数都至多为一。相应地，在我们的形式化中，我们引入

```
class Linear (L : SList C) where
  nodup : L.toList.Nodup
```

这一概念还原了对称幺半范畴相干定理的另一种经典形式：

引理 4.11. 设 L_1 和 L_2 是取值于集合 J 的两个对称列表。

1. 若存在态射 $L_1 \to L_2$，则 L_2 是线性的，当且仅当 L_1 是线性的。*2.* 若 L_1 或 L_2 中至少一个是线性的，则从 L_1 到 L_2 至多存在一个态射。

证明。 两个对称列表之间存在态射，当且仅当它们对应的列表互为置换。因此，第一点可由以下事实得出：一个列表没有重复元素，当且仅当它是一个没有重复元素的列表的置换。

对于第二点，假设其中一个列表是线性的。根据推论 2.9 从 L_1 到 L_2 的态射集合与双射集合一一对应

$$\phi : \{0, \dots, \mathrm{length}(L_2) - 1\} \xrightarrow{\sim} \{0, \dots, \mathrm{length}(L_1) - 1\}$$

使得 $L_2[\phi(i)] = L_1[i]$。由于两个列表都没有重复元素（由第一点可知），若存在这样的双射 ϕ，则 $\phi(i)$ 的值必定是 L_1 中取值为 $L_2[\phi(i)]$ 的唯一索引。因此，这样的双射至多存在一个。 □

在 Lean 中，上述引理第二个要点的结论，自然可以用证明助手核心库中的 [redacted] 类来表达，该类表示某个类型至多只有一个元素。因此，引理4.11 的形式化就是注册一系列 [redacted] 实例：

```
lemma linear_iff_of_iso {x y : SList C} (e : x ≅ y) : Linear x ↔ Linear y := by …
instance subsingleton_hom_of_linear_left (L L' : SList C) [hₒ : Linear L] :
    Subsingleton (L → L') where …
instance subsingleton_iso_of_linear_right (L L' : SList C) [Linear L'] :
    Subsingleton (L ≅ L') where …
instance subsingleton_hom_of_linear_right (L L' : SList C) [Linear L'] :
    Subsingleton (L → L') where …
instance subsingleton_iso_of_linear_left (L L' : SList C) [Linear L'] :
    Subsingleton (L' ≅ L) where …
```

借助这套基础设施，我们便可以在证明中使用 [redacted] 策略 `Mathlib` 。它会将证明目标识别为带有 [redacted] 实例的类型中两个项之间的等式；如果 Lean 能推断出该实例，就会关闭这个目标。

4.4 构造 Λ

在小节 4.2中，我们为对称列表向量构造了一个对偶，它们是 $\mathcal{K}_{\mathrm{SList}}$ 中的 1-态射。具有拉回的范畴 C 上的跨度也有一个典范对偶：将跨度 $S : c \xleftarrow{f} \widehat{S} \xrightarrow{g} c'$ 映射为其转置 ${}^tS : c' \xleftarrow{g} \widehat{S} \xrightarrow{f} c$ 的运算。给定 $f : c \to c'$，这个对偶会交换跨度 f^* 和 $f_!$。如果愿意相信我们正试图构造的伪函子 $\Lambda : \mathrm{Pith}\ (\mathrm{Span}\ (\mathrm{Fin})) \to \mathcal{K}^{\mathrm{op}}{}_{\mathrm{SList}}$ 保持其源和目标上的对偶，那么定义便只有一种可能：

定理—定义 4.12。存在唯一一个取值于 $\mathcal{K}^{\mathrm{op}}{}_{\mathrm{SList}}$ 的 *Pith-Beck-Chevalley* 系统，它定义在 Fin 上并满足以下性质：

- 伪函子 $V\colon \mathrm{Fin}^{\mathrm{op}} \to \mathcal{K}^{\mathrm{op}}{}_{\mathrm{SList}}$ 将集合 J 映到自身，并将函数 $f\colon J \to K$ 映到复合

$$J \xrightarrow{f} K \xrightarrow{\iota_K} \mathrm{SList}(K).$$

- 伪函子 $U\colon \mathrm{Fin} \to \mathcal{K}^{\mathrm{op}}{}_{\mathrm{SList}}$ 将集合 J 映到自身，并将 $f\colon J \to K$ 映到 $\mathcal{D}_{J,\,K}(V(f))$。

