arXiv:2601.05972v1 [cs.PL] 9 Jan 2026

CuTe 布局的范畴论基础

JackCarlisle JayShah Reuben Stern Paul VanKoughnett

Colfax Research

research@colfax-intl.com

2026年1月

摘要

英伟达的 CUTLASS 库提供了一套强大且表达力丰富的方法，用于描述和操作图形处理器上的多维张量数据。这些方法在概念上以 CuTe 布局这一抽象概念及其丰富的代数为基础，其中包括复合、逻辑乘积和逻辑除法等运算。本文聚焦于一类自然出现的可处理布局，提出一个范畴框架，以理解这种布局代数。为此，我们定义了两个范畴**Tuple** 和**Nest** ，其态射可导出布局。我们定义了这些范畴中态射的一系列运算，并证明它们与相应的布局运算相容。此外，我们还完整刻画了由该构造得到的布局。最后，我们提供了这些范畴构造的 Python 实现，并附上测试，以验证其与 CUTLASS 的行为一致。此实现可在我们的 Git 仓库中找到 https://github.com/ColfaxResearch/layout-categories。

目录

1 引言 **3**
1.1 主要结果概述 .61.2 组织结构 .81.3 相关工作 .91.4 实现.111.5 记号.17

2 布局 a 及其代数 18
2.1 平坦布局 18
2.1.1 元组 .182.1.2 基本定义 .192.1.3 基本运算 .272.1.4 平坦合并 .362.1.5 紧凑平坦布局 .
412.1.6 补集 .442.1.7 更多运算 .542.1.8 可处理平坦布局.56

2.2 嵌套元组 .58
2.2.1 概形 .582.2.2 基本定义 .602.2.3 替换.632.2.4 细化 .64

2.3 布局 67
2.3.1 基本定义 .672.3.2 基本运算 .702.3.3 合并.722.3.4 相对合并 .742.3.5 紧凑布局 .
772.3.6 补集 .782.3.7 复合 .802.3.8 逻辑除法 .812.3.9 逻辑乘积 .842.3.10 可处理布局 .85

3 布局范畴 **87**
3.1 范畴 **Tuple** 87
3.1.1 基本定义 .873.1.2 从元组态射到扁平布局 .913.1.3 示例 .993.1.4 元组态射的实现 .1033.1.5 元组态射上的运算 .106

3.2 范畴 **Nest** .124
3.2.1 基本定义 .1243.2.2 从嵌套元组态射到布局 .1253.2.3 示例 .1283.2.4 嵌套元组态射的实现 .1303.2.5 细化.1313.2.6 嵌套元组态射的运算 .139

4 计算 部分14 **7**
4.1 可处理布局的复合 .147
4.1.1 相互细化.1484.1.2 从相互细化到可复合态射 .1524.1.3 复合算法 .1534.1.4 示例 .1544.1.5 更一般的复合 .1604.1.6 复合可容许性 .161

4.2 逻辑除法与逻辑乘积.1634.2.1 逻辑除法示例 .1634.2.2 逻辑积示例 .164

A 入门 范畴导论 166
A.1 什么是 a 范畴? 166
A.2 什么是 a 函子? 169

第1章

引言

在现代计算中，尤其是在 GPU 编程中，性能在很大程度上取决于多维数据在内存中的存储和访问方式。我们关注的大多数数据——例如图像、视频和机器学习中的张量——本质上都是多维的，而计算机内存从根本上说是一维的。这意味着，当我们要加载、存储或以其他方式处理数据时，需要将其多维逻辑坐标映射到一维物理坐标。这种映射称为**布局**，对于正确且高效地读写内存至关重要。此外，对于图形处理器的 SIMT 执行模型而言，布局用于描述和处理数据上的线程划分。这对于确保优化的内存访问模式，以及正确调用用于支持张量核心等目标的专用硬件指令非常重要。

作为一个启发性示例，假设我们想存储这个 4×8 矩阵

$$A = \begin{bmatrix} 12.47 & 87.21 & 34.08 & 56.93 & 45.65 & 9.17 & 73.02 & 21.39 \\ 64.88 & 30.41 & 1.72 & 88.04 & 92.55 & 17.06 & 50.91 & 68.77 \\ 3.33 & 77.19 & 61.58 & 29.46 & 15.82 & 80.75 & 44.62 & 39.28 \\ 91.40 & 26.12 & 6.97 & 53.03 & 58.66 & 33.79 & 11.20 & 70.55 \end{bmatrix}$$

在内存中。为此，我们需要为 A的每个条目指定一个内存地址。我们通过为 A的第 (0,0) 个条目选择一个地址，并为 A的每个其他条目指定一个**偏移量**来实现这一点。一种常见的选择是**行主序**布局

$L^{\text{row}} = (4,8) : (8,1) \ =$

0	1	2	3	4	5	6	7
8	9	10	11	12	13	14	15
16	17	18	19	20	21	22	23
24	25	26	27	28	29	30	31

这个非 化 $L^{\text{row}} = (4,\ 8) : (8,\ 1)$ 表示矩阵中第 $(i,\ j)$ 个条目的偏移量是

$$(i, j) \cdot (8, 1) = 8i + j.$$

另一种常见的选择是**列优先**布局

$$L^{\text{col}} = (4,8) : (1,4) \;=$$

0	4	8	12	16	20	24	28
1	5	9	13	17	21	25	29
2	6	10	14	18	22	26	30
3	7	11	15	19	23	27	31

再次，记法 $L^{\text{col}} = (4,\ 8) : (1,\ 4)$ 表示矩阵中第 $(i,\ j)$ 个条目的偏移量为

$$(i, j) \cdot (1, 4) = i + 4j.$$

这些布局非常有用，但并不能满足所有用途。例如，在高性能计算中，人们通常通过以下方式计算矩阵乘积 AB：

1. 将操作数矩阵 A 和 B 划分为分块；2. 计算各个分块的矩阵乘积；3. 合并这些部分结果，得到完整结果 AB。

例如，我们可以将 4×8 矩阵 A 划分为 2×2 个分块，如下图所示。

$$A = \begin{bmatrix} \begin{bmatrix} 12.47 & 87.21 \\ 64.88 & 30.41 \end{bmatrix} & \begin{bmatrix} 34.08 & 56.93 \\ 1.72 & 88.04 \end{bmatrix} & \begin{bmatrix} 45.65 & 9.17 \\ 92.55 & 17.06 \end{bmatrix} & \begin{bmatrix} 73.02 & 21.39 \\ 50.91 & 68.77 \end{bmatrix} \\ \begin{bmatrix} 3.33 & 77.19 \\ 91.40 & 26.12 \end{bmatrix} & \begin{bmatrix} 61.58 & 29.46 \\ 6.97 & 53.03 \end{bmatrix} & \begin{bmatrix} 15.82 & 80.75 \\ 58.66 & 33.79 \end{bmatrix} & \begin{bmatrix} 44.62 & 39.28 \\ 11.20 & 70.55 \end{bmatrix} \end{bmatrix}$$

现在假设我们想从 A 中切出各个分块，并假定它在内存中按列主序格式排列。为此，可以手动计算偏移量，如下所示：对于第 $(i,\ j)$ 个分块，索引到该分块左上角条目的偏移量为 $2i + 8j$。另一方面，为了更好地组织这一计算，我们可以使用分块的**交错**布局

$$L^{\text{tiled}} =$$

0	2	8	10	16	18	24	26
1	3	9	11	17	19	25	27
4	6	12	14	20	22	28	30
5	7	13	15	21	23	29	31

其中，列由 A 的分块给出，行由分块形状内的坐标给出。在这里，我们使用 **反字典序** 对分块以及分块内的坐标进行线性枚举，因此布局 L^{tiled} 的顶层形状为（4，8）。

不过请注意， L^{tiled} 所示的交错模式意味着，它无法表示为布局（4，8）：（a， b），其中步幅为 a， b。相反，我们可以分解形状（4，8）的模式并定义

$$L^{\text{tiled}} = ((2,2),(2,4)) : ((1,4),(2,8)).$$

之前的偏移量计算 $2i + 8j$ 随后通过在坐标 (0，(i， j)) 上计算 L^{tiled} 得到，而分块布局本身由第一个模式给出。因此，在为 A 赋予布局 L^{tiled} 以形成 A^{tiled} 后，我们可以将 A 的第 (i， j) 个分块表示为切片

$$A_{i,j} = A^{\text{tiled}}(_,(i,j)).$$

CUTLASS 中提出的一个关键思想是，可以通过某些基本运算，从更简单的布局系统地推导出有用但更复杂的辅助布局，例如 L^{tiled}。对于 L^{tiled}，相关运算称为**逻辑除法**。如果我们写成

$$T = (2,2) : (1,4) =$$

0	4
1	5

对于分块布局，则 L^{tiled} 就是逻辑除法

$$L^{\text{tiled}} = L^{\text{col}} \oslash T$$

如下图所示。

$L^{\text{col}} =$

0	4	8	12	16	20	24	28
1	5	9	13	17	21	25	29
2	6	10	14	18	22	26	30
3	7	11	15	19	23	27	31

$T =$

0	4
1	5

$L^{\text{col}} \oslash T =$

0	2	8	10	16	18	24	26
1	3	9	11	17	19	25	27
4	6	12	14	20	22	28	30
5	7	13	15	21	23	29	31

除了逻辑除法之外，其他基本布局运算还包括**逻辑乘积**、**补**，以及最重要的**复合**。这些布局运算是 CUTLASS 的基础，深入理解其行为有助于编写正确且高性能的

代码。然而，这些运算的定义和构造相当微妙。例如，布局 $B \circ A$ 与 A 和 B 的复合，仅当 A 和 B 满足特定的整除约束时才有明确定义，而 CUTLASS 会在底层检查这些约束。尤其是，两个布局何时可复合，以及应如何理解它们的复合，并不总是显而易见。

1.1 主要结果概述

本文的主要思路是，通过将注意力限制在**可处理布局**上，构建一个直观而强大的布局数学框架。这些布局的条目满足一个简单的整除条件（见定义 2.3.10.1）。可处理布局几乎涵盖了实践中遇到的所有布局，例如

- 行主序 和列优先 布局，它们无处不在，
- 紧致 布局，它们将数据存储在连续内存地址中，
- 投影，它们会广播数据的多个副本，以及
- 膨胀，它们支持填充加载和存储。

如果 L 是可处理布局，那么我们可以用一个**图表**来表示 L。例如，布局 L^{row}、L^{col} 和 L^{tiled} 可由以下图表表示。

$$(4,8):(8,1) \quad \rightsquigarrow \quad \begin{array}{ccc} 8 & & 4 \\ & \searrow\!\!\nearrow & \\ 4 & & 8 \end{array}$$

$$(4,8):(1,4) \quad \rightsquigarrow \quad \begin{array}{ccc} 8 & \longmapsto & 8 \\ 4 & \longmapsto & 4 \end{array}$$

$$((2,\ 2),\ (2,\ 4)):((1,\ 4),\ (2,\ 8)) \quad \rightsquigarrow \quad \begin{array}{ccc} & & 4 \longmapsto 4 \\ 8 & \begin{array}{c}\nearrow\\ \searrow\end{array} & \begin{array}{c} 2 \\ 2 \end{array} \begin{array}{c}\searrow\!\!\nearrow\\ \searrow\!\!\nearrow\end{array} \begin{array}{c} 2 \\ 2 \end{array} \\ 4 & \begin{array}{c}\searrow\\ \nearrow\end{array} & 2 \longmapsto 2 \end{array}$$

这些图表可以解释为**范畴**中的态射。这样，我们便能借助**范畴论** 的强大工具来描述布局及其运算。[1]

更准确地说，我们定义一个范畴 **Nest**，其对象是由正整数组成的嵌套元组，其态射 $f: S \to T$ 对应于上述这类图示（详情参见定义 3.1.1.13 和定义3.2.1.1）。如果 L 是一个**非退化** 可处理布局（参见定义2.3.1.24），那么存在一个本质上唯一的 **Nest**-态射 f，它编码了 L，如下列对应定理所示。

定理 A. *(参见3.2.2.15)* 存在一一对应

$$\left\{ \begin{array}{c} \textit{Non-degenerate} \\ \textit{tractable layouts} \end{array} \right\} \longleftrightarrow \left\{ \begin{array}{c} \textit{Non-degenerate} \\ \textbf{Nest}\textit{-morphisms} \\ \textit{of standard form} \end{array} \right\}$$

[1]我们在附录A 为不熟悉该主题的读者简要介绍了范畴论。

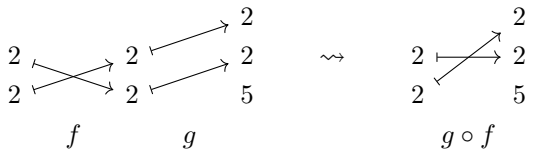
诸如复合、逻辑除法和逻辑乘积等布局运算，都可以在范畴**Nest**中得到自然的解释。如果

$$S \xrightarrow{f} T \xrightarrow{g} U$$

是 **Nest**态射，那么我们可以构造复合

$$S \xrightarrow{g \circ f} U$$

将相关图示粘合在一起即可。例如，

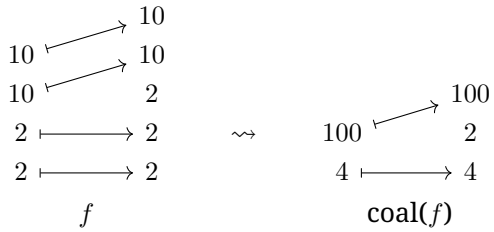


我们证明，**Nest** 中的复合与布局复合相容。

定理 B. *(见 3.2.6.21)* 若 f 和 g 是非退化的可复合**Nest**态射，则

$$L_{g \circ f} = L_g \circ L_f.$$

我们可以通过合并相邻箭头来合并一个**Nest**态射 f。例如，

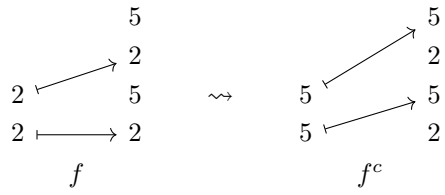


我们证明，此运算与布局合并相容。

定理 C. *(见 3.2.6.13)* 若 f 是一个 **Nest**-态射，则

$$L_{\mathsf{coal}(f)} = \mathsf{coal}(L_f).$$

Nest-态射 f 的补是未被 f命中的条目的包含映射。例如，



我们证明，**Nest** 中的补集与布局补相容。

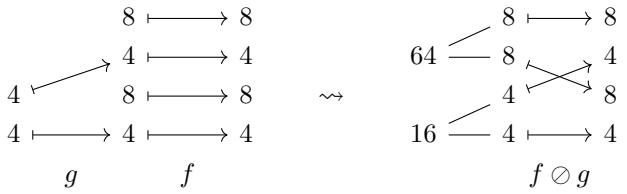
定理 D. *(见 3.2.6.20)* 若 $f\colon S \to T$ 是一个单射 **Nest**-态射，且 $N =$ 大小(T)，则

$$\mathsf{coal}(L_{f^c}) = \mathsf{comp}(L_f, N).$$

我们定义 **Nest**-态射的整除关系，以及逻辑除法运算

$$f, g \mapsto f \oslash g$$

当 g 整除 f 时。例如，



我们证明，**Nest** 中的逻辑除法与布局的逻辑除法相容。

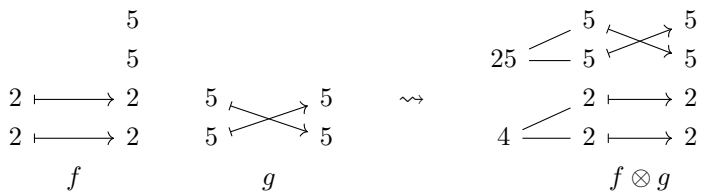
定理 E. （见 *3.2.6.26*）若 f 和 g 是非退化的 **Nest**-态射，且 g 整除 f，则

$$\mathsf{coal}(L_{f \oslash g}) = \mathsf{coal}(L_f \oslash L_g).$$

我们定义 **Nest**-态射的乘积可容许性，以及一种逻辑乘积运算

$$f, g \mapsto f \otimes g$$

当 f 和 g 乘积可容许时。例如，



我们证明，**Nest** 中的逻辑乘积与布局的逻辑乘积相容。

定理 F. *(*参见 *3.2.6.31*）若 f 和 g 是非退化的 **Nest**-态射，且 f 和 g 乘积可容许，则

$$L_{f \otimes g} = L_f \otimes L_g.$$

在第 4 章中，我们将说明如何使用我们的新框架来计算重要的布局运算，例如复合、逻辑除法和逻辑乘积。具体而言，我们给出一种算法（算法 4.1.3），用于计算可处理布局 $B \circ A$ 的复合，其中 A 和 B为可处理布局。略去细节，我们算法的基本思路是：若要计算复合 $B \circ A$，可以用恰当选取的 **Nest**-态射 f 和 g来表示 A 和 B，将这些态射复合，得到$g \circ f$，再取其编码布局，从而得到

$$B \circ A = L_{g \circ f}.$$

我们通过许多示例来说明这一算法。

1.2 组织结构

本文的组织结构如下。

在第1.4节中，我们将详细介绍布局的 `cute` 实现。我们以模块 `tract`的形式提供范畴**Nest** 的 Python 实现 `tract`，并说明 `tract` 与 `cute`的兼容性。我们的 Python 实现可以在 Git 仓库中找到 `https://github.com/ColfaxResearch/layout-categories`。

第 2 章是关于布局及其代数的全面参考资料。本章对布局及其支持的运算给出了严谨的定义，并确立了这些运算的基本性质。本章包含大量示例，对从业程序员也可能有所帮助。

在第3章中，我们提出了一个用于处理可处理布局的新数学框架。具体而言，我们将布局及其代数与**范畴**和**算子**理论联系起来。本章内容本身具有独立的数学价值，同时也具有实际意义，因为它提供了一个用于可视化布局并计算其各种运算的新框架。

在第 4章中，我们给出了一个用于计算可处理布局复合的算法 A 和B ，该算法使用了第 3章中建立的框架。我们通过许多示例说明了这一复合算法。

1.3 相关工作

尽管本研究具有理论性质，但其动机来自 GPU 编程中的实际应用，尤其是 CUTLASS。我们强调，此处发展的理论与实现无关：它不依赖于实际实现布局时所使用的特定编程语言或运行时系统。不过，在处理具体实现时，仍会涉及某些实际考量。例如，CUTLASS 区分编译时常量（静态变量）和运行时值（动态变量）。这些信息有助于在代码生成过程中进行编译器优化。这类特定于实现的细节虽然对性能至关重要，却不属于我们数学框架的讨论范围。有关此主题的进一步讨论，请参阅 CuTe 文档 [5]。

我们为布局构建的数学框架，与计算机科学和数学中的多个领域密切相关。我们简要回顾了 GPU 编程及其相邻领域的相关工作，以便更全面地了解我们的贡献。

- **CUTLASS 的应用**。 CUTLASS 的先进应用包括 FlashAttention[7, 21], EVT [4],和 SonicMoE [10]。对于希望深入了解 CUTLASS 和 CuTe 实际用法的读者，我们推荐英伟达 [3, 24, 25]和 Colfax Research [18, 20, 17, 19] 提供的、关于使用这些库进行 GPU 编程的完整教程系列。

- **数据布局优化** 数据布局优化技术旨在通过仔细考虑张量在内存中的存储方式，改善缓存局部性和内存访问模式 [30, 8, 15,11], [22]。选择高效的内存存储和访问模式对于图形处理器性能至关重要，因为内存带宽往往是性能瓶颈。

- **现代布局系统** 像 CuTe [5, 6, 16]和 Triton 线性布局[14, 32] 这样的布局系统，已成为管理张量计算中内存存储与访问的行业标准。Triton 线性布局基于 $\mathbb{F}_2$-线性代数，并继承了 $\mathbb{F}_2$-线性算子复合所带来的复合结构。这些布局也天然兼容布局置换，而布局置换通常无法表示为 CuTe 布局。另一方面，这些布局的表达能力不如 CuTe 布局，因为它们的大小和余大小必须等于 2 的幂，且无法表达通过非 2 的幂整数进行缩放等变换。最近有研究表明，这两种布局系统都可以用

整数集关系 [1]来表示。这为处理 CuTe 和 Triton 线性布局以及更一般的布局（例如具有非矩形形状的布局）提供了共同基础。

- **多面体编译** 多面体模型 [28], [29], [26] 为分析和变换具有仿射边界和数组访问的循环嵌套提供了数学框架。该模型的主要抽象是将迭代空间表示为某个多面体中的整数点集合。这种形式化方法支持复杂的循环变换，在保持程序语义的同时优化局部性和并行性。Pluto[2], 、Polly [9],和 Tensor Comprehensions [27] 等工具利用多面体技术自动生成优化后的代码。

- **张量收缩/分解** 张量缩并 [23, 31, 12]将矩阵乘法推广到更高秩的张量，在机器学习和科学计算中无处不在。高效实现张量缩并，依赖于对收缩顺序和中间张量布局的最优选择。

1.4 实现

在本节中，我们将演示如何在英伟达的 CuTe DSL 中使用布局，我们将其记为cute。我们以 Python 模块tract 的形式，在 Git 仓库 https://github.com/ColfaxResearch/layout-categories中提供了范畴框架的实现。在此，我们展示 cute 与 tract的兼容性。

1. **构造元组和嵌套元组：**我们按如下方式在 Python 中构造元组和嵌套元组。

```
S = (2,2,2)
T = ((2,2),(5,5))
U = ((2,2),4,(9,(3,3)))
```

注意，如果要构造长度为 1 的元组，必须在元组的条目后加上逗号。例如，

```
S = (10,)
T = (10)
```

返回

```
S = (10,)
T = 10
```

2. **构造布局和态射：**我们构造一个布局

$$L = S : D$$

在 cute 中，如下所示。

```
L = cute.make_layout(shape=S, stride=D)
```

例如，

```
A = cute.make_layout(shape=((4,4),4), stride=((16,1),4))
B = cute.make_layout(shape=(8,64), stride=(64,1))
C = cute.make_layout(shape=100, stride=2)
```

返回值

```
A = ((4,4),4):((16,1),4)
B = (8,64):(64,1)
C = 100:2
```

我们构造一个嵌套元组态射

$$S \xrightarrow[\alpha]{f} T$$

在 tract 中，如下所示。

```
f = tract.make_morphism(domain=S, codomain=T, map_=alpha)
```

例如，

```
f = tract.make_morphism(domain=(4,4), codomain=(4,2,4), map_=(1,3))
g = tract.make_morphism(domain=(2,2,2,2), codomain=(2,2,2,2), map_=(1,0,4,2))
h = tract.make_morphism(domain=(16,(4,4),(4,4)), codomain=(16,4,4), map_=(1,2,0,3,0))
```

返回值

```
f = (4,4)--(1,3)-->(4,2,4)
g = (2,2,2,2)--(1,0,4,2)-->(2,2,2,2)
h = (16,(4,4),(4,4))--(1,2,0,3,0)-->(16,4,4)
```

请注意，在 `tract` 中指定映射时，我们使用符号 0，而不是 $*$。

3. **在可处理布局与态射之间转换：**若 L 是一个布局，我们可以检查 L 是否可处理，方法是

```
tract.is_tractable(L)
```

例如，

```
A = cute.make_layout(shape=(2,2,2), stride=(1,2,4))
B = cute.make_layout(shape=(2,2,2), stride=(1,7,4))
A_is_tractable = tract.is_tractable(A)
B_is_tractable = tract.is_tractable(B)
```

返回值

```
A = (2,2,2):(1,2,4)
B = (2,2,2):(1,7,4)
A_is_tractable = True
B_is_tractable = False
```

如果 L 是可处理布局，那么我们可以用以下方式构造标准表示 f_L

```
tract.compute_morphism(L)
```

例如，

```
L = cute.make_layout(shape=(2,2,2), stride=(1,2,4))
f_L = tract.compute_morphism(L)
```

返回

```
L = (2,2,2):(1,2,4)
f_L = (2,2,2)--(1,2,3)-->(2,2,2)
```

如果 f 是一个嵌套元组态射，我们可以构造由 f 编码的布局 L_f，其中

```
tract.compute_layout(f)
```

例如，

```
f = tract.make_morphism(domain=((5,5),8), codomain=(5,8,5), map_=(1,3,2))
L_f = tract.compute_layout(f)
```

返回值

```
f = ((5,5),8)--(1,3,2)-->(5,8,5)
L_f = ((5,5),8):((1,40),5)
```

4. **复合**：在有定义时，此运算会从一对布局 $B \circ A$ 生成布局 A 和B。精确定义请参见定义 2.3.7.1 。我们可以在 cute中使用计算复合 $B \circ A$

```
cute.composition(B,A)
```

例如，运行

```
A = cute.make_layout(shape=((4,4),4), stride=((16,1),4))
B = cute.make_layout(shape=(8,64), stride=(64,1))
B_o_A = cute.composition(B,A)
```

返回

```
A = ((4,4),4):((16,1),4)
B = (8,64):(64,1)
B_o_A = ((4,4),(2,2)):((2,64),(256,1))
```

如果 f和 g是可复合嵌套元组态射，我们可以在 tract中使用以下方式计算复合 $g \circ f$

```
tract.compose(f,g)
```

例如，

```
f = tract.make_morphism(domain=((2,2),(2,2)), codomain=((2,2,2),(2,2,2)), map_=(3,2,6,5))
g = tract.make_morphism(domain=((2,2,2),(2,2,2)), codomain=(2,2,2,2), map_=(1,0,2,0,3,4))
g_o_f = tract.compose(f,g)
```

返回

```
f = ((2,2),(2,2))--(3,2,6,5)-->((2,2,2),(2,2,2))
g = ((2,2,2),(2,2,2))--(1,0,2,0,3,4)-->(2,2,2,2)
g_o_f = ((2,2),(2,2))--(2,0,4,3)-->(2,2,2,2)
```

5. **合并**：此操作会根据布局 coal(A) 生成一个布局 A。详情请参阅定义 2.3.3.1。我们可以在 coal(A) 中计算 cute，其中

```
cute.coalesce(A)
```

例如，

```
A = cute.make_layout(shape = ((2,2),(2,2),(5,5)), stride = ((1,2),(16,32),(64,640)))
coal_A = cute.coalesce(A)
```

返回

```
A = ((2,2),(2,2),(5,5)):((1,2),(16,32),(64,640))
coal_A = (4,20,5):(1,16,640)
```

此外还有一个相对合并操作 $A \mapsto \text{coal}(A,\ S)$，它接收一个额外的嵌套元组作为输入 S，该元组由形状进一步细化 A。详见定义2.3.4.7 。我们可以在$\text{coal}(A,\ S)$中计算 cute ，其中

```
A = cute.make_layout(shape = ((2,2),(3,3),(5,5)), stride = ((1,2),(4,12),(36,180)))
S = ((2,2),9,25)
coal_A_over_S = cute.coalesce(A,target_profile=S)
```

返回值

```
A = ((2,2),(3,3),(5,5)):((1,2),(4,12),(36,180))
S = ((2,2),9,25)
coal_A_over_S = ((2,2),9,25):((1,2),4,36)
```

如果 f 是嵌套元组态射，我们可以构造 $\text{coal}(f)$。详情请参见定义 3.2.6.11 。我们在 tract 中使用以下方式计算 $\text{coal}(f)$:

```
tract.coalesce(f)
```

例如,

```
f = tract.make_morphism(domain=(2,2,10,10), codomain = (2,2,2,10,10), map_=(1,2,4,5))
coal_f = tract.coalesce(f)
```

返回值

```
f = (2,2,10,10)--(1,2,4,5)-->(2,2,2,10,10)
coal_f = (4,100)--(1,3)-->(4,2,100)
```

6. **补**：在有定义时，此操作从布局 A和正整数 N生成布局 $\text{comp}(A,\ N)$。详情请参见定义 2.3.6.5。我们可以在 cute中使用以下方式计算 $\text{comp}(A,\ N)$:

```
cute.complement(A,N)
```

例如,

```
A = cute.make_layout(shape = ((2,2),(2,2)), stride = ((8,2),(64,256)))
comp_A = cute.complement(A,4096)
```

返回值

```
A = ((2,2),(2,2)):((8,2),(64,256))
comp_A = (2,2,4,2,8):(1,4,16,128,512)
```

如果 f 是一个嵌套元组态射，那么我们可以构造 f^c 关于 f 的补。详情请参见定义 3.2.6.17。我们在 tract 中计算 f^c，其中

```
tract.complement(f)
```

例如,

```
f = tract.make_morphism(domain=(2,2), codomain=(2,5,2,5), map_=(1,3))
comp_f = tract.complement(f)
```

返回值

```
f = (2,2)--(1,3)-->(2,5,2,5)
comp_A = (5,5)--(2,4)-->(2,5,2,5)
```

7. **逻辑除法**：当此运算有定义时，它会根据一对布局 $A \oslash B$ 和 A生成布局 B。详情请参见定义 2.3.8.1 。我们在 cute 中计算 $A \oslash B$ ，其中

```
cute.logical_divide(A,B)
```

例如，

```
A = cute.make_layout((64,32), stride = (32,1))
B = cute.make_layout((4,4), stride = (1,64))
quotient = cute.logical_divide(A,B)
```

返回

```
A = (64,32):(32,1)
B = (4,4):(1,64)
quotient = ((4,4),(16,8)):((32,1),(128,4))
```

若 f和 g是嵌套元组态射，且 g整除 f，则可以构造逻辑除法 $f \oslash g$。详情请参见定义 3.2.6.23 。我们在 tract 中计算 $f \oslash g$ ，其中

```
tract.logical_divide(f,g)
```

例如，

```
f = tract.make_morphism(domain=(4,8,4,8), codomain=(4,8,4,8), map_=(1,2,3,4))
g = tract.make_morphism(domain=(4,4), codomain=(4,8,4,8), map_=(1,3))
quotient = tract.logical_divide(f,g)
```

返回

```
f = (4,8,4,8)--(1,2,3,4)-->(4,8,4,8)
g = (4,4)--(1,3)-->(4,8,4,8)
quotient = ((4,4),(8,8))--(1,3,2,4)-->(4,8,4,8)
```

8. **逻辑乘积**：定义后，该操作会根据一对布局 $A \otimes B$ 生成布局 A和 B。详情请参阅定义 2.3.9.1 。我们在 cute 中计算 $A \otimes B$ ，其中包含

```
cute.logical_product(A,B)
```

例如，运行

```
A = cute.make_layout((3,10,10), stride = (200,1,20))
B = cute.make_layout((2,2), stride = (1,2))
product = cute.logical_product(A,B)
```

返回

```
A = (3,10,10):(200,1,20)
B = (2,2):(1,2)
product = ((3,10,10),(2,2)):((200,1,20),(10,600))
```

如果 f 和 g 是嵌套元组态射，并且 f 和 g 是乘积可容许的，那么我们可以构造逻辑乘积 $f \otimes g$。详见定义 3.2.6.28 。我们在 $f \otimes g$ 中计算 `tract` ，其中

```
tract.logical_product(f,g)
```

例如，

```
f = tract.make_morphism(domain=(2,2), codomain=(2,2,5,5), map_=(1,2))
g = tract.make_morphism(domain=(5,5), codomain=(5,5), map_=(2,1))
product = tract.logical_product(f,g)
```

返回值

```
f = (2,2)--(1,2)-->(2,2,5,5)
g = (5,5)--(2,1)-->(5,5)
product = ((2,2),(5,5))--(1,2,4,3)-->(2,2,5,5)
```

1.5 记号

$\mathbb{Z} = \{. . ., -1, 0, 1, 2, .. . \}\mathbb{N} = \{0,1, 2, .. . \}\mathbb{Z}_{>0} = \{1,2, . .. \}\mathbb{F}_2 = \{0,1\}$, 2 阶有限域。$[0, n) = \{0, .. . , n-1\}$，以及 $[0, 0) = \varnothing.n = \{1, 2, .. . , n\}$，且 $0 = \varnothing.n_* = \{*,1, 2, .. . , n\}\delta_i^m = (0,. . ., 1, . . . , 0)$，长度为 m 的元组，其中第 i 个条目为 1，其余条目均为 0。$\Sigma_n = n$上的对称群。$X^\sigma = (x_{\sigma(1)}, .. . , x_{\sigma(m)})$，其中 $X = (x_1, .. . , x_m)$ 是一个元组，$\sigma \in \Sigma_m.$ 是一个置换$X \star Y = X$ 和 Y.的平坦连接$X^\flat =$ 嵌套元组 $X.$ 的扁平化Xprof$(=)$嵌套元组 $X.$ 的轮廓$X_1(X_k, .. . , =)$ $X_1, .. . , X_k.(X_1, .. . , X_k)$ 的（嵌套）串接$Q =$ Q-替换: 将 $X_1, .. . , X_k$ 替换到轮廓 Q.中Tuple$(V) =$ 条目取自集合 $V.$ 的元组集合 Nest$(V) =$ 条目取自集合 $V.$ 的嵌套元组集合Profile $=$ 轮廓集合。FlatLayout $=$ 平坦布局集合。Layout $=$ 布局集合。$B \circ A =$ A 与 $B.$ 的复合$A \oslash B =$ A 对 $B.$ 的逻辑除法$A \otimes B =$ A 与 $B.$ 的逻辑乘积Set $=$ 集合范畴。FinSet $=$ 有限集范畴。Fin $=$ FinSet 的全子范畴，其对象为 $n \geq 0.$ 的 nFinSet$_*$ $=$ 带基点有限集范畴。Fin$_*$ $=$ FinSet$_*$ 的全子范畴，其对象为 $n \geq 0$ 的 n_*

Tuple $=$ 元组及元组态射构成的范畴。**Nest** $=$ 嵌套元组及嵌套元组态射构成的范畴

Ref $=$ 嵌套元组和细化的范畴。**Cat** $=$ （小）范畴和函子的范畴。

第 2 章

布局及其代数

本章旨在提供一套全面且具有数学基础的布局理论。我们首先在第 2.1 节构建平坦布局理论。在第 2.2 节介绍嵌套元组的必要背景知识，以便在第 2.3 节完整地涵盖布局的一般性。