上述定理中的描述可能先验地显得不完整，因为它没有阐明拟函子性数据，即 V 与 U，也没有描述所需的基变换同构。事实上，如下文将证明的，所有这些同构都由定理 4.12中规定的取值唯一确定。我们将使用4.3 中引入的线性对称列表概念以及引理 4.11 来使这一点显而易见。

证明。注意到，给定函数 $f\colon J \to K$，其复合

$$J \xrightarrow{f} K \xrightarrow{\iota_K} \mathrm{SList}(K)$$

是函数 $V(f)$，它将 j映到单元素对称列表 $[f(j)],$，该列表是线性的。这一描述给出了唯一的幺元同构：映射 $\iota_J\colon j \mapsto [j]$ 是 $\mathcal{K}_{\mathrm{SList}}$ 中 J 上的恒等态射，因此 $V(\mathrm{Id}_J)$ 依定义显然同构于恒等态射，并且由于在每个 $j \in J$ 处的取值都是线性对称列表，由引理 4.11，该同构必定唯一。类似地，存在唯一的复合同构：由 $\mathcal{K}_{\mathrm{SList}}$ 中复合的定义可见，给定 $f\colon J \to K$ 与 $g\colon K \to L$，复合$V(g) \circ V(f)$ 是一个函数，它将 $j \in J$映到与 $[g(f(j))]$ 同构的对称列表；列表是线性的这一事实再次保证了唯一性。

集合 $U(f)\colon = \mathcal{D}_{J,\,K}(V(f))$并令 $k \in K$。命题4.9让我们可以计算出

$$\|U(f)(k)\| = \sum_{j \in J} \mathfrak{c}(k, \|V(f)(j)\|) \cdot \{j\},$$

并且，由于根据 $V(f)$ 的定义，$\|V(f)(j)\| = \{f(j)\}$，我们得到 $\mathfrak{c}(k, \|V(f)(j)\|)$ 在 $k = f(j)$ 成立时等于 1，否则为 0。因此，

$$\|U(f)(k)\| = f^{-1}(\{k\}) \tag{2}$$

其中，集合 $f^{-1}(\{k\})$ 被视为一个所有元素重数均为一的多重集。特别地，$U(f)(k)$ 是一个线性的对称列表。类似地，给定 $f\colon J \to K$并且

$g\colon\ K \to L$，我们可以使用引理 4.8 来计算，对于所有 $l \in L$，

$$\begin{aligned}\|(U(g)\circ U(f))(l)\| &= \sum_{k\in K} \mathfrak{c}(k, \|U(g)(l)\|)\cdot\|U(f)(k)\| \\ &= \sum_{k\in K} \mathfrak{c}\left(k, g^{-1}\left(\{l\}\right)\right)\cdot f^{-1}\left(\{k\}\right) \\ &= \sum_{k\in g^{-1}(\{l\})} f^{-1}\left(\{k\}\right) \\ &= f^{-1}\left(g^{-1}(\{l\})\right).\end{aligned}$$

因此，由于前一个等式两边都是底层多重集相同的逐点线性对称列表，存在唯一的同构

$$U(g\circ f) \rightarrowtail U(f)\circ U(g)$$

由引理 4.11可知。类似的论证表明，存在唯一的同构 $U(\mathrm{Id}_J) \rightarrowtail \mathrm{Id}_{J}$。由于涉及的所有对称列表都是线性的，与 U 和 V的结合子和单位子的相容性自动成立，因为每个所需恒等式的两边都位于至多含有一个元素的集合中。

定义基变换同构并证明它们与纵向复合和水平复合相容时，也以类似方式用到了对称列表的线性：给定一个笛卡尔方块

$$\begin{array}{ccc} I & \xrightarrow{t} & J \\ {\scriptstyle l}\downarrow & \lrcorner\ S & \downarrow{\scriptstyle r} \\ K & \xrightarrow[b]{} & L \end{array}$$

有限集中的，我们可以利用引理 4.8、引理 4.9 和等式（2）计算得到，一方面，对于 $k \in K$，

$$\begin{aligned}\|(U(r)\circ V(b))(k)\| &= \sum_{l\in L} \mathfrak{c}(l, \{b(k)\})\cdot r^{-1}(\{l\}) \\ &= r^{-1}(\{b(k)\})\end{aligned}$$