2.1 扁平布局

本节将考察**扁平布局**。它是布局的一个重要子类，其形状和步幅均为元组，而不是更一般的嵌套元组。为了使讨论更加严谨，我们先确定与元组相关的记号。

2.1.1 元组

定义 2.1.1.1. 若 V 是一个集合，则一个取值于 V 的*Tuple* 是一个有限有序列表

$$X = (x_1, \dots, x_m)$$

对每个 $1 \le i \le m$ 的元素 $x_i \in V$。此类元组 $X = (x_1, \dots, x_m)$ 的长度 为

$$\mathsf{len}(X) = m.$$

我们用 Tuple(V) 表示所有条目属于 V 的元组所构成的集合。我们特别关注 $V = \mathbb{Z}$ 的情形，在这种情况下，我们称 $X \in \text{Tuple}(\mathbb{Z})$ 为整数元组。 如果 X 是一个整数元组，那么 X 的大小 是乘积

$$\mathsf{size}(X) = x_1 \cdots x_m.$$

例 2.1.1.2. 下面列出一些元组的示例，以及它们的长度和大小：

$$\begin{aligned} X &= (3, 128, 128), && \mathsf{len}(X) = 3, && \mathsf{size}(X) = 49152 \\ X &= (512), && \mathsf{len}(X) = 1, && \mathsf{size}(X) = 512 \\ X &= (), && \mathsf{len}(X) = 0, && \mathsf{size}(X) = 1 \end{aligned}$$

定义 2.1.1.3. 如果 $X = (x_1, \dots, x_m)$ 和 $Y = (y_1, \dots, y_n)$ 是元组，那么我们记作

$$X \star Y = (x_1, \dots, x_m, y_1, \dots, y_n)$$

X 和 Y的串接。

例 2.1.1.4. 如果 $X = (64,\ 32)$且 $Y = (8,\ 8,\ 8)$，那么

$$X \star Y = (64, 32, 8, 8, 8).$$

注记 2.1.1.5：如果 V是一个集合，那么所有条目属于 的元组所构成的集合

$$\mathsf{Tuple}(V) = \coprod_{m \geq 0} V^{\times m}$$

是 V上的自由结合幺半群。幺半群乘积是串接，单位元是空元组（）。

定义 2.1.1.6。 如果 X 和 X' 是元组，则称 X' 整除 X，如果存在一个元组 X''，使得

$$X' \star X'' = X.$$

例 2.1.1.7。 如果 $X' =$（81，9）且 $X =$（81，9，64，8），那么 X'整除 X，因为元组 $X'' =$（64，8）满足

$$X' \star X'' = X.$$

定义 2.1.1.8。 如果 $X =$（x_1，.....，x_m）是一个元组，而 $\sigma \in \Sigma_m$是一个置换，则记作 te

$$X^\sigma = (x_{\sigma(1)}, \ldots, x_{\sigma(m)})$$

关于 X 通过 σ进行的置换。这指定了 Σ_m 对 $\mathbb{Z}^{\times m}$的右作用。

例 2.1.1.9。 如果 $X =$(8，16，32，64) 且 $\sigma =$(1 2)(3 4)，那么

$$X^\sigma = (16, 8, 64, 32).$$

记号 2.1.1.10。 如果 n 是一个正整数，我们记作

$$[0, n) = \{0, 1, \ldots, n-1\},$$

并且如果 $S =$(s_1、...、 s_m) 是一个正整数元组，我们记作

$$[0, S) = [0, s_1) \times \cdots \times [0, s_m)$$

表示满足 $0 \leq x_i < s_i$的元组集合 (x_1、...、 x_m)。

例 2.1.1.11。如果 $S = (3,\ 2)$，那么

$$[0, S) = \{(0,0), (1,0), (2,0), (0,1), (1,1), (2,1)\}$$

2.1.2 基本定义

记号确定后，我们就可以定义平坦布局了。

定义 2.1.2.1. 一个平坦布局是一对

$$\begin{aligned} L &= S : D \\ &= (s_1, \ldots, s_m) : (d_1, \ldots, d_m) \end{aligned}$$

由一个正整数元组组成

$$\begin{aligned}\mathsf{shape}(L) &= S\\ &= (s_1, \ldots, s_m)\end{aligned}$$

称为 L 的形状，以及一个由非负整数组成的元组

$$\begin{aligned}\mathsf{stride}(L) &= D\\ &= (d_1, \ldots, d_m)\end{aligned}$$

称为 L 的步幅。

注记 2.1.2.2。 如果 L 是平坦布局，那么根据定义，形状(L) 和 步幅(L) 的长度相同。注记 2.1.2.3。 平坦布局是定义 *2.3.1.1* 中更一般的 布局 的一个例子，因此我们有时也将平坦布局 L 称为布局。

例 2.1.2.4。 以下是一些平坦布局的示例:

$$\begin{aligned}L_1 &= (2,2,2) : (1,2,4),\\ L_2 &= (128) : (5),\\ L_3 &= (16,12,512,512) : (0,0,1,512),\\ L_4 &= (6,1,12,2,2) : (2,0,12,144,1),\\ L_5 &= () : ().\end{aligned}$$

例 2.1.2.5。 我们可以将布局 $L = (8) : (5)$ 表示为

$L =$

0	5	10	15	20	25	30	35

我们可以将布局 $L = (3,\ 5) : (2,\ 10)$ 表示为

$L =$

0	10	20	30	40
2	12	22	32	42
4	14	24	34	44

我们精确 地说明这些图示如何表示注记2.1.2.17中的相关布局 .

扁平布局中最重要的示例可能是列优先和行主序布局，我们将在下文定义它们。

定义 2.1.2.6。设

$$L = (s_1, \ldots, s_m) : (d_1, \ldots, d_m)$$

是平坦布局。我们称 L 为列优先，如果

$$d_i = s_1 \cdots s_{i-1}$$

对于每个 $1 \leq i \leq m$。我们称 L 为行主序，如果

$$d_i = s_{i+1} \cdots s_m.$$

对于每个 $1 \leq i \leq m$。

例 2.1.2.7. 该布局

$$L = (3,4) : (1,3) = \begin{array}{|c|c|c|c|} \hline 0 & 3 & 6 & 9 \\ \hline 1 & 4 & 7 & 10 \\ \hline 2 & 5 & 8 & 11 \\ \hline \end{array}$$

为列优先，而该布局

$$L = (3,4) : (4,1) = \begin{array}{|c|c|c|c|} \hline 0 & 1 & 2 & 3 \\ \hline 4 & 5 & 6 & 7 \\ \hline 8 & 9 & 10 & 11 \\ \hline \end{array}$$

为行主序。这些图清楚地说明了这些术语的由来：如果 L是秩为 2 的列主序布局，那么 L 的各列是连续的；如果 L 是秩为 2 的行主序布局，那么 L的各行是连续的。

例 2.1.2.8. 这些布局

$$\begin{aligned} L_1 &= (2,2,2,2,2) : (1,2,4,8,16) \\ L_2 &= (3,128,128) : (1,3,384) \\ L_3 &= (64) : (1) \end{aligned}$$

为列优先，而这些布局

$$\begin{aligned} L_4 &= (2,2,2,2,2) : (16,8,4,2,1) \\ L_5 &= (3,128,128) : (16384,128,1) \\ L_6 &= (64) : (1) \end{aligned}$$

是行主序。

No现在我们已经看过几个示例，接下来定义平坦布局的一些重要属性 ts.

定义 2.1.2.9. 假设 $L = (s_1, \ldots, s_m) : (d_1, \ldots, d_m)$ 是一个平坦布局。

- L 的秩是

$$\mathsf{rank}(L) = m.$$

- L 的大小 是
$$\mathsf{size}(L) = \prod_{i=1}^{m} s_i.$$
- 余大小 L 的余大小是
$$\mathsf{cosize}(L) = 1 + \sum_{i=1}^{m} (s_i - 1) \cdot d_i.$$
- 对于任意秩为 $1 \le i \le$的 L， L 的第 i个模式 是这一对
$$\mathsf{mode}_i(L) = s_i : d_i.$$

例 2.1.2.10. 布局
$$L = (64, 32) : (1, 128)$$
的秩(*L*)为 $= 2$，大小(*L*)为 $= 2048$，余大小(*L*)为 $= 4032$。 L 的模式为
$$\mathsf{mode}_1(L) = 64 : 1$$
$$\mathsf{mode}_2(L) = 32 : 128.$$

例 2.1.2.11. 布局
$$L = (3, 8, 8, 8) : (1, 3, 24, 192).$$
的秩(*L*)为 $= 4$，大小(*L*)为 $= 1536$，余大小(*L*)为 $= 1536$。布局 L 有四个模式，例如模式$_3$(*L*) $= 8$: 24。

例 2.1.2.12. 布局
$$L = (2, 2, 2, 2, 2) : (160, 80, 40, 20, 10).$$
的秩为 rank(*L*) $= 5$，大小为 size(*L*) $= 32$，余大小为 cosize(*L*) $= 311$。布局 L 有 5 个模式，例如 mode$_5$(*L*) $= 2$: 10。

如果 L 是平坦布局，那么 L 编码了一个 坐标函数 φ_L。 L 的坐标函数是一个多维到一维变换，通过与步幅 stride(*L*) 求点积得到。回想一下，如果 $S = (s_1, \ldots, s_m)$ 是一个正整数元组，那么
$$[0, S) = [0, s_1) \times \cdots \times [0, s_m)$$
是所有满足 $0 \le x_i < s_i$ 的元组 $(x_1, \ldots, x_m)$ 的集合。特别地，如果 $S = ()$ 是空元组，那么 $[0, S) = \{()\}$。

构造 2.1.2.13 (坐标函数)。如果
$$L = (s_1, \ldots, s_m) : (d_1, \ldots, d_m)$$
是平坦布局，那么 L的坐标函数 是函数
$$[0, \mathsf{shape}(L)) \xrightarrow{\varphi_L} \mathbb{Z}$$
给出为
$$\begin{aligned}\varphi_L(x_1, \ldots, x_m) &= (x_1, \ldots, x_m) \cdot (d_1, \ldots, d_m) \\ &= x_1 d_1 + \cdots + x_m d_m.\end{aligned}$$

坐标函数 φ_L 可通过包含映射 $[0,$余大小$(L))\subset\mathbb{Z}$进行分解，我们记作

$$[0,\mathsf{shape}(L))\xrightarrow{\varphi_L^{\mathsf{cosize}(L)}}[0,\mathsf{cosize}(L))\subset\mathbb{Z}$$

对于分解映射。更一般地，对于任意 $N\geq$余大小(L)，我们将 φ_L^N 记作经由 $[0,N)\subset\mathbb{Z}$ 对 φ_L的分解，并稍微宽泛地使用术语，将这样的映射 φ_L^N称为 L的坐标函数。

例 2.1.2.14. 若 $L=(2,3):(1,5)$，则坐标函数

$$\varphi_L:[0,2)\times[0,3)\to\mathbb{Z}$$

由下式给出

$$\begin{aligned}\varphi_L(0,0)&=(0,0)\cdot(1,5)=0,\\ \varphi_L(1,0)&=(1,0)\cdot(1,5)=1,\\ \varphi_L(0,1)&=(0,1)\cdot(1,5)=5,\\ \varphi_L(1,1)&=(1,1)\cdot(1,5)=6,\\ \varphi_L(0,2)&=(0,2)\cdot(1,5)=10,\\ \varphi_L(1,2)&=(1,2)\cdot(1,5)=11.\end{aligned}$$

例 2.1.2.15. 若 $L=(2,2):(64,2)$，则坐标函数

$$\varphi_L:[0,2)\times[0,2)\to\mathbb{Z}$$

由下式给出

$$\begin{aligned}\varphi_L(0,0)&=(0,0)\cdot(64,2)=0,\\ \varphi_L(1,0)&=(1,0)\cdot(64,2)=64,\\ \varphi_L(0,1)&=(0,1)\cdot(64,2)=2,\\ \varphi_L(1,1)&=(1,1)\cdot(64,2)=66.\end{aligned}$$

例 2.1 .2.16. 如果 $E=():()$ 是空布局，那么 E 的坐标函数就是由下式给出的映射

$$\varphi_E:\{()\}\to\mathbb{Z}$$

给定为

$$\varphi(())=0.$$

注记 2.1.2.17。 现在，例如，我们可以精确描述该像所描绘的布局的含义

0	10	20	30	40
2	12	22	32	42
4	14	24	34	44

为 $L = (3,\ 5) : (2,\ 10)$：网格中的第（$i,\ j$）个单元格标记为值

$$\varphi_L(i,j) = (i,j) \cdot (2,10) = 2i + 10j$$

L 的坐标函数。

在实践中，平坦布局 L 最重要的不变量是其布局函数 Φ_L，它通过对坐标函数进行前复合而得到

$$\varphi_L : [0,S) \to \mathbb{Z}$$

与逆字典序同构的逆映射

$$\mathsf{colex}_S : [0,S) \to [0,\mathsf{size}(S)).$$

定义 **定义 2.1.2.18**。假设 $S = (s_1,\ldots,s_m)$ 是一个正整数元组，并回想 t

$$[0,S) = [0,s_1) \times \cdots \times [0,s_m).$$

该逆字典序同态是映射

$$[0,S) \xrightarrow{\quad\mathsf{colex}_S\quad} [0,\mathsf{size}(S))$$

$$(x_1,\ldots,x_m) \longmapsto \sum_{i=1}^{m} s_1 \cdots s_{i-1} x_i.$$

当元组 S 的含义可从上下文明确时，我们有时将 $\mathsf{colex} = \mathsf{colex}_S$ 简写。colexicographic 同构的逆映射是映射

$$[0,\mathsf{size}(S)) \xrightarrow{\ \mathsf{colex}_S^{-1}\ } [0,S)$$

定义如下

$$\mathsf{colex}_S^{-1}(x) = (x_1,\ldots,x_m)$$

其中

$$x_i = \left\lfloor \frac{x}{s_1 \cdots s_{i-1}} \right\rfloor \mod s_i.$$

注意，如果 $S = ()$ 是空元组，那么

$$\mathsf{colex}_{()} : \{()\} \to \{0\}$$

and

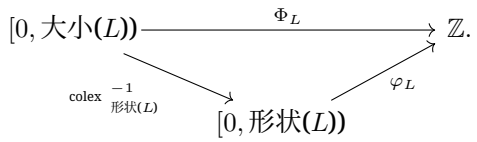
$$\mathsf{colex}_{()}^{-1} : \{0\} \to \{()\}$$

它们是典范同构。

构造 2.1.2.19（布局函数）。如果

$$L = (s_1,\ldots,s_m) : (d_1,\ldots,d_m),$$

是一个平坦布局，那么 的布局函数是复合



明确地说， Φ_L 由以下给出

$$\Phi_L(x) = x_1 d_1 + \cdots + x_m d_m$$

其中

$$x_i = \left\lfloor \frac{x}{s_1 \cdots s_{i-1}} \right\rfloor \mod s_i.$$

布局函数 Φ_L 可通过包含映射 $[0,$余大小$(L))$ $\subset \mathbb{Z}$进行分解，我们记作

$$[0, \mathsf{size}(L)) \xrightarrow{\Phi_L^{\mathsf{cosize}(L)}} [0, \mathsf{cosize}(L)) \subset \mathbb{Z}$$

表示分解映射。更一般地，对于任意 $N \geq$余大小(L)，我们将 Φ_L^N 记作 Φ_L通过 $[0, N) \subset \mathbb{Z}$的分解；稍微滥用一下术语，我们将这样的映射 φ_L^N称为 L的布局函数。

例 2.1.2.20. 若 $L = (2,3) : (1, 5)$，则布局函数

$$\Phi_L : [0, 6) \to \mathbb{Z}$$

由下式给出

$$\begin{aligned}
\Phi_L(0) &= (0, 0) \cdot (1, 5) = 0, \\
\Phi_L(1) &= (1, 0) \cdot (1, 5) = 1, \\
\Phi_L(2) &= (0, 1) \cdot (1, 5) = 5, \\
\Phi_L(3) &= (1, 1) \cdot (1, 5) = 6, \\
\Phi_L(4) &= (0, 2) \cdot (1, 5) = 10, \\
\Phi_L(5) &= (1, 2) \cdot (1, 5) = 11.
\end{aligned}$$

例 2.1.2.21. 若 $L = (2,2) : (64, 2)$，则布局函数

$$\Phi_L : [0, 4) \to \mathbb{Z}$$

由下式给出

$$\begin{aligned}
\Phi_L(0) &= (0, 0) \cdot (64, 2) = 0, \\
\Phi_L(1) &= (1, 0) \cdot (64, 2) = 64, \\
\Phi_L(2) &= (0, 1) \cdot (64, 2) = 2, \\
\Phi_L(3) &= (1, 1) \cdot (64, 2) = 66.
\end{aligned}$$

例 2.1.2.22. 如果 $L = (4, 2, 2) : (3,3, 100)$，那么例如， L 的布局函数满足

$$\begin{aligned}
\Phi_L(7) &= (3, 1, 0) \cdot (3, 3, 100) = 12, \\
\Phi_L(9) &= (1, 0, 1) \cdot (3, 3, 100) = 103.
\end{aligned}$$

例 2.1.2.23. 如果 $E =() : ()$ 是空布局，那么

$$\Phi_E : \{0\} \to \mathbb{Z}$$

由下式给出

$$\Phi_E(0) = 0.$$

例 2.1.2.24. 如果 L 是任意平坦布局，那么 L 的布局函数 Φ_L 满足

$$\Phi_L(0) = 0.$$

注记 2.1.2.25。 如果 $S = (s_1, \dots, s_m)$ 是一个正整数元组，那么逆字典序同构

$$[0, S) \xrightarrow{\mathsf{colex}_S} [0, \mathsf{size}(S))$$

等于列主序布局的坐标函数 $\varphi_L^{\mathsf{cosize}(L)}$

$$L = (s_1, s_2, \dots, s_m) : (1, s_1, \dots, s_1 \cdots s_{m-1}).$$

这意味着，如果平坦布局 L 是列主序的，那么

$$\begin{aligned}\Phi_L^{\mathsf{cosize}(L)} &= \varphi_L^{\mathsf{cosize}(L)} \circ \mathsf{colex}^{-1}_{\mathsf{shape}(L)} \\ &= \varphi_L^{\mathsf{cosize}(L)} \circ \left(\varphi_L^{\mathsf{cosize}(L)}\right)^{-1} \\ &= \mathsf{id}_{[0,\mathsf{size}(L))}\end{aligned}$$