这尤其是线性的。另一方面

$$\begin{aligned}\|(V(t)\circ U(l))(k)\| &= \sum_{i\in I} \mathfrak{c}\left(i, l^{-1}\left(\{k\}\right)\right)\cdot\{t(i)\} \\ &= \sum_{i\in l^{-1}(\{k\})} \{t(i)\}\end{aligned}$$

$j \in J$ 在 $\|(V(t) \circ U(l))(k)\|$ 中的重数，因此等于满足 $l(i) = k$ 且 $t(i) = j$的元素 $i \in I$ 所构成子集的基数。由于方块 S 是笛卡尔的，这个基数至多为一；当 $r(j) = b(k)$时，基数恰为一，因此总和

$$\sum_{i \in l^{-1}(\{k\})} \{t\ (i)\ \}$$ 也描述集合 $r^{-1}\ (\{b\ (k)\ \})$ 。

因此，由于两个对称列表的底层多重集相等，存在 某个 基变换同构 $U\ (r)\ \circ V\ (b)\ \leftrightharpoons V\ (t)\ \circ U\ (l)$ 。此外，由于所涉及的对称列表都是线性的，根据引理 4.11，这个基变换同构实际上必定是唯一的。

这个基变换同构对于拉回方块的纵向和横向粘合的相容性是自动成立的，对于纵向和横向单位方块的相容性也是如此。事实上，所有这些恒等式都可以表述为：给定的基变换同构等于态射的另一种复合。根据上述计算，所有基变换同构的源和目标逐点都是线性对称列表，因此根据引理 4.11的第二点，从源到目标存在唯一的 2-态射，所以所有这些恒等式都成立。

□

该定理的形式化遵循上述论证。其大部分内容是进行相关的多重集计算，以证明所有涉及的对称列表都实现了我们先前引入的 类型类。随后，所有 2-态射之间的相容性都可以用前面提到的 策略来解决。

命题4.12 完成了定理1.1的证明，因为该命题中定义的 Pith-Beck-Chevalley 系统定义了一个伪函子 $\Lambda\colon \mathrm{Pith}(\mathrm{Span}\,(\mathrm{Fin})) \to \mathcal{K}^{\mathrm{op}}\,\mathrm{SList}$，并使用定理3.3。最后，我们可以定义所需的伪函子，对称幺半范畴进行去偏，从而完成定理1.1的证明：

```
abbrev Λ : Bicategory.Pith (Spans FintypeCat ⊤ ⊤) ⥤ᵖ Kleisli^op := …
-- `Kleisli.pseudoOfSymmMonCat C` is the pseudofunctor called C_𝒦 in the text
universe v u in
def pseudoOfSymmMonCat
    (C : Type u) [Category.{v} C] [MonoidalCategory C] [SymmetricCategory C] :
    Bicategory.Pith (Spans FintypeCat.{0} ⊤ ⊤) ⥤ᵖ Cat.{v, u} :=
  Λ.comp (Kleisli.pseudoOfSymmMonCat C)
```

请注意，在上面的列表中，为了确保良好的宇宙性质，我们为 Kleisli 双范畴以及有限类型的跨度双范畴的髓使用了最低可能的宇宙层级。在 Lean 中，对于每个宇宙 ，都有一个有限类型范畴 u，它对应于对象的底层类型所处的宇宙。同样，实际上对于每个宇宙 ，都有一个由对称列表构成的 Kleisli 双范畴 u，而伪函子 Λ 实际上是由一族伪函子组成的 ，其中每个宇宙 u对应一个伪函子。如果 和 是不同宇宙中的类型，则函数 的类型只能保证位于 中。因此，一般而言，我们在命题 $\mathsf{C}_{\mathcal{K}}$ 中定义的伪函子 4.2 在使用以下方式进行构造时，可能会引入宇宙层级提升

`SList.Kleisli.{u}` 对于一个并非 u-小的幺半范畴。通过选取 u 为最低的宇宙层级 0 作为有限类型范畴的 Kleisli 双范畴的宇宙层级，我们确保不会发生层级提升，并保留执行该构造所基于的对称幺半范畴的原始宇宙层级。