就是 $[0,\mathsf{size}(L)$ 上的恒等映射）。

注记 2.1.2.26。 存在互不相同的布局 $A \neq B$，满足 $\Phi_A = \Phi_B$。例如，布局

$$\begin{aligned}A &= (7, 7) : (1, 7) \\ B &= (49) : (1)\end{aligned}$$

虽然不相等，但 $\Phi_A = \Phi_B$。稍后我们将精确刻画两个平坦布局 A 和 B 何时具有相同的布局函数（参见命题 2.1.4.18）。

在继续讨论布局运算之前，我们需要定义非退化的概念。

定义 2.1.2.27. 设

$$L = (s_1, \dots, s_m) : (d_1, \dots, d_m)$$

是一个平坦布局。若对任意 $1 \le i \le m$，都有以下关系，则称 L 是非退化的

$$s_i = 1 \quad \Rightarrow \quad d_i = 0.$$

例 2.1.2.28. 这些布局

$$\begin{aligned}L_1 &= (4, 1) : (1, 0) \\ L_2 &= (8, 1, 8, 1) : (2, 0, 16, 0)\end{aligned}$$

是非退化的，而这些布局

$$\begin{aligned}L_3 &= (4, 1) : (1, 4) \\ L_4 &= (8, 1, 8, 1) : (2, 16, 16, 256)\end{aligned}$$

是退化的。

观察 2.1.2.29. 假设布局 L 是非退化的，并不会造成实质性的一般性损失。更准确地说，如果

$$\begin{aligned}L &= (s_1, \dots, s_m) : (d_1, \dots, d_m) \\ L' &= (s_1, \dots, s_m) : (d'_1, \dots, d'_m)\end{aligned}$$

是形状相同的平坦布局，并且 $d_i = d'_i$ 只要 $s_i > 1$，那么 $\varphi_L = \varphi_{L'}$，且 $\Phi_L = \Phi_{L'}$。特别是，只要 $s_i = 1$，我们就可以自由地设定 $d_i = 0$，而不改变 L的坐标函数或布局函数。

2.1.3 基本运算

介绍完平坦布局的基本术语后，我们转而讨论它们支持的运算。本节将定义一些基本运算，这些运算对于构造更复杂的运算是必不可少的，例如合并、补和复合。

2.1.3.1 限制

如果 L 是一个平坦布局，通常需要将其限制到 L的模式的某个子集。回想一下，对于非负整数 m，我们记作

$$\langle m\rangle = \{1,\dots,m\}.$$

定义 2.1.3.1. 设

$$L = (s_1,\dots,s_m) : (d_1,\dots,d_m)$$

是一个平坦布局，并设

$$I = \{i_1 < \cdots < i_k\} \subset \langle m\rangle$$

是一个子集。我们将在 L上的 I 限制定义为平坦布局

$$L\mid_I = (s_{i_1},\dots,s_{i_k}) : (d_{i_1},\dots,d_{i_k}).$$

例 2.1.3.2. 如果

$L = (3,6) : (10,5) =$

0	5	10	15	20	25
10	15	20	25	30	35
20	25	30	35	40	45

且 $I = \{2\}$，则

$L\mid_I = (6) : (5) =$

0	5	10	15	20	25

例 2.1.3.3. 如果

$$L = (3,8,8,8) : (1,3,24,192)$$

且 $I = \{1,\ 2,\ 3\}$，则

$$L\mid_I = (3,8,8) : (1,3,24).$$

例 2.1.3.4. 如果

$$L = (s_1,\dots,s_m) : (d_1,\dots,d_m)$$

是平坦布局且为 $I = \langle m\rangle$，那么

$$L\mid_I = L.$$

例 2.1.3.5. 如果

$$L = (s_1,\dots,s_m) : (d_1,\dots,d_m)$$

是平坦布局且 $I = \varnothing$ 是空集，那么

$$L\mid_I = () : ()$$

是空布局。

2.1.3.2 挤压

如果 L 是平坦布局，那么操作 $L \mapsto$squeeze(L) 会移除 L 中所有满足 $s_i = 1$ 的模式 $s_i : d_i$ 。

构造 2.1.3.6. 设

$$L = (s_1, \ldots, s_m) : (d_1, \ldots, d_m)$$

是一个平坦布局，并令

$$I = \{i \in \langle m \rangle \mid s_i > 1\}$$

为对应形状条目不等于 1 的指标集合。我们定义

$$\mathsf{squeeze}(L) = L \mid_I .$$

例 2.1.3.7. 如果

$$L = (64, 64, 1) : (1, 64, 0),$$

then

$$\mathsf{squeeze}(L) = (64, 64) : (1, 64).$$

例 2.1.3.8. 如果

$$L = (64, 64, 1, 32, 1) : (2048, 32, 0, 1, 0)$$

then

$$\mathsf{squeeze}(L) = (64, 64, 32) : (2048, 32, 1).$$

例 2.1.3.9. 若 L 是平坦布局，则

$$\mathsf{squeeze}(L) = L$$

当且仅当 shape(L) 不包含值为 1 的条目时。

例 2.1.3.10. 若 L 是平坦布局，则

$$\mathsf{squeeze}(L) = () : ()$$

当且仅当 shape(L) 的所有条目都等于 1 时，它是空布局。

该构造的一个重要性质是， $L \mapsto$squeeze(L) 不会改变 L的布局函数。

引理 2.1.3.11. 若 L是平坦布局，则

1. 挤压(L) 的大小 =大小(L)，*2.* 挤压
(L) 的余大小 = 余大小(L)，以及 *3.*
$\Phi_{\mathsf{squeeze}(L)} = \Phi_L$。

证明。设

$$I = \{i_1 < \cdots < i_k\} \subset \langle m \rangle$$

用 $s_{i_j} > 1$ 表示指标集合，因此

$$\mathsf{squeeze}(L) = (s_{i_1}, \ldots, s_{i_k}) : (d_{i_1}, \ldots, d_{i_k}).$$

对于第一个断言，我们计算

$$\mathsf{size}(\mathsf{squeeze}(L)) = \prod_{j=1}^{k} s_{i_j} = \left(\prod_{j=1}^{k} s_{i_j}\right) \cdot \left(\prod_{\langle m\rangle \setminus I} 1\right) = \prod_{i=1}^{m} s_i = \mathsf{size}(L).$$

对于第二个断言，我们计算

$$\begin{aligned}\mathsf{cosize}(\mathsf{squeeze}(L)) = 1 + \sum_{j=1}^{k}(s_{i_j}-1)\cdot d_{i_j} &= 1 + \sum_{j=1}^{k}(s_{i_j}-1)\cdot d_{i_j} + \left(\sum_{\langle m\rangle\setminus I} 0\right)\\ &= 1 + \sum_{i=1}^{m}(s_i-1)\cdot d_i\\ &= \mathsf{cosize}(L).\end{aligned}$$

对于第三个断言，只需证明从平坦布局中移除形如 $1:d_i$ 的模式后，布局函数保持不变。设 $L = (s_1, \ldots, s_m) : (d_1, \ldots, d_m)$，并假设某个 $s_i = 1$。令

$$L' = (s'_1, \ldots, s'_{m-1}) : (d'_1, \ldots, d'_{m-1})$$

表示从 L 中移除第 i 个模式后得到的平坦布局，因此

$$s'_j = \begin{cases} s_j & j < i \\ s_{j+1} & i \le j < m, \end{cases} \qquad \text{and} \qquad d'_j = \begin{cases} d_j & j < i \\ d_{j+1} & i \le j < m. \end{cases}$$

L 的布局函数为

$$\Phi_L(x) = x_1 d_1 + \cdots + x_m d_m$$

其中 $x_j = \left\lfloor \frac{x}{s_1 \cdots s_{j-1}} \right\rfloor$ 模 s_j，而 L' 的布局函数为

$$\Phi_{L'}(x) = x'_1 d'_1 + \cdots + x'_{m-1} d'_{m-1}$$

其中 $x'_j = \left\lfloor \frac{x}{s'_1 \cdots s'_{j-1}} \right\rfloor$ 模 s'_j。我们注意到

$$x'_j = \begin{cases} x_j & j < i \\ x_{j+1} & i \le j < m, \end{cases}$$

并且由于 $x_i \in [0, 1)$ 必然为 0，我们有

$$\begin{aligned}\Phi_L(x) &= x_1 d_1 + \cdots + x_m d_m\\ &= x_1 d_1 + \cdots + x_{i-1} d_{i-1} + x_{i+1} d_{i+1} + \cdots + x_m d_m\\ &= x'_1 d'_1 + \cdots + x'_{m-1} d'_{m-1}\\ &= \Phi_{L'}(x).\end{aligned}$$

□

2.1.3.3 过滤零值

如果 L 是平坦布局，则操作 $L \mapsto \mathsf{filter}(L)$ 会移除所有满足 $d_i = 0$ 的模式 $s_i : d_i$。

定义 2.1.3.12. 假设

$$L = (s_1, \ldots, s_m) : (d_1, \ldots, d_m)$$

是一个平坦布局，并令

$$I = \{i \in \langle m \rangle \mid d_i > 0\}$$

为对应步长项不等于 0 的指标集合。我们定义

$$\mathsf{filter}(L) = L \mid_I .$$

例 2.1.3.13. 如果

$$L = (64, 8, 8, 128) : (8, 1, 0, 512)$$

then

$$\mathsf{filter}(L) = (64, 8, 128) : (8, 1, 512).$$

例 2.1.3.14. 如果

$$L = (3, 2) : (12, 0) =$$

0	0
12	12
24	24

then

$$\mathsf{filter}(L) = (3) : (12) =$$

0
12
24

例 2.1.3.15. 若

$$L = (3, 8, 8, 8) : (16, 0, 0, 0)$$

then

$$\mathsf{filter}(L) = (3) : (16).$$

例 2.1.3.16. 若 L 是平坦布局，则

$$\mathsf{filter}(L) = L$$

当且仅当 stride(L) 的所有条目均非零。

例 2.1.3.17. 若 L 是平坦布局，则

$$\mathsf{filter}(L) = () : ()$$

当且仅当 stride(L) 的所有条目都等于 0 时，它是空布局。

2.1.3.4 置换

回想一下，如果 $X = (x_1, \ldots, x_m)$ 是一个长度为 m 的 Tuple，且 $\sigma \in \Sigma_m$ 是一个置换，那么我们写作

$$X^\sigma = (x_{\sigma(1)}, \ldots, x_{\sigma(m)}).$$

将 X 按 σ 置换。

定义 2.1.3.18. 若 $L = (s_1, \ldots, s_m) : (d_1, \ldots, d_m)$ 是秩为 m 的平坦布局，且 $\sigma \in \Sigma_m$ 是一个置换，则定义

$$\begin{aligned} L^\sigma &= \mathsf{shape}(L)^\sigma : \mathsf{stride}(L)^\sigma \\ &= (s_{\sigma(1)}, \ldots, s_{\sigma(m)}) : (d_{\sigma(1)}, \ldots, d_{\sigma(m)}). \end{aligned}$$

例 2.1.3.19. 若

$$L = (4, 2) : (12, 2) = \begin{array}{|c|c|} \hline 0 & 2 \\ \hline 12 & 14 \\ \hline 24 & 26 \\ \hline 36 & 38 \\ \hline \end{array}$$

且 $\sigma = (1\ 2) \in \Sigma_2$ 是该对换，则

$$L^\sigma = (2, 4) : (2, 12) = \begin{array}{|c|c|c|c|} \hline 0 & 12 & 24 & 36 \\ \hline 2 & 14 & 26 & 38 \\ \hline \end{array}$$

是转置布局。

例 2.1.3.20. 若

$$L = (15, 12, 10) : (240, 1, 24)$$

且 $\sigma = (1\ 2) \in \Sigma_3$，则

$$L^\sigma = (12, 15, 10) : (1, 240, 24).$$

例 2.1.3.21. 若

$$L = (2, 2, 2, 2, 2) : (1, 2, 4, 8, 16)$$

且 $\sigma = (1\ 5)(3\ 2\ 4) \in \Sigma_5$，则

$$L^\sigma = (2, 2, 2, 2, 2) : (16, 8, 2, 4, 1).$$

例 2.1.3.22. 若

$$L = (s, \ldots, s) : (d, \ldots, d)$$

是一个所有模式都相等的平坦布局，那么对于任意 $\sigma \in \Sigma_m$，我们有

$$L^\sigma = L.$$

2.1.3.5 排序

如果 L 是一个平坦布局，通常可以置换 L，使其模式按递增顺序排列，具体如下。

定义 2.1.3.23. 我们定义整数对 $s:d$ 上的线性序如下

$$s:d \preceq s':d' \quad \Leftrightarrow \quad \begin{array}{c} d<d', \text{ or} \\ d=d' \text{ and } s\leq s'. \end{array}$$

例 2.1.3.24. 我们有

$$5:8 \preceq 4:12 \preceq 5:12.$$

定义 2.1.3.25. 设 L 是一个平坦布局。若对于任意 $1 \leq i <$rank(L)，我们都有，则称 L 已排序

$$\mathsf{mode}_i(L) \preceq \mathsf{mode}_{i+1}(L).$$

例 2.1.3.26. 这些布局

$$L_1 = (128,64,2,2):(1,128,8192,16384)$$
$$L_2 = (2,2,2):(1,1,1)$$

是已排序的，而这些布局

$$L_3 = (2,4,8,16):(64,1,2,4)$$
$$L_4 = (5,32,16):(1,5,5)$$

并非已排序。

例 2.1.3.27. 空布局 $E = () : ()$ 已排序。

例 2.1.3.28. 若

$$L = (s_1,\ldots,s_m):(0,\ldots,0)$$

是一个所有步幅(L)条目均等于 0 的平坦布局，那么 L 已排序当且仅当

$$s_1 \leq s_2 \leq \cdots \leq s_m.$$

平坦布局 L 是否已排序，与布局函数Φ_L 对 L的行为密切相关，如下述引理所述。

Le引理 2.1.3.29. 设 L 是一个平坦布局。若 Φ_L 单调不减，则 L 已排序。

证明。 我们证明逆否命题。假设 L 未排序。我们将证明，在 Φ_L 的定义域中存在某个$x \leq y$，使得 $\Phi_L(x) > \Phi_L(y)$。如果存在某个 $1 \leq i < m$，使得 $d_i > d_{i+1}$，那么我们可以令

$$x = \prod_{j<i} s_j, \quad \text{and} \quad y = \prod_{j<i+1} s_j,$$

在这种情况下， $x < y$，但是

$$\begin{aligned} \Phi_L(x) &= (0,\ldots,1,0,\ldots,0)\cdot(d_1,\ldots,d_i,d_{i+1},\ldots,d_m) \\ &= d_i \\ &> d_{i+1} \\ &= (0,\ldots,0,1,\ldots,0)\cdot(d_1,\ldots,d_m) \\ &= \Phi_L(y). \end{aligned}$$

另一方面，如果存在某个 $1 \leq i < m$，使得 $d_i = d_{i+1}$ 和 $s_i > s_{i+1}$，我们可以设定

$$x = (s_i - 1)\left(\prod_{j<i} s_j\right), \qquad \text{and} \qquad y = (s_{i+1} - 1)\left(\prod_{j<i+1} s_j\right),$$

在这种情况下 $x < y$，但是

$$\begin{aligned}
\Phi_L(x) &= (0, \ldots, s_i - 1, 0, \ldots, 0) \cdot (d_1, \ldots, d_i, d_{i+1}, \ldots, d_m) \\
&= (s_i - 1)d_i \\
&> (s_{i+1} - 1)d_i \\
&= (s_{i+1} - 1)d_{i+1} \\
&= (0, \ldots, 0, s_{i+1} - 1, \ldots, 0) \cdot (d_1, \ldots, d_m) \\
&= \Phi_L(y).
\end{aligned}$$

我们得出结论， Φ_L 不是非递减的。 □

注记 2.1.3.30. 前一个引理的逆命题不成立。例如，平坦布局

$$L = (3, 5, 7) : (1, 1, 1)$$

已排序，但是

$$\Phi_L(7) = (0, 2, 0) \cdot (1, 1, 1) = 2$$

严格大于

$$\Phi_L(16) = (0, 0, 1) \cdot (1, 1, 1) = 1.$$

如果 L 是一个平坦布局，那么我们可以置换 L 的模式，得到已排序布局 sort(L)。

构造 2.1.3.31. 假设

$$L = (s_1, \ldots, s_m) : (d_1, \ldots, d_m)$$

是一个平坦布局。定义 $\langle m \rangle$ 上的线性序 $\preceq$，使得 $i \preceq j$ 当且仅当

1. 模式$_i$(L) $\preceq$模式$_j$(L)，且 2. 如果模式$_i$(L) =模式$_j$(L)，那么 $i \leq j$。

令 $\sigma \in \Sigma_m$ 为与 $\langle m \rangle$ 上的线性序 $\preceq$ 相关联的置换。我们将 sort(L) 定义为 L 的置换，按照 σ 进行：

$$\mathsf{sort}(L) = L^{\sigma}.$$

例 2.1.3.32. 如果

$$L = (2, 4, 8, 16) : (64, 1, 2, 4)$$

then

$$\mathsf{sort}(L) = (4, 8, 16, 2) : (1, 2, 4, 64).$$

例 2.1.3.33. 如果

$$L = (5, 32, 16) : (1, 5, 5)$$

then

$$\mathsf{sort}(L) = (5, 16, 32) : (1, 5, 5).$$

例 2.1.3.34. 如果 L 已排序，那么 $\mathsf{sort}(L) = L$。特别地，这意味着 $\mathsf{sort}(-)$ 是一种幂等运算：

$$\mathsf{sort}(\mathsf{sort}(L)) = \mathsf{sort}(L).$$

观察 2.1.3.35。 如果 L 是平坦布局，那么通常有 $\Phi_{\mathsf{sort}(L)} \neq \Phi_L$。然而，布局函数 Φ_L 和 $\Phi_{\mathsf{sort}(L)}$ 始终具有相同的像。为说明这一点，我们写成

$$L = (s_1, \ldots, s_m) : (d_1, \ldots, d_m), \text{ and}$$
$$\mathsf{sort}(L) = (s_{\sigma(1)}, \ldots, s_{\sigma(m)}) : (d_{\sigma(1)}, \ldots, d_{\sigma(m)})$$

对于某个置换 $\sigma \in \Sigma_m$。如果整数 n 在 Φ_L的像中，那么存在一个元组 $(x_1, \ldots\ldots, x_m) \in \prod_{i=1}^{m} [0, s_i)$，使得

$$x_1 d_1 + \cdots + x_m d_m = n$$

在这种情况下，元组 $(x_{\sigma(1)}, \ldots\ldots, x_{\sigma(m)}) \in \prod_{i=1}^{m} [0, s_{\sigma(i)})$ 满足

$$x_{\sigma(1)} d_{\sigma(1)} + \cdots + x_{\sigma(m)} d_{\sigma(m)} = n.$$

This 这证明了 $\mathsf{Image}(\Phi_{\mathsf{sort}}(L)) \subseteq \mathsf{Image}(\Phi_L)$，而反向的包含映射可以类似地证明。

2.1.3.6 拼接

回顾一下，如果 $X = (x_1, \ldots\ldots, x_m)$ 和 $Y = (y_1, \ldots\ldots, y_n)$ 是元组，那么 X 和 Y 的串接是元组

$$X \star Y = (x_1, \ldots, x_m, y_1, \ldots, y_n).$$

这个定义自然地扩展到平坦布局的串接。

定义 2.1.3.36. 设

$$L_1 = S_1 : D_1$$
$$L_2 = S_2 : D_2$$

是平坦布局。那么， L_1 和 L_2 的串接是平坦布局

$$L_1 \star L_2 = S_1 \star S_2 : D_1 \star D_2.$$

平坦布局的串接满足结合律，因此更一般地，若 $L_1, \ldots, L_k$ 是平坦布局，我们可以构造串接

$$L_1 \star \cdots \star L_k.$$

例 2.1.3.37. 若 $L_1 = (7,2) : (2,1)$ 且 $L_2 = (3,3,3) : (0,10,30)$，则

$$L_1 \star L_2 = (7,2,3,3,3) : (2,1,0,10,30).$$

例 2.1.3.38. 如果 $E = () : ()$ 是空布局，那么对于任意平坦布局 L，我们有

$$L \star E = L = E \star L.$$

观察 2.1.3.39. 假设

$$L = (s_1, \dots, s_m) : (d_1, \dots, d_m)$$

是一个平坦布局。若记

$$L_i = (s_i) : (d_i),$$

则可将 L写成串接

$$L = L_1 \star \cdots \star L_m.$$

若 $L_1, \dots, L_k$是平坦布局，则串接 $L_1 \star \cdots \star L_k$的布局函数由 $L_1, \dots, L_k$的布局函数确定，如下所示。

命题 2.1.3.40。 设 $L_1, \dots, L_k$分别是形状为 $S_1, \dots, S_k$、大小为 $N_1, \dots, N_k$的平坦布局。那么坐标函数

$$[0, S_1 \star \cdots \star S_k) \xrightarrow{\varphi_{L_1 \star \cdots \star L_k}} \mathbb{Z}$$

$L_1 \star \cdots \star L_k$的坐标函数等于复合

$$[0, S_1 \star \cdots \star S_k) \xrightarrow{\cong} [0, S_1) \times \cdots \times [0, S_k) \xrightarrow{\varphi_{L_1} + \cdots + \varphi_{L_k}} \mathbb{Z},$$

$$X_1 \star \cdots \star X_k \longleftrightarrow (X_1, \dots, X_k)$$

而布局函数

$$[0, N_1 \cdots N_k) \xrightarrow{\Phi_{L_1 \star \cdots \star L_k}} \mathbb{Z}$$

$L_1 \star \cdots \star L_k$等于复合

$$[0, N_1 \cdots N_k) \xrightarrow{\mathsf{colex}^{-1}_{(N_1, \dots, N_k)}} [0, N_1) \times \cdots \times [0, N_k) \xrightarrow{\Phi_{L_1} + \cdots + \Phi_{L_k}} \mathbb{Z}.$$

证明。 对每个 $1 \le i \le k$，记 $L_i = S_i : D_i$。第一个断言成立，因为如果

$$X \in [0, S_1 \star \cdots \star S_k)$$

对应于

$$X_1 \star \cdots \star X_k \in [0, S_1) \times \cdots \times [0, S_k)$$

在典范同构 $[0, S_1 \star \cdots \star S_k) \cong [0, S_1) \times \cdots \times [0, S_k)$下，则

$$\begin{aligned} \varphi_{L_1 \star \cdots \star L_k}(X) &= X \cdot (D_1 \star \cdots \star D_k) \\ &= (X_1 \star \cdots \star X_k) \cdot (D_1 \star \cdots \star D_k) \\ &= (X_1 \cdot D_1) + \cdots + (X_k \cdot D_k) \\ &= \varphi_{L_1}(X_1) + \cdots + \varphi_{L_k}(X_k). \end{aligned}$$

对于第二个断言，我们来证明图表

$$\begin{array}{ccc} [0, N_1) \times \cdots \times [0, N_1) & \xrightarrow{\mathsf{colex}^{-1}_{S_1} \times \cdots \times \mathsf{colex}^{-1}_{S_k}} & [0, S_1) \times \cdots \times [0, S_1) \\ \Big\uparrow{\scriptstyle \mathsf{colex}^{-1}_{(N_1, \dots, N_k)}} & & \Big\uparrow{\scriptstyle \cong} \\ [0, N_1 \cdots N_k) & \xrightarrow[\mathsf{colex}^{-1}_{S_1 \star \cdots \star S_k}]{} & [0, S_1 \star \cdots \star S_k) \xrightarrow[\varphi_{L_1 \star \cdots \star L_k}]{} \mathbb{Z} \end{array} \qquad \varphi_{L_1} + \cdots + \varphi_{L_k} : [0, S_1) \times \cdots \times [0, S_k) \to \mathbb{Z}$$

交换。左侧的方形图交换，因为逆字典序同构满足结合律；右侧的三角形图则由第一个断言可知交换。 □

我们可以如下描述串联布局的重要属性。

命题 2.1.3.41. 假设 L_1、.. .、 L_k 都是平坦布局。那么

1. $L_1 \star \cdots \star L_k$ 的秩为

$$\mathsf{rank}(L_1 \star \cdots \star L_k) = \sum_{i=1}^{k} \mathsf{rank}(L_i),$$

2. $L_1 \star \cdots \star L_k$ 的大小为

$$\mathsf{size}(L_1 \star \cdots \star L_k) = \prod_{i=1}^{k} \mathsf{size}(L_i),$$

3. $L_1 \star \cdots \star L_k$ 的余大小为

$$\mathsf{cosize}(L_1 \star \cdots \star L_k) = 1 - k + \sum_{i=1}^{k} \mathsf{cosize}(L_i).$$

证明。 对每个 $1 \le i \le k$，记 $L_i = S_i : D_i$。对于 1，我们计算

$$\mathsf{rank}(L_1 \star \cdots \star L_k) = \mathsf{len}(S_1 \star \cdots \star S_k) = \sum_{i=1}^{k} \mathsf{len}(S_i) = \sum_{i=1}^{k} \mathsf{rank}(L_i).$$

对于 2，我们计算

$$\mathsf{size}(L_1 \star \cdots \star L_k) = \mathsf{size}(S_1 \star \cdots \star S_k) = \prod_{i=1}^{k} \mathsf{size}(S_i) = \prod_{i=1}^{k} \mathsf{size}(L_i).$$

对于 3，我们计算

$$\begin{aligned}
\mathsf{cosize}(L_1 \star \cdots \star L_k) &= 1 + \mathsf{max}(\Phi_{L_1 \star \cdots \star L_k}) \\
&= 1 + \sum_{i=1}^{k} \mathsf{max}(\Phi_{L_i}) \\
&= 1 - k + (1 + \mathsf{max}(\Phi_{L_1})) + \cdots + (1 + \mathsf{max}(\Phi_{L_1})) \\
&= 1 - k + \mathsf{cosize}(L_1) + \cdots + \mathsf{cosize}(L_k).
\end{aligned}$$

其中我们使用了我们对 $\Phi_{L_1 \star \cdots \star L_k}$ 命题2.1.3.40中的识别。 □

2.1.4 平坦合并

我们已经看到，平坦布局 L 的布局函数 Φ_L 是一个重要的不变量。在许多情况下，我们只关心布局函数 Φ_L，并且可以自由地使用任何布局函数为 Φ_L 的布局。平坦合并运算

$$L \mapsto \mathsf{coal}^{\flat}(L)$$

为 我们提供布局函数为 Φ_L 的最简单平坦布局（参见命题 2.1.4.19） . 我们首先定义合并平坦布局的概念。

定义 2.1.4.1. 假设 $L=(s_1,\dots,s_m):(d_1,\dots,d_m)$ 是一个平坦布局。若 L 是合并的，则

1. 对于任意 $1\le i\le m$，我们都有 $s_i\neq 1$;
2. 对于任意 $1\le i<m$，我们都有 $s_id_i\neq d_{i+1}$。

例 2.1.4.2. 平坦布局

$$L=(3,5,2):(7,21,4)$$

由于 $3\cdot 7=21$，因此未合并。

例 2.1.4.3. 平坦布局

$$L=(2,7,6):(1,3,10)$$

已合并。

例 2.1.4.4. 空布局 $E=():()$ 已合并。

例 2.1.4.5. 如果 $L=(s):(d)$ 且 $s\neq 1$，那么 L 是合并的。

例 2.1.4.6. 如果 $L=(s_1,\dots,s_m):(d_1,\dots,d_m)$ 是秩为 >1 的列主序布局，且 $\mathsf{rank}(L)$，那么L 不是合并的，因为对于任意 $1\le i<m$，我们有

$$s_id_i=s_i(s_1\cdots s_{i-1})=s_1\cdots s_i=d_{i+1}.$$

例 2.1.4.7. 如果 $L=(s_1,\dots,s_m):(d_1,\dots,d_m)$ 是一个满足所有$1\le i\le m$都有 $s_i>1$ 的行主序布局，那么 L 是合并的：如果 $1\le i<m$，那么

$$s_id_i=s_is_{i+1}\cdots s_m>s_{i+2}\cdots s_m=d_{i+1}.$$

例 2.1.4.8. 形如以下形式的平坦布局

$$L=(s_1,\dots,s_m):(0,\dots,0)$$

当且仅当 $m\le 1$ 时是合并的。

如果 L 是平坦布局，那么我们可以通过移除满足 $s_i=1$ 的模式，并合并满足 $s_id_i=d_{i+1}$ 的模式，得到一个布局函数与 L 相同的合并布局 $\mathsf{coal}^{\flat}(L)$。更具体地说，我们进行如下构造。

构造 2.1.4.9。 设 L是一个平坦布局，并记为

$$\mathsf{squeeze}(L)=(s_1,\dots,s_m):(d_1,\dots,d_m).$$

令 $\sim$ 为由以下条件生成的<styles id='2'>$m\rangle$ 上的等价关系： $i\sim i+1$ 若

$$s_id_i=d_{i+1}.$$

商集$\langle m\rangle/\sim$ 在 $i\le i'$ 时按照 $[i]\le[i']$ 排序，因此我们可以将 $\langle m\rangle/\sim$ 与 $\langle\bar{m}\rangle$视为相同，其中 $\bar{m}$ 是$\langle m\rangle/\sim$的大小。如果 $i\in\langle\bar{m}\rangle$ 对应于等价类

$$I=\{i',i'+1,\dots,i'+k\}\in\langle m\rangle/\sim,$$

¯那么我们定义整数 $\bar{s}_i$ 和 d_i 为

$$\bar{s}_i = s_{i'} s_{i'+1} \cdots s_{i'+k}$$

and

$$\bar{d_i} = d_{i'},$$

并定义

$$\mathsf{coal}^\flat(L) = (\bar{s}_1, \ldots, \bar{s}_{\bar{m}}) : (\bar{d_1}, \ldots, \bar{d}_{\bar{m}}).$$

观察 2.1.4.10. 考察定义，我们可以等价地将 $\mathsf{coal}^\flat(L)$ 定义为通过以下操作得到的平坦布局：

$$\mathsf{squeeze}(L) = (s_1, \ldots, s_m) : (d_1, \ldots, d_m)$$

通过迭代执行操作

$$s_i, s_{i+1} : d_i, s_i d_i \qquad \rightsquigarrow \qquad s_i s_{i+1} : d_i$$

直到结果合并为止。

例 2.1.4.11. 如果 $L = (2,2,2,2,2) : (8, 16, 1024, 2048, 4096)$，那么

$$\mathsf{coal}^\flat(L) = (4, 8) : (8, 1024).$$

例 2.1.4.12. 如果 $L = (3,4,1,5) : (1, 8, 3, 32)$，那么

$$\mathsf{coal}^\flat(L) = (3, 20) : (1, 8).$$

例 2.1.4.13。 如果 $L = (s_1, \ldots\ldots, s_m) : (d_1, \ldots\ldots, d_m)$ 是列优先的，且并非所有 s_i 都等于 1，则

$$\mathsf{coal}^\flat(L) = (s_1 \cdots s_m) : (1).$$

例 2.1.4.14。 如果 L 是行主序的，则

$$\mathsf{coal}^\flat(L) = \mathsf{squeeze}(L).$$

下面说明操作 $L \mapsto \mathsf{coal}^\flat(L)$ 会得到一个合并布局。

引理 2.1.4.15。 如果 L 是平坦布局，则$\mathsf{coal}^\flat(L)$ 是合并的。

证明。 沿用构造2.1.4.9的记号，令

$$\mathsf{squeeze}(L) = (s_1, \ldots, s_m) : (d_1, \ldots, d_m)$$

并令

$$\mathsf{coal}^\flat(L) = (\bar{s}_1, \ldots, \bar{s}_{\bar{m}}) : (\bar{d_1}, \ldots, \bar{d}_{\bar{m}}).$$

我们想证明 $\mathsf{coal}^\flat(L)$ 是合并的。假设 $1 \le i \le \bar{m}$。那么 i 对应于一个（非空）等价类 $I \in \langle m \rangle / \sim$，并且

$$\bar{s}_i = \prod_{i' \in I} s_{i'}$$

是整数的乘积 $s_{i'} > 1$，所以 $\bar{s}_i > 1$。

¯ ¯假设 $1 \le i < \bar{m}$。我们断言 $\bar{s}_i d_i \neq d_{i+1}$。假设 i 对应于等价类

$$\{i', i'+1, \dots, i'+k\} \in \langle m \rangle / \sim,$$

并假设 $i+1$ 对应于等价类

$$\{i'+k+1, i'+k+2, \dots, i'+k+\ell\} \in \langle m \rangle / \sim .$$

那么利用 $s_{i'+t} d_{i'+t} = d_{i'+t+1}$ 关于 $0 \le t < k$的等式，我们可以写成

$$\begin{aligned} \bar{s}_i \bar{d}_i = \bar{d}_i \bar{s}_i &= d_{i'} s_{i'} s_{i'+1} \cdots s_{i'+k} \\ &= d_{i'+1} s_{i'+1} \cdots s_{i'+k} \\ &\vdots \\ &= d_{i'+k} s_{i'+k} \\ &= s_{i'+k} d_{i'+k} \end{aligned}$$

并且由于 $i'+k$ 和 $i'+k+1$ 不属于同一个等价类，因此有

$$\bar{s}_i \bar{d}_i = s_{i'+k} d_{i'+k} \neq d_{i'+k+1} = \bar{d}_{i+1}.$$

□

例 2.1.4.16. 如果 L 是合并的，那么 $\mathsf{coal}^\flat(L) = L$。特别地，这意味着 $\mathsf{coal}^\flat(-)$ 是一个幂等运算：

$$\mathsf{coal}^\flat(\mathsf{coal}^\flat(L)) = \mathsf{coal}^\flat(L).$$

接下来，我们说明合并平坦布局不会改变布局函数。

引理 2.1.4.17. 如果 L 是平坦布局，那么 $\Phi_{\mathsf{coal}^\flat(L)} = \Phi_L$。

证明。 根据观察 2.1.4.10，只需证明将 $s_i, s_{i+1} : d_i, s_i d_i$ 替换为$s_i s_{i+1} : d_i$ 后，布局函数保持不变。假设

$$L = (s_1, \dots, s_m) : (d_1, \dots, d_m)$$

是一个平坦布局，并且存在某个 $1 \le i < m$ 使得 $d_{i+1} = s_i d_i$。令

$$L' = (s'_1, \dots, s'_{m-1}) : (d'_1, \dots, d'_{m-1})$$

表示由 L 将 L 的第 i 个和第 $(i+1)$ 个模式合并而得到的平坦布局。更准确地说，我们有

$$s'_j = \begin{cases} s_j & j < i \\ s_i s_{i+1} & j = i \\ s_{j+1} & i < j < m, \end{cases} \qquad \text{and} \qquad d'_j = \begin{cases} d_j & j \le i \\ d_{j+1} & i < j < m. \end{cases}$$

L 的布局函数为

$$\Phi_L(x) = x_1 d_1 + \cdots + x_m d_m$$

其中 $x_j = \left\lfloor \dfrac{x}{s_1 \cdots s_{j-1}} \right\rfloor$ 模 s_j，而 L' 的布局函数为

$$\Phi_{L'}(x) = x'_1 d'_1 + \cdots + x'_{m-1} d'_{m-1}$$

其中 $x'_j = \left\lfloor \frac{x}{s'_1 \cdots s'_{j-1}} \right\rfloor$ 模 s'_j。我们观察到

$$x'_j = \begin{cases} x_j & j < i \\ x_i + x_{i+1}s_i & j = i \\ x_{j+1} & i < j < m, \end{cases}$$

因此

$$\begin{aligned} \Phi_L(x) &= x_1d_1 + \cdots + x_md_m \\ &= x_1d_1 + \cdots + x_id_i + x_{i+1}s_id_i + \cdots + x_md_m \\ &= x_1d_1 + \cdots + (x_i + x_{i+1}s_i)d_i + \cdots + x_md_m \\ &= x'_1d'_1 + \cdots + x'_{m-1}d'_{m-1} \\ &= \Phi_{L'}(x). \end{aligned}$$

□

我们可以使用这 个合并操作来刻画两个平坦布局何时具有相同的布局函数 .