5 结论与未来工作

总而言之，定理 1.1 中的伪函子是通过图表所描述的复合得到的：

$$\mathrm{Pith}(\mathrm{Span}(\mathrm{Fin})) \xrightarrow{\Lambda} \mathcal{K}^{\mathrm{op}} \xrightarrow{\mathsf{C}_{\mathcal{K}}} \mathrm{Cat}$$

其中 $\mathcal{K}$ 是相对伪单子对应的 Kleisli 双范畴，该相对伪单子将一个集合映射为该集合上的自由对称幺半范畴。伪函子 $\mathsf{C}_{\mathcal{K}}$ 是对称幺半范畴双范畴的（双范畴）Yoneda嵌入在自由对称幺半范畴上的限制，而伪函子 Λ 则将有限集与自由对称幺半范畴之间的联系汇集起来。我们所实现的对称幺半范畴相干定理版本，允许我们首先用对称列表更好地表示 $\mathcal{K}$；而推论 2.9 中对对称列表态射的完整描述，则完整展现了有限集的双射与对称列表之间的联系。

作为未来工作，我们计划形式化本文所构造的去偏方法的一些应用。我们预计，这项工作将使我们能够形式化对称幺半范畴中对象的张量幂。更具体地说，我们应当能够将赋值$x \mapsto x^{\otimes n}$ 构造为从范畴 C 到 C 中 Σ_n-等变对象的函子，具体做法是先定义一个函子

$$\mathrm{B}\Sigma_n \to \mathrm{End}_{\mathrm{Pith}(\mathrm{Span}(\mathrm{Fin}))}(\{*\}, \{*\})$$

将唯一的对象映射到集合 $\{1, \ldots, n\}$，然后令 $\mathsf{C}^{\otimes}$ 作用于其上，从而得到双函子 $\mathrm{B}\Sigma_n \to \mathsf{C}^{\otimes}(\{*\}) \to \mathsf{C}^{\otimes}(\{*\})$，我们可以交换变量的顺序，以生成所需的双函子。

我们这些构造的另一个可能推论是，能够对对称幺半范畴内部的交换幺半群对象进行去偏：将对称幺半范畴 x 内部的交换幺半群对象范畴 C 解释为松对称幺半函子 $\{*\} \to$ 范畴 C后，我们在命题4.4 中对松幺半函子进行的去偏，就给出了从（一个等价于）终伪函子的伪函子到范畴 $\mathsf{C}_{\mathcal{K}}$的松自然变换。当伪函子对松自然变换的晶须复合在 `Mathlib`中得到定义时，这将给出从 $\mathrm{Pith}(\mathrm{Span}(\mathrm{Fin}))$出发的伪函子之间的相应变换，用以编码交换幺半群对象 x的更高元数运算。

未来的工作还包括形式化这样一个事实：我们在本工作中构造的函子 Λ 在局部上给出范畴之间的等价，这为 Pith(Span(Fin))

确实是对称幺半普通范畴的“伪代数理论”，并全面形式化对称幺半普通范畴与 Cranch 的观点之间的联系 [6]。这一点可从如下事实看出：函数/纤维化对应实现了从 I 到 J 的有限集跨度的核与函数之间的等价$I \to \mathrm{Core}\left(\mathrm{Fin}_{/J}\right)$，后者通过相干定理等价于 $I \rightsquigarrow J$中的 $\mathcal{K}$ 1-态射范畴。因此，需要形式化一个自然同构：它从由 Λ 诱导的 1-态射上的函子与第二个等价的复合，指向函数/纤维化对应。我们预计这部分不会遇到重大障碍。

上一段提到的局部等价将使我们更接近我们认为去偏最彻底的形式：伪函子双范畴与对称幺半范畴双范畴之间的双等价。前者的对象为伪函子，1-态射为松、 oplax（分别）或强自然变换，即 Pith(Span(Fin)) →Cat 中保积的伪函子；后者的1-态射为松（分别为 oplax 或强）幺半函子的对称幺半范畴。为实现这一目标，还需要将缺失的基本双范畴构造形式化并加入 `Mathlib`，例如双等价、双范畴中的积、伪函子对积的保性，以及双等价的局部刻画等。

我们希望随着时间推移，将这项工作的成果贡献给 `Mathlib` 。我们坚信，无偏对称幺半范畴是实现对称幺半范畴中对象的更复杂操作（例如前文提到的一般幺半范畴中的对称张量幂）的关键一步；随着高阶范畴论在 `Mathlib` 中发展，它也将成为连接普通范畴论与高阶范畴论的必要条件。同样，我们希望将上一段提到的缺失双范畴基础设施贡献给 `Mathlib`。