命题 2.1.4.18. 假设 A 和 B 是平坦布局。那么

$$\Phi_A = \Phi_B \quad \Leftrightarrow \quad \mathsf{coal}^\flat(A) = \mathsf{coal}^\flat(B).$$

证明。 如果 $\mathsf{coal}^\flat(A) = \mathsf{coal}^\flat(B)$，那么根据引理 2.1.4.17，我们有

$$\Phi_A = \Phi_{\mathsf{coal}^\flat(A)} = \Phi_{\mathsf{coal}^\flat(B)} = \Phi_B.$$

反过来，假设 $\mathsf{coal}^\flat(A) \neq \mathsf{coal}^\flat(B)$。我们将证明 $\Phi_A \neq \Phi_B$。记作

$$\begin{aligned} \mathsf{coal}^\flat(A) &= (s_1, \ldots, s_m) : (d_1, \ldots, d_m), \\ \mathsf{coal}^\flat(B) &= (t_1, \ldots, t_n) : (e_1, \ldots, e_m). \end{aligned}$$

如果 m, n 中一个非零而另一个为 0，那么显然 $\Phi_A \neq \Phi_B$，因此我们可以假设 $m, n \geq 1$。令 i表示满足 $(s_i, d_i) \neq (t_i, e_i)$ 的最小整数。那么，特别地，对于任意 $j < i$，我们有 $s_1 \cdots s_j = t_1 \cdots t_j$。需要考虑两种情形:

- (情形 1)：假设 $d_i \neq e_i$。令 $N = s_1 \cdots s_{i-1} = t_1 \cdots t_{i-1}$。那么

  $$\Phi_{\mathsf{coal}^\flat(A)}(N) = d_i \neq e_i = \Phi_{\mathsf{coal}^\flat(B)}(N)$$

  因此 $\Phi_{\mathsf{coal}^\flat(A)} \neq \Phi_{\mathsf{coal}^\flat(B)}$，从而 $\Phi_A \neq \Phi_B$。

- (情形 2)：假设 $d_i = e_i$，因此 $s_i \neq t_i$。不失一般性，我们可以假设 $s_i < t_i$。令 $N = s_1 \cdots s_i = (t_1 \cdots t_{i-1})s_i$。那么

  $$\Phi_{\mathsf{coal}^\flat(A)}(N) = d_{i+1}$$

  而

  $$\begin{aligned} \Phi_{\mathsf{coal}^\flat(B)}(N) &= s_ie_i \\ &= s_id_i, \end{aligned}$$

  并且由于 $\mathsf{coal}^\flat(A)$ 是合并的，我们有 $d_{i+1} \neq s_id_i$。由此可得 $\Phi_{\mathsf{coal}^\flat(A)} \neq \Phi_{\mathsf{coal}^\flat(B)}$，因此 $\Phi_A \neq \Phi_B$。

□

前一个命题为我们提供了 $\mathsf{coal}^\flat(L)$ 的以下抽象刻画。

命题 2.1.4.19。 如果 L 是一个平坦布局，那么 $\mathsf{coal}^\flat(L)$ 是布局函数为 Φ_L且秩最小的唯一平坦布局。

证明。 假设 L' 是一个平坦布局，且 $\Phi_{L'} = \Phi_L$。那么根据命题 2.1.4.18，我们有

$$\mathsf{coal}^\flat(L) = \mathsf{coal}^\flat(L'),$$

由此可得

$$\mathsf{rank}(\mathsf{coal}^\flat(L)) = \mathsf{rank}(\mathsf{coal}^\flat(L')) \leq \mathsf{rank}(L'),$$

等号成立当且仅当

$$L' = \mathsf{coal}^\flat(L') = \mathsf{coal}^\flat(L).$$

□

2.1.5 紧平坦布局

在讨论布局补之前，我们必须定义一类重要的布局，称为紧致平坦布局。这类布局是布局函数为双射的平坦布局。就表示布局的标准网格图而言，平坦布局 L 是紧致的，当且仅当每个整数 $0 \leq i <$ $\mathsf{size}(L)$恰好出现一次。例如，布局

$A = (3,\ 6) : (1,\ 3)\ =$

0	3	6	9	12	15
1	4	7	10	13	16
2	5	8	11	14	17

是紧致的，而这些布局

$B = (3,6) : (2,6)\ =$

0	6	12	18	24	30
2	8	14	20	26	32
4	10	16	22	28	34

and

$C = (3,\ 6) : (1,\ 2)\ =$

0	2	4	6	8	10
1	3	5	7	9	11
2	4	6	8	10	12

并非紧致。更准确地说，我们有如下定义。

定义 2.1.5.1. 假设 L 是一个平坦布局。若 L 满足以下条件，我们称其为紧致

$$[0, \text{大小}(L)) \xrightarrow{\Phi_L^{\text{余大小}(L)}} [0, \text{余大小}(L))$$

是一个同构。

例 2.1.5.2. 平坦布局

$$L = (2, 2, 2, 2) : (1, 2, 4, 8)$$

是紧致的。更一般地，如果 L 是列优先的，那么 L 是紧致的。

例 2.1.5.3. 平坦布局

$$L = (3, 64, 32) : (2048, 32, 1)$$

是紧致的。更一般地，如果 L 是行主序的，那么 L 是紧致的。

例 2.1.5.4. 空布局

$$E = () : ()$$

是紧致的。

例 2.1.5.5. 假设

$$L = (s_1, \ldots, s_m) : (d_1, \ldots, d_m)$$

是平坦布局。如果 L 存在某个模式，其 $s_i > 1$ 和 $d_i = 0$，那么 L 不是紧致的。

我们可以如下明确刻画紧凑布局。

命题 2.1.5.6。 设 L 是一个平坦布局，并记

$$\mathsf{squeeze}(L) = (s_1, \ldots, s_m) : (d_1, \ldots, d_m).$$

那么 L 是紧致的，当且仅当存在一个置换 $\sigma \in \Sigma_m$，使得

$$d_{\sigma(i)} = s_{\sigma(1)} \cdots s_{\sigma(i-1)}$$

对所有 $1 \le i \le m$ 都成立。换言之，L 是紧致的，当且仅当存在一个置换 $\sigma \in \Sigma_m$，使得 挤压$(L)^{\sigma}$ 是列优先的。

证明。 设 L 是一个平坦布局，并记

$$\mathsf{squeeze}(L) = (s_1, \ldots, s_m) : (d_1, \ldots, d_m).$$

首先，设 L 是紧致的，因此存在一个置换 $\sigma \in \Sigma_m$，使得对每个 $1 \le i \le m$ 都有 $d_{\sigma(i)} = s_{\sigma(1)} \cdots s_{\sigma(i-1)}$。若记 $S^{\sigma} = (s_{\sigma(1)}, \ldots, s_{\sigma(m)})$，则可将 $\Phi_L^{\mathsf{cosize}(L)}$ 写成如下复合

$$[0, \mathsf{size}(L)) \xrightarrow{\mathsf{colex}_S^{-1}} [0, S) \xrightarrow{\cong} [0, S^{\sigma}) \xrightarrow{\mathsf{colex}_{S^{\sigma}}} [0, \mathsf{cosize}(L))$$

$$(x_1, \ldots, x_m) \longmapsto (x_{\sigma(1)}, \ldots, x_{\sigma(m)})$$

由于其中每个映射都是同构，因此复合映射 $\Phi_L^{\mathsf{cosize}(L)}$ 也是同构。

反之，假设 $\Phi_L^{\mathsf{cosize}(L)}$ 是同构。首先，我们注意到步长 d_1、.....、 d_m 必须两两不同：假设 $d_i = d_j$，并令 δ_i^m 和 δ_j^m 分别表示第 i 个（即第 j 个）条目为 1、其余所有条目均为 0 的元组。这些元组满足

$$\delta_i^m \cdot (d_1, \ldots, d_m) = d_i = d_j = \delta_j^m \cdot (d_1, \ldots, d_m),$$

由于 $\Phi_L^{\mathsf{cosize}(L)}$ 是单射，我们必有 $i = j$。鉴于步长 d_1、.....、 d_m 两两不同，令 $\sigma \in \Sigma_m$ 为满足以下条件的置换

$$d_{\sigma(1)} < d_{\sigma(2)} < \cdots < d_{\sigma(m)}.$$

我们将对 $i \geq 1$ 进行归纳法，证明 $d_{\sigma(i)} = s_{\sigma(1)} \cdots s_{\sigma(i-1)}$。对于基例 $i = 1$，注意到 1 在 $\Phi_L^{\mathsf{cosize}(L)}$ 的像中，而 $\Phi_L^{\mathsf{cosize}(L)}$ 的最小非零值为 $d_{\sigma(1)}$，因此可得$d_{\sigma(1)} = 1$。假设 $i > 1$，并且我们已经对所有 $j < i$ 证明了该命题。考虑步幅 $d_{\sigma(i)}$。我们知道不存在形如 $(x_1$、.....、 x_{i-1}、0、.....、$0)^\sigma$ 的 Tuple，使得

$$(x_1, \ldots, x_{i-1}, 0, \ldots, 0)^\sigma \cdot (d_1, \ldots, d_m) = s_{\sigma(1)} \cdots s_{\sigma(i-1)},$$

因为此类表达式的最大可能值为

$$\sum_{j=1}^{i-1} (s_{\sigma(j)} - 1)(s_{\sigma(1)} \cdots s_{\sigma(j-1)}) = s_{\sigma(1)} \cdots s_{\sigma(i-1)} - 1.$$

由于 $\Phi_L^{\mathsf{cosize}}(L)$ 是满射，且 $d_{\sigma(i)} < d_{\sigma(i+1)} < \cdots < d_{\sigma(m)}$，因此 $\Phi_L^{\mathsf{cosize}(L)}$ 的下一个最大值是 $d_{\sigma(i)}$，所以必有 $d_{\sigma(i)} = s_{\sigma(1)} \cdots s_{\sigma(i-1)}$，如所述。 □

我们通过给出一组等价条件来结束本节：平坦布局 L 是紧致的。

命题 2.1.5.7. 设 L 是一个平坦布局。那么下列条件等价。

1. L 是紧致的。*2.*
$\mathsf{coal}^\flat(L)$ 是紧致的。*3.* 挤
压(L) 是紧致的。

4. 排序(L)是紧致的。

证明。 1、2 和 3 等价，这是因为

$$\Phi_L = \Phi_{\mathsf{coal}^\flat(L)} = \Phi_{\mathsf{squeeze}(L)}.$$

接下来要证明，当且仅当 sort(L) 是紧致的， L 才是紧致的。利用以下事实

$$\mathsf{squeeze}(\mathsf{sort}(L)) = \mathsf{sort}(\mathsf{squeeze}(L)),$$

我们有

$$\begin{aligned} \mathsf{sort}(L) \text{ is compact.} \quad &\Leftrightarrow \quad \mathsf{squeeze}(\mathsf{sort}(L)) \text{ is compact.} \\ &\Leftrightarrow \quad \mathsf{sort}(\mathsf{squeeze}(L)) \text{ is compact.} \end{aligned}$$

现在，对于某个置换 $\tau \in \Sigma_m$，有 sort(squeeze(L)) = squeeze(L)$^\tau$，因此存在一个置换 σ，使得当且仅当存在一个置换 $\sigma' \in \Sigma_m$，使 sort(squeeze(L)) 为列优先时，squeeze(L)$^\sigma$ 才是列优先，即 $\sigma' = \tau^{-1}\sigma$。由此可得

$$\begin{aligned} \mathsf{sort}(\mathsf{squeeze}(L)) \text{ is compact.} \Leftrightarrow \quad &\mathsf{squeeze}(L) \text{ is compact.} \\ \Leftrightarrow \quad &L \text{ is compact.} \end{aligned}$$

□

2.1.6 补集

本节中，我们定义互补平坦布局的概念。回顾定义 2.1.5.1 ，平坦布局 L 是紧致的，如果布局函数

$$\Phi_L^{\mathsf{cosize}(L)} : [0, \mathsf{size}(L)) \to [0, \mathsf{cosize}(L))$$

是一个同构。

定义 2.1.6.1. 设 A 和 B 是平坦布局。若串联布局 $A \star B$ 是紧致的，则称 B 是 A的补，并记作 $A \perp B$。

例 2.1.6.2. 若 $A = (3) : (5)$ 且 $B = (5) : (1)$，则 $A \perp B$，因为

$$A \star B = (3, 5) : (5, 1)$$

是紧致的。

$B =$

0	1	2	3	4

$A =$

0
5
10

0	1	2	3	4
5	6	7	8	9
10	11	12	13	14

$= A \star B$

例 2.1.6.3. 若 $A = (4,\ 2,\ 10) : (1400,\ 2,\ 20)$ 且 $B = (2,\ 5,\ 7,\ 2) : (1,\ 4,\ 200,\ 5600)$，则 $A \perp B$，因为

$$A \star B = (4, 2, 10, 2, 5, 7, 2) : (1400, 2, 20, 1, 4, 200, 5600)$$

是紧致的。

例 2.1.6.4. 若 A 是平坦布局，且 $E = () : ()$ 是空布局，则 $A \perp A$ 当且仅当A是紧致的，因为

$$A \star E = A.$$

例 2.1.6.5. 若 A 和 B 是平坦布局，则

$$A \perp B \quad \Leftrightarrow \quad B \perp A.$$

例 2.1.6.6. 若 A 是平坦布局，则 $A \perp A$ 当且仅当 $\mathsf{size}(A) = 1$。

观察 2.1.6.7. 要使 A 存在补，必须满足 Φ_A 是单射。不过，确实存在平坦布局 A，使得 Φ_A 是单射，而 A 不存在补。例如，考虑布局

$$A = (2, 2) : (1, 3).$$

A 的布局函数是单射，因为

$$\Phi_A(0) = 0,\ \Phi_A(1) = 1,\ \Phi_A(2) = 3, \text{ and } \Phi_A(3) = 4,$$

但 A 不存在补：假设

$$B = (s_1, \ldots, s_m) : (d_1, \ldots, d_m)$$

是任何其他平坦布局。如果不存在一个 Tuple

$$(x_1, x_2, y_1, \ldots, y_m) \in [0, 2) \times [0, 2) \times [0, s_1) \times \cdots \times [0, s_m)$$

使得 $\varphi_{A \star B}(x_1,\ x_2,\ y_1,\ \ldots\ldots,\ y_m) = 2$，那么 $A \star B$就不是紧致的。反之，假设存在这样的 Tuple $(x_1,\ x_2,\ y_1,\ \ldots\ldots,\ y_m)$。那么 $\varphi_B(y_1,\ \ldots\ldots,\ y_m) \in \{0,\ 1,\ 2\}$。

- (情形 1)：如果 $\varphi_B(y_1,\ \ldots\ldots,\ y_m) = 0$，那么

$$\varphi_{A \star B}(0, 0, 0, \ldots, 0) = 0 = \varphi_{A \star B}(0, 0, y_1, \ldots, y_m).$$

- (情形 2)：如果 $\varphi_B(y_1,\ \ldots\ldots,\ y_m) = 1$，那么

$$\varphi_{A \star B}(1, 0, 0, \ldots, 0) = 1 = \varphi_{A \star B}(0, 0, y_1, \ldots, y_m).$$

- (情形 3)：如果 $\varphi_B(y_1,\ \ldots\ldots,\ y_m) = 2$，那么

$$\varphi_{A \star B}(0, 1, 0, \ldots, 0) = 3 = \varphi_{A \star B}(1, 0, y_1, \ldots, y_m).$$

无论哪种情形，我们都可以推断出 $\varphi_{A \star B}$ 不是单射，因此 $\Phi_{A \star B}$也不是单射。这意味着 $A \star B$ 不是紧致的，所以 B 不是 A的补。

观察 2.1.6.8. 补集并不唯一。例如，如果

$$A = (8, 8) : (2, 32),$$

那么每个布局

$$\begin{aligned} B_1 &= (2, 2) : (1, 16) \\ B_2 &= (2, 2) : (16, 1) \\ B_3 &= (5, 2, 2, 1) : (256, 1, 16, 0) \end{aligned}$$

都是 A的一个补。相反，存在一个（可能为空的）布局集合

$$\mathsf{complements}^\flat(A) = \{\text{flat layouts } B \mid B \text{ is a complement of } A\}.$$

其中的布局都与 A互为补。

下面给出一组等价条件，用来判定 B 是否为 A 的补（参见命题 2.1.6.10）。为此，我们需要以下技术引理，它描述了串接与运算 挤压$(-)$、排序$(-)$ 和 $\mathsf{coal}^\flat(-)$ 之间的相互作用。

引理 2.1.6.9. 设 A和 B 是平坦布局。那么

1. 挤压$(A \star B)$ $=$ 挤压(A) $\star$ 挤压(B)，
2. 排序$(A \star B)$ $=$ 排序$(L \star$排序$(B))$，以及
3. $\mathsf{coal}^\flat(A \star B)$ $=$ $\mathsf{coal}^\flat(A \star \mathsf{coal}^\flat(B))$。

证明。写

$$A = (s_1, \dots, s_m) : (d_1, \dots, d_m)$$
$$B = (t_1, \dots, t_n) : (e_1, \dots, e_n).$$

如果令 $\{i_1 < \cdots < i_{m'}\} \subset \langle m \rangle$ 表示由 $s_{i_k} \neq 1$ 构成的指标，而 $\{j_1, \dots, j_{n'}\} \subset \langle n \rangle$ 表示由 $t_{j_\ell} \neq 1$ 构成的指标，那么

$$\begin{aligned}\mathsf{squeeze}(A \star B) &= (s_{i_1}, \dots, s_{i_{m'}}, t_{j_1}, \dots, t_{j_{n'}}) : (d_{i_1}, \dots, d_{i_{m'}}, e_{j_1}, \dots, e_{j_{n'}}) \\ &= \mathsf{squeeze}(A) \star \mathsf{squeeze}(B).\end{aligned}$$

这就证明了 1。对于 2，我们注意到，对于任意平坦布局 L 和任意置换 $\sigma \in \Sigma_{\mathsf{len}(L)}$，都有 $\mathsf{sort}(L) = \mathsf{sort}(L^\sigma)$。结论由以下观察得出：

$$A \star \mathsf{sort}(B) = (A \star B)^\sigma$$

其中 σ 是如下形式的块排列 $\sigma = \mathsf{id} \times \sigma' \in \Sigma_m \times \Sigma_n \subset \Sigma_{m+n}$。对于 3，只需证明 $A \star B$ 和 $A \star \mathsf{coal}^\flat(B)$ 具有相同的布局函数。这由命题 2.1.3.40 得出。 □

Prop命题 2.1.6.10。 设 A 和 B 是平坦布局。则以下条件等价 t.

1. $A \perp B$。 2. $B \perp A$。 3. $A \perp$ 挤压(B)。 4. $A \perp \mathsf{coal}^\flat(B)$。 5. $A \perp$ 排序(B)。

证明。 我们使用命题 2.1.5.7和引理 2.1.6.9 来证明这些条件等价。首先，我们注意到 排序$(A \star B) =$ 排序$(B \star A)$，这意味着条件 1 和 2 等价。接下来，根据引理 2.1.6.9，若 $\mathsf{op}(-)$ 是以下任一运算： 挤压$(-)$、排序$(-)$ 或 $\mathsf{coal}^\flat(-)$，则

$$\mathsf{op}(A \star B) = \mathsf{op}(A \star \mathsf{op}(B)),$$

因此

$$\begin{aligned} A \perp B \quad &\Leftrightarrow \quad A \star B \text{ is compact.} \\ &\Leftrightarrow \quad \mathsf{op}(A \star B) \text{ is compact.} \\ &\Leftrightarrow \quad \mathsf{op}(A \star \mathsf{op}(B)) \text{ is compact.} \\ &\Leftrightarrow \quad \mathsf{op}(B) \text{ is a complement of } A. \end{aligned}$$

□

我们希望刻画平坦布局何时存在补。为此，我们作出如下定义。

定义 2.1.6.11. 假设 A 是一个平坦布局，并记

$$\mathsf{sort}(\mathsf{squeeze}(A)) = (s_1, \dots, s_m) : (d_1, \dots, d_m).$$

我们称 A 是可补的，如果对于任意 $1 \leq i < m$，整数 $s_i d_i$ 整除 d_{i+1}。

例 2.1.6.12. 平坦布局

$$A_1 = (4,1,1,4,4) : (64,0,0,1,8)$$

是可补的，而平坦布局

$$A_2 = (4,4,4) : (64,1,1)$$

是不可补的。

例 2.1.6.13. 平坦布局

$$A_1 = (10,2) : (4,80)$$

可补的，而平坦布局

$$A_2 = (10,2) : (80,4)$$

是不可补的。

Ex例 2.1.6.14. 如果 A 是紧致的，那么根据命题 2.1.5.6， A 是可补的。

例 2.1.6.15. 假设

$$A = (s_1, \ldots, s_m) : (d_1, \ldots, d_m)$$

是一个平坦布局 。如果存在任何 $1 \le i \le m$ ，使得 $s_i \neq 1$ 且 $d_i = 0$，那么 A 不可补 .

如果 A 可补，那么我们可以按如下方式构造 A 的一个补。

构造 2.1.6.16. 假设 A是一个平坦布局，并记

$$\mathsf{sort}(\mathsf{squeeze}(A)) = (s_1, \ldots, s_m) : (d_1, \ldots, d_m).$$

如果 A 可补，那么我们定义平坦布局 $\mathsf{comp}^\flat(A)$ 如下

$$\mathsf{comp}^\flat(A) = \mathsf{coal}^\flat(C)$$

其中

$$C = \left(d_1, \frac{d_2}{s_1 d_1}, \frac{d_3}{s_2 d_2}, \ldots, \frac{d_m}{s_{m-1} d_{m-1}}\right) : \left(1, s_1 d_1, s_2 d_2, \ldots, s_{m-1} d_{m-1}\right).$$

例 2.1.6.17. 如果 $A = (8,8) : (1,8)$，那么

$$\mathsf{comp}^\flat(A) = () : ()$$

是空 布局。更一般地，如果 A 是紧致的，那么 $\mathsf{comp}^\flat(A) = () : ()$ 是空布局 .

例 2.1.6.18. 如果 $A = (2,2) : (2,8)$，那么

$$\mathsf{comp}^\flat(A) = (2,2) : (1,4).$$

例 2.1.6.19. 如果 $A = (3,3,8) : (16,96,1)$，那么

$$\mathsf{comp}^\flat(A) = (2,2) : (8,48).$$

让我们证明，$\mathsf{comp}^\flat(A)$ 确实是 A 的补。

引理 2.1.6.20。 假设 A 是一个平坦布局。如果 A 是可补的，那么

$$A \perp \mathsf{comp}^\flat(A).$$

证明。 令我们写为

$$\mathsf{sort}(\mathsf{squeeze}(A)) = (s_1, \dots, s_m) : (d_1, \dots, d_m),$$

使得 $\mathsf{comp}^\flat(A) = \mathsf{coal}^\flat(C)$，其中

$$C = \left(d_1, \frac{d_2}{s_1 d_1}, \frac{d_3}{s_2 d_2}, \dots, \frac{d_m}{s_{m-1} d_{m-1}}\right) : \left(1, s_1 d_1, s_2 d_2, \dots, s_{m-1} d_{m-1}\right).$$

根据命题 2.1.6.10，只需证明 C 是 排序(挤压(A)) 的补。这是因为串接

$$\mathsf{sort}(\mathsf{squeeze}(A)) \star C$$

等于

$$\left(s_1, \dots, s_m, d_1, \frac{d_2}{s_1 d_1}, \dots, \frac{d_m}{s_{m-1} d_{m-1}}\right) : (d_1, \dots, d_m, 1, s_1 d_1, \dots, s_{m-1} d_{m-1}),$$

且其排序等于

$$\left(d_1, s_1, \frac{d_2}{s_1 d_1}, \dots, \frac{d_m}{s_{m-1} d_{m-1}}, s_m\right) : (1, d_1, s_1 d_1, \dots, s_{m-1} d_{m-1}, d_m)$$

这是列优先的。 □

我们已经证明，如果 A 是可补的，那么 A 存在一个补。接下来，我们证明反过来也成立。

命题 2.1.6.21. 设 A 是平坦布局。那么，存在 A 的一个补 B，当且仅当 A 是可补的。

证明。 如果 A 是可补的，那么根据引理 2.1.6.20，布局 $B = \mathsf{comp}^\flat(A)$ 是A的一个补。反过来，设存在一个补 B，它是 A的补，并考虑平坦布局

$$\begin{aligned} L &= \mathsf{sort}\big(\mathsf{squeeze}(A) \star \mathsf{squeeze}(B)\big) \\ &= (s_1, \dots, s_m) : (d_1, \dots, d_n). \end{aligned}$$

由于 $\Phi_L(0) = 0$，且 Φ_L 是单射，因此我们知道 $d_1 \neq 0$。我们将证明 $d_i = s_1 \cdot \cdots s_{i-1}$，也就是说，$L$ 是列优先的。由于

$$\Phi_L^{\mathsf{cosize}(L)} : [0, \mathsf{size}(L)) \to [0, \mathsf{cosize}(L))$$

是一个双射，因此我们知道 1 在 Φ_L的像中，这意味着 $d_1 = 1$。设 $1 < i \leq m$，并假设我们已经证明，对于所有 $j < i$，都有 $d_j = s_1 \cdot \cdots s_{j-1}$。考虑步幅 d_i。我们知道，不存在这样的 $(x_1, \dots, x_{i-1}, 0, \dots, 0)$，使得 $(x_1, \dots, x_{i-1}, 0, \dots, 0) \cdot (d_1, \dots, d_m) = s_1 \cdot \cdots s_{i-1}$，因为这种表达式可能取得的最大值是

$$\sum_{j=1}^{i-1} (s_j - 1)(s_1 \cdots s_{j-1}) = s_1 \cdots s_i - 1.$$

由于 Φ_L 是满射，且 $d_i \leq d_{i+1} \leq \cdots \leq d_m$，因此 Φ_L 的下一个最大值是 d_i，所以必有 $d_i = s_1 \cdot \cdots s_{i-1}$，如前所述。

回到我们的主要目标，考虑布局

$$\mathsf{sort}(\mathsf{squeeze}(A)) = (s'_1, \ldots, s'_{m'}) : (d'_1, \ldots, d'_{m'}).$$

那么存在 $j_1 < \cdots < j_{m'}$，使得 $s'_i = s_{j_i}$且对每个 $1 \leq i \leq m'$都有 $d'_i = d_{j_i}$。如果 $1 \leq i < m'$，那么

$$s'_i d'_i = s_{j_i} d_{j_i} = s_{j_i} s_1 \cdots s_{j_i - 1}$$

整除

$$d'_{i+1} = s_1 \cdots s_{j_{i+1}-1},$$

因此我们得出结论： A 是可补的。 □

我们的下一个目标是给出平坦布局 A的补 $\mathsf{comp}^\flat(A)$ 的抽象刻画。为此，我们需要以下引理。

引理 2.1.6.22. 设 A 是一个平坦布局。如果 A 是可补的且已排序，那么布局函数

$$\Phi_A : [0, \mathsf{size}(A)) \to \mathbb{Z}$$

是递增的。

证明。写

$$A = (s_1, \ldots, s_m) : (d_1, \ldots, d_m).$$

若 $1 \leq k \leq m$，我们断言

$$d_1(s_1 - 1) + d_2(s_2 - 1) + \cdots + d_{k-1}(s_{k-1} - 1) \leq d_k.$$

若 $k = 1$，则此命题显然成立，并由关于 k 的归纳法可得

$$\begin{aligned} d_1(s_1 - 1) + \cdots + d_{k-2}(s_{k-2} - 1) + d_{k-1}(s_{k-1} - 1) &\leq d_{k-1} + d_{k-1}(s_k - 1) \\ &= d_{k-1} s_{k-1} \\ &\leq d_k. \end{aligned}$$

现在，假设我们有 x， $y \in [0, \text{大小}(A))$，且 $x \leq y$。在 colexico- 图同构下，这些整数对应于元组。

$$(x_1, \ldots, x_m), (y_1, \ldots, y_m) \in [0, s_1) \times \cdots \times [0, s_m)$$

由于 $x \leq y$，我们知道存在某个极大的 $1 \leq k \leq m$ ，使得 $x_k < y_k$，并且对于所有 $k < \ell \leq m$，$x_\ell = y_\ell$ 。现在我们可以计算

$$\begin{aligned} \Phi_A(x) &= d_1 x_1 + \cdots + d_{k-1} x_{k-1} + d_k x_k + d_{k+1} x_{k+1} + \cdots + d_m x_m \\ &= d_1 x_1 + \cdots + d_{k-1} x_{k-1} + d_k x_k + d_{k+1} y_{k+1} + \cdots + d_m y_m \\ &\leq d_1(s_1 - 1) + \cdots + d_{k-1}(s_{k-1} - 1) + d_k x_k + d_{k+1} y_{k+1} + \cdots + d_m y_m \\ &\leq d_k + d_k x_k + d_{k+1} y_{k+1} + \cdots + d_m y_m \\ &= d_k(x_k + 1) + d_{k+1} y_{k+1} + \cdots + d_m y_m \\ &\leq d_k y_k + d_{k+1} y_{k+1} + \cdots + d_m y_m \\ &\leq d_1 y_1 + \ldots d_m y_m \\ &= \Phi_A(y). \end{aligned}$$

□

命题 2.1.6.23. 假设 A 和 B 是平坦布局。如果

1. $A \perp B$, *2.* 大小$(B) =$ 大小$(\mathsf{comp}^\flat(A))$, *3.* B 是合并的，*4.* B 已排序，

那么 $B = \mathsf{comp}^\flat(A)$。

证明。 条件 1 和 2 意味着 Φ_B 和 $\Phi_{\mathsf{comp}^\flat(A)}$ 具有相同的像。由于 B 和$\mathsf{comp}^\flat(A)$ 已排序，根据引理 2.1.6.22 ，我们知道 Φ_B 和$\Phi_{\mathsf{comp}^\flat(A)}$ 是递增的。结合这两个事实，可得 $\Phi_B = \Phi_{\mathsf{comp}^\flat(A)}$。命题 2.1.4.18 和条件 3 随即意味着

$$B = \mathsf{coal}^\flat(B) = \mathsf{coal}^\flat(\mathsf{comp}^\flat(A)) = \mathsf{comp}^\flat(A).$$

□

定义 2.1.6.24。 设 A 和 B 是平坦布局，且 N 是一个正整数。我们称 B 是 A 的N-补，如果 B 是 A 的补，并且

$$\mathsf{size}(A) \cdot \mathsf{size}(B) = N.$$

定义 2.1.6.25。 设 A 是一个平坦布局，并记

$$\mathsf{sort}(\mathsf{squeeze}(A)) = (s_1, \ldots, s_m) : (d_1, \ldots, d_m).$$

我们称 A 是 N-可补的，当且仅当

1. 对所有 $1 \leq i < m$，整数 $s_i d_i$ 整除 d_{i+1}，且
2. 整数 $s_m d_m$ 整除 N。

观察 2.1.6.26. 如果 A 可补，且 $s_m : d_m$ 是布局排序 $\mathsf{squeeze}(A)$ 中的最后一个模式，那么当且仅当 N 是 $s_m d_m$的正整数倍时， A 才是 N-可补的。

观察 2.1.6.27. N-补并不唯一。例如，如果 $A = (2, 2) : (1, 50)$ 且 $N = 100$，那么布局 $B_1 = (25) : (2)$ 和 $B_2 = (5, 5) : (2, 10)$ 都是 A的 N-补。更一般地，如果 B是 A的 N-补，那么 $\mathsf{coal}^\flat(B)$ 也是 A 的 N-补。

注记 2.1.6.28. 假设 A 是一个平坦布局，且 B_1 和 B_2 是 A的 N-补。那么布局函数 Φ_{B_1} 和 Φ_{B_2} 不必相等，但它们必然具有相同的像。例如，若$A = (4) : (63)$，且 $N = 252$，则 $B_1 = (7, 9) : (1, 7)$ 和 $B_2 = (9, 7) : (7, 1)$ 是 A的 N-补，且 $\Phi_{B_1} \neq \Phi_{B_2}$，因为

$$\Phi_{B_1}(1) = 1 \neq 7 = \Phi_{B_2}(1).$$

作为一个更一般的例子，若 B 是 A的 N-补，那么 $\mathsf{sort}(B)$ 也是 A的 N-补。

构造 2.1.6.29. 假设 A 是一个平坦布局， N 是一个正整数，且 A 是可 N-补的。若我们写作

$$\mathsf{sort}(\mathsf{squeeze}(A)) = (s_1, \ldots, s_m) : (d_1, \ldots, d_m),$$