参考文献

[1] Nathanael Arkor、Philip Saville 与 Andrew Slattery。相对伪单子的代数双范畴。2025。arXiv： 2501.12510 [math.CT]。url： https://arxiv.org/abs/2501.12510。[2] TomBachmann 与 Marc Hoyois。“动机同伦论中的范数”。载于：*Astérisque*425（2021年9月）。issn：2492-5926。doi： 10.24033/ast.1147。url： http://dx.doi.org/10.24033/ast.1147。[3] Ilya Beylin 与 Peter Dybjer。“从幺半群的规范化证明中提取幺半范畴的相干性证明”。载于：证明与程序的类型。Stefano Berardi 与 Mario Coppo 编。柏林、海德堡：Springer 柏林海德堡，1996，第47–61页。isbn：978-3-540-70722-6。[4] Anders Björner 与 Francesco Brenti。考克斯特群组合学。第231卷。数学研究生教材。Springer，纽约，2005，第xiv+363页。isbn：978- 3540-442387；3-540-44238-3。

[5] Francis Borceux。范畴代数手册。数学及其应用百科全书。剑桥大学出版社，1994。[6] James Cranch。“代数理论与（$\infty, 1$）-范畴”。载于：（2009）。arXiv：1011.3243 [math.AT]。url：https://arxiv.org/abs/1011.3243。[7] B. J. Day。“线性单子”。载于：澳大利亚数学学会通报17.2（1977），第177–192页。doi：10.1017/S0004972700010418。[8] M. Fiore、N. Gambino、M. Hyland 与 G. Winskel。“相对伪单子、Kleisli 双范畴与代换幺半群结构”。载于：Selecta Mathematica 24.3（2017年11月），第2791–2830页。issn：1420-9020。doi：10.1007/s00029-017-0361-3。url：http://dx.doi.org/10.1007/s00029-017-0361-3。[9] Claudio Hermida。“可表示多范畴”。载于：数学进展151.2（2000），第164–225页。issn：0001-8708。doi：https://doi.org/10.1006/aima.1999.1877。url：https://www.sciencedirect.com/science/article/pii/S0001870899918777。[10] Alexander E. Hoffnung。“2-范畴中的跨：一个幺半三范畴”。载于：（2013）。arXiv：1112.0560 [math.CT]。url：https://arxiv.org/abs/1112.0560。[11] J.F. Jardine。“超相干性”。载于：纯粹与应用代数杂志75.2（1991），第103–194页。issn：0022-4049。doi：https://doi.org/10.1016/0022-4049(91)90122-I。url：https://www.sciencedirect.com/science/article/pii/002240499190122I。[12] leanprover-community。mathlib4。https://github.com/leanprover-community/mathlib4。版本 v4.28.0-rc1。2026。[13] Jacob Lurie。高阶代数。url：https://www.math.ias.edu/~lurie/papers/HA.pdf。[14] Jacob Lurie。Kerodon。url：https://kerodon.net/。[15] Saunders Maclane。“自然结合律与交换律”。载于：莱斯研究所小册子——莱斯大学研究49（1963），第28–46页。[16] Ieke Moerdijk 与 Ittay Weiss。“树状集合”。载于：代数与几何拓扑7.3（2007年11月），第1441–1470页。issn：1472-2747。doi：10.2140/agt.2007.7.1441。url：http://dx.doi.org/10.2140/agt.2007.7.1441。[17] Stefano Piceghello。“同伦类型论中幺半群胚与对称幺半群胚的相干性”。博士学位论文。卑尔根大学，2021。url：https://hdl.handle.net/11250/2830640。[18] Christopher J. Schommer-Pries。二维扩展拓扑场论的分类。2014。arXiv：1112.1000 [math.AT]。url：https://arxiv.org/abs/1112.1000。[19] mathlib 社区。“Lean 数学库”。载于：第9届ACM SIGPLAN 认证程序与证明国际会议论文集。CPP 2020。美国路易斯安那州新奥尔良：ACM，2020年1月。doi：10.1145/3372885.3373824。url：https://doi.org/10.1145/3372885.3373824。