则我们定义平坦布局 $\mathsf{comp}^\flat(A,N)$如下：

$$\mathsf{comp}^\flat(A,N) = \mathsf{coal}^\flat(C)$$

其中

$$C = \left(d_1, \frac{d_2}{s_1 d_1}, \frac{d_3}{s_2 d_2}, \ldots, \frac{N}{s_m d_m}\right) : \left(1, s_1 d_1, s_2 d_2, \ldots, s_m d_m\right).$$

例 2.1.6.30. 若 $A = (3,10) : (80,4)$ 且 $N = 2400$，则

$$\mathsf{comp}^\flat(A,N) = (4,2,10) : (1,40,240).$$

引理 2.1.6.31. 假设 A 是一个平坦布局，N 是一个正整数，且 A 是 N-可补的。那么 $\mathsf{comp}^\flat(A,N)$ 是 A的一个 N-补。

证明。 我们写作

$$\mathsf{sort}(\mathsf{squeeze}(A)) = (s_1, \ldots, s_m) : (d_1, \ldots, d_m),$$

使得 $\mathsf{comp}^\flat(A,N) = \mathsf{coal}^\flat(C)$，其中

$$C = \left(d_1, \frac{d_2}{s_1 d_1}, \frac{d_3}{s_2 d_2}, \ldots, \frac{N}{s_m d_m}\right) : (1, s_1 d_1, s_2 d_2, \ldots, s_m d_m)\,.$$

首先，我们计算

$$\begin{aligned}
\mathsf{size}(A) \cdot \mathsf{size}(B) &= \left(\prod_{i=1}^{m} s_i\right) \cdot \left(d_1 \cdot \left(\prod_{i=2}^{m} \frac{d_i}{s_{i-1} d_{i-1}}\right) \cdot \frac{N}{s_m d_m}\right) \\
&= \frac{\left(\prod_{i=1}^{m} s_i\right)\left(\prod_{i=1}^{m} d_i\right)}{\left(\prod_{i=1}^{m} s_i d_i\right)} \cdot N \\
&= N.
\end{aligned}$$

我们需要验证 $A \star B$ 是紧致的。等价地，我们需要验证 $\Phi^N_{A\star B}$ 是一个同构。根据引理 2.1.5.6，只需证明

$$\mathsf{squeeze}(A) \star \mathsf{squeeze}(B)$$

是紧致的。这是因为该布局等于

$$\left(s_1, \ldots, s_m, d_1, \frac{d_2}{s_1 d_1}, \ldots, \frac{N}{s_m d_m}\right) : (d_1, \ldots, d_m, 1, s_1 d_1, \ldots, s_m d_m)$$

因此，其排序

$$\mathsf{sort}(\mathsf{squeeze}(A) \star \mathsf{squeeze}(B))$$

等于

$$\left(d_1, s_1, \frac{d_2}{s_1 d_1}, \ldots, \frac{d_m}{s_{m-1} d_{m-1}}, s_m, \frac{N}{s_m d_m}\right) : (1, d_1, s_1 d_1, \ldots, s_{m-1} d_{m-1}, d_m, s_m d_m)$$

它是列优先的。 □

命题 2.1.6.32. 假设 A 是平坦布局，且 N 是正整数。那么，当且仅当 A是 N-可补的时，才存在 A的一个N-补 B。

证明。 如果 A是 N-可补的，那么根据引理 2.1.6.31，布局 $B = \text{补}^\flat(L, N)$ 是 N- 的补布局 A。

另一 方面，假设存在 N-补 B，它是 A 的补。考虑平坦布局

$$\begin{aligned} L &\coloneqq \mathsf{sort}\big(\mathsf{squeeze}(A) \star \mathsf{squeeze}(B)\big) \\ &= (s_1, \ldots, s_m) : (d_1, \ldots, d_n). \end{aligned}$$

由于 $\Phi_L(0) = 0$，且 Φ_L是单射，我们知道 $d_1 \neq 0$。我们将论证 $d_i = s_1 \cdots s_{i-1}$，也就是说， L 是列优先的。由于

$$\Phi_L^N : [0, N) \to [0, N)$$

是双射，我们知道 1 属于 Φ_L 的像，这意味着 $d_1 = 1$。假设 $1 < i \leq m$，并且假设我们已证明对于所有 $j < i$， $d_j = s_1 \cdots s_{j-1}$。考虑步幅 d_i。我们知道不存在这样的 $(x_1, \ldots, x_{i-1}, 0, \ldots, 0)$，使得 $(x_1, \ldots, x_{i-1}, 0, \ldots, 0) \cdot (d_1, \ldots, d_m) = s_1 \cdots s_{i-1}$，因为这种表达式可能取得的最大值是

$$\sum_{j=1}^{i-1} (s_j - 1)(s_1 \cdots s_{j-1}) = s_1 \cdots s_i - 1.$$

由于 Φ_L 是满射，且 $d_i \leq d_{i+1} \leq \cdots \leq d_m$，因此 Φ_L 的下一个最大值是 d_i，所以必有 $d_i = s_1 \cdots s_{i-1}$，如所述。

回到我们的主要目标，考虑布局

$$\mathsf{sort}(\mathsf{squeeze}(A)) = (s'_1, \ldots, s'_{m'}) : (d'_1, \ldots, d'_{m'}).$$

那么存在 $j_1 < \cdots < j_{m'}$，使得 $s'_i = s_{j_i}$，并且对每个 $1 \leq i \leq m'$ 都有 $d'_i = d_{j_i}$。若 $1 \leq i < m'$，则

$$s'_i d'_i = s_{j_i} d_{j_i} = s_{j_i} s_1 \cdots s_{j_i - 1}$$

整除

$$d'_{i+1} = s_1 \cdots s_{j_{i+1} - 1}.$$

若 $i = m'$，则

$$s'_{m'} d'_{m'} = s_{j_{m'}} d_{j_{m'}} = s_{j_{m'}} s_1 \cdots s_{j_{m'} - 1}$$

整除

$$N = s_1 \cdots s_m.$$

我们得出， A 是 N-可补的。 □

命题 2.1.6.33。 假设 N 是正整数，且 A 是 N-可补的平坦布局。如果B 是一个平坦布局，使得

1. B 是 L的 N-补，*2.* B 是合并的，且 *3.* B 已排序，

那么 $B = \text{comp}^\flat(A, N)$。

证明。 条件 1 和 2 意味着 Φ_B 和 $\Phi_{\mathsf{comp}^\flat(A,\ N)}$ 具有相同的像。由于 B 和 $\mathsf{comp}^\flat(A,\ N)$ 已排序，根据引理 2.1.6.22 可知 Φ_B 和 $\Phi_{\mathsf{comp}^\flat(A,\ N)}$ 是递增的。结合这两个事实，可得 $\Phi_B = \Phi_{\mathsf{comp}^\flat(A,\ N)}$。命题 2.1.4.18 以及条件 3 意味着

$$B = \mathsf{coal}^\flat(B) = \mathsf{coal}^\flat(\mathsf{comp}^\flat(A, N)) = \mathsf{comp}^\flat(A, N).$$

□

引理 2.1.6.34. 假设 A 是平坦布局。若 $N_1 \leq N_2$ 是正整数，且 A 是N_1-可补的， A 是 N_2-可补的，则

$$\Phi_{\mathsf{comp}^\flat(A,N_2)}\,|_{[0,N_1)} = \Phi_{\mathsf{comp}^\flat(A,N_1)}.$$

证明。令

$$\mathsf{sort}(\mathsf{squeeze}(A)) = (s_1, \ldots, s_m) : (d_1, \ldots, d_m),$$
$$C = \left(d_1, \frac{d_2}{s_1 d_1}, \frac{d_3}{s_2 d_2}, \ldots, \frac{d_m}{s_{m-1} d_{m-1}}\right) : \left(1, s_1 d_1, s_2 d_2, \ldots, s_{m-1} d_{m-1}\right)$$

并令

$$E_1 = \left(\frac{N_1}{s_m d_m}\right) : (s_m d_m),$$
$$E_2 = \left(\frac{N_2}{s_m d_m}\right) : (s_m d_m),$$
$$C_1 = C \star E_1,$$
$$C_2 = C \star E_2,$$

使得

$$\mathsf{comp}^\flat(A) = \mathsf{coal}^\flat(C)$$
$$\mathsf{comp}^\flat(A, N_1) = \mathsf{coal}^\flat(C_1)$$
$$\mathsf{comp}^\flat(A, N_2) = \mathsf{coal}^\flat(C_2).$$

那么我们有一个交换图

$$\begin{array}{ccccccc}
[0, \mathsf{size}(C_1)) & \xrightarrow{\mathrm{colex}^{-1}_{(\mathsf{size}(C), N_1)}} & [0, \mathsf{size}(C)) \times [0, N_1) & \xrightarrow{\Phi_C \times s_m d_m} & \mathbb{Z} \times \mathbb{Z} & \xrightarrow{+} & \mathbb{Z} \\
\downarrow{\scriptstyle \subseteq} & & \downarrow{\scriptstyle \mathsf{id} \times \subseteq} & & \downarrow{\scriptstyle \mathsf{id}} & & \downarrow{\scriptstyle \mathsf{id}} \\
[0, \mathsf{size}(C_2)) & \xrightarrow{\mathrm{colex}^{-1}_{(\mathsf{size}(C), N_2)}} & [0, \mathsf{size}(C)) \times [0, N_2) & \xrightarrow{\Phi_C \times s_m d_m} & \mathbb{Z} \times \mathbb{Z} & \xrightarrow{+} & \mathbb{Z}
\end{array}$$

其中，根据命题 2.1.3.40，顶行的复合是 $C_1 = C \star E_1$的布局函数，底行的复合是 $C_2 = C \star E_2$的布局函数。这说明 Φ_{C_2} 在 [0,大小(C_2)) 上的限制是 Φ_{C_1}，结论由以下事实可得：

$$\Phi_{\mathsf{comp}^\flat(A,N_1)} = \Phi_{C_1}$$
$$\Phi_{\mathsf{comp}^\flat(A,N_2)} = \Phi_{C_2}.$$

□

2.1.7 更多运算

本节定义平坦布局上的若干更多运算，即复合、扁平除法和扁平积。这些是对（嵌套）布局上更自然运算的平坦化变体。我们不常使用这些运算，但为了完整起见，仍将其列出。

2.1.7.1 复合

如果 A 和 B 是平坦布局，那么复合布局 $B \circ A$ 也是一个平坦布局，其布局函数是 A 和 B的布局函数的复合。更准确地说，我们有如下定义。

定义 2.1.7.1. 假设 A 和 B 是平坦布局。我们称平坦布局 C 为A 和 B的**复合**，并记作 $C = B \circ A$，如果

1. C 是非退化的，2.
shape(A) = shape(R),
3. $\Phi_R = \Phi_B \circ \Phi_A^{\mathsf{size}(B)}$。

注记 2.1.7.2. 请注意，定义中的条件 2 确保 Φ_R 和 Φ_A 具有相同的定义域，而条件 3 意味着 cosize(A) $\leq$ size(B)。

例 2.1.7.3。 如果 A = (2，3) : (5，6) 且 B = (80) : (10)，那么

$$B \circ A = (2, 3) : (50, 60).$$

更一般地，如果

$$A = (s_1, \ldots, s_m) : (d_1, \ldots, d_m)$$

是一个非退化的平坦布局，并且

$$B = (t) : (e)$$

是一个秩 1 的平坦布局，且 $t \geq$ 余大小(A)，那么 A 和 B 可复合，并且

$$B \circ A = (s_1, \ldots, s_m) : (td_1, \ldots, td_m).$$

例 2.1.7.4。 如果 A = (128，128) : (0，0) 且 B = (64，32) : (1，64)，那么

$$B \circ A = (128, 128) : (0, 0).$$

更一般地，如果 A 是一个平坦布局，且其每个步幅条目都为零，并且 B 是任意平坦布局，那么A 和 B 可与 $B \circ A = A$复合。

例 2.1.7.5. 如果 A = (64，32) : (2，256) 且 B = (2048，2048) : (1，2048)，那么

$$B \circ A = (64, 32) : (2, 256).$$

更一般地，如果 A 是任意平坦布局，且 B 是一个余大小(A) $\leq$ 大小(B)的列主序平坦布局，那么 $B \circ A = A$。

例 2.1.7.6. 如果 A = (4) : (2) 且 B = (2，2，6) : (12，6，1)，那么不存在满足 R with$R = B \circ A$ 的平坦布局。

注记 2.1.7.7。 如果 B' 和 B 具有相同的布局函数，那么 $B \circ A = B' \circ A$。

注记 2.1.7.8。 扁平布局是更一般的 布局 (定义 2.3.1.1) 的一种特殊情况。事实证明，在某些情况下（例如例 2.1.7.6），不存在满足 C与 $C = B \circ A$的扁平布局，但存在满足 C 与 $C = B \circ A$ 的（嵌套）布局（见例 2.3.7.6）。因此，我们将推迟对复合的进一步讨论和分析，直到定义了具有完整一般性的布局。

2.1.7.2 扁平除法

如果 A 和 B 都是扁平布局，那么 A 除以 B 的扁平除法是更自然的布局逻辑除法 的扁平化版本。详情请参见第2.3.8 节。

定义 2。 **1.7.9.** 设 A 和 B 是平坦布局，且 B 是 size(A)-可补的，其中

$$B^c = \mathsf{comp}^\flat(B, \mathsf{size}(A)).$$

我们将 A 除以 B 的扁平除法定义为平坦布局

$$A \oslash^\flat B = A \circ (B \star B^c).$$

例 2.1.7.10. 若 $A = (2,2,2,2) : (1,4,2,8)$，且 $B = (2,2) : (4,2)$，则

$$A \oslash^\flat B = (2,2,2,2) : (4,2,1,8).$$

例 2.1.7.11. 若 $A = (3,5,9,6) : (54,0,6,1)$，且 $B = (6,3) : (135,1)$，则

$$A \oslash^\flat B = (6,3,5,9) : (1,54,0,6).$$

E例 2.1.7.12. 如果 A 是任意平坦布局，且 $B = () : ()$ 是空布局，那么

$$A \oslash^\flat B = A.$$

2.1.7.3 扁平积

如果 A 和 B 都是平坦布局，那么它们的平坦积 $A \otimes^\flat B$ 是 A 和 B 的更自然的布局逻辑乘积的扁平化版本。详情请参阅第2.3.9 节。

定义 2.1.7.13. 设 A 和 B 是平坦布局，且 A 是 size(A)·cosize(B)-可补的，其中

$$A^c = \mathsf{comp}^\flat(A, \mathsf{size}(A) \cdot \mathsf{cosize}(B)).$$

我们将 A 和 B 的平坦积定义为

$$A \otimes^\flat B = A \star (A^c \circ B).$$

例 2.1.7.14. 如果 $A = (2,2,2) : (1,2,4)$ 且 $B = (2,2,2) : (1,2,4)$，那么

$$A \otimes^\flat B = (2,2,2,2,2,2) : (1,2,4,8,16,32).$$

例 2.1.7.15. 如果 $A = (2,2,2) : (1,2,4)$，且 $B = (3,5) : (5,1)$，那么

$$A \otimes^\flat B = (2,2,2,3,5) : (1,2,4,40,8).$$

例 2.1.7.16. 如果 A 是任意平坦布局，且 $B = () : ()$ 是空布局，那么

$$A \otimes^\flat B = A.$$

2.1.8 可处理平坦布局

本节将定义一类性质特别良好的平坦布局，称为可处理的平坦布局。可处理的平坦布局涵盖了最重要的布局示例，例如行主序、列主序、紧致布局和可补布局。稍后我们会看到，可处理的平坦布局恰好是从某个范畴**Tuple**中产生的布局。

定义 2.1.8.1. 设 L 是一个平坦布局，并记为

$$\mathsf{sort}(L) = (s_1, \ldots, s_m) : (d_1, \ldots, d_m).$$

若对每个 $1 \leq i < m$，都有下式成立，则称 L 是可处理的

1. $d_i = 0$，或 2.
   $s_i d_i$ 整除 d_{i+1}。

例 2.1.8.2. 平坦布局

$$L = (12) : (17)$$

是可处理的。更一般地，任何秩 1 的平坦布局都是可处理的。

例 2.1.8.3. 平坦布局

$$L = (2, 4, 32) : (1, 2, 8)$$

是可处理的。更一般地，任何列主序布局

$$L = (s_1, \ldots, s_m) : (1, s_1, \ldots, s_1 \cdots s_{m-1})$$

都是可处理的。

例 2.1.8.4. 平坦布局

$$L = (2, 4, 32) : (128, 32, 1)$$

是可处理的。更一般地，任何行主序布局

$$L = (s_1, \ldots, s_m) : (s_2 \cdots s_m, \ldots, s_m, 1)$$

都是可处理的。

例 2.1.8.5. 平坦布局

$$L = (3, 3, 1, 3, 3, 1, 3) : (81, 1, 0, 9, 3, 0, 27)$$

是可处理的。更一般地，任何紧致平坦布局都是可处理的。

例 2.1.8.6. 平坦布局

$$L = (3, 7, 7) : (0, 15, 0)$$

is t可处理。更一般地，任何恰好只有一个非零步幅的平坦布局都是可处理的。

例 2.1.8.7. 平坦布局

$$L = (2, 2, 2, 2) : (1, 2048, 16, 64)$$

是可处理的。更一般地，任何可补的平坦布局都是可处理的。

例 2.1.8.8. 设 L 是一个平坦布局。若 L 是可处理的，且 $I \subset \langle m \rangle$ 是任意子集，则限制 $L\,|_I$ 是可处理的。特别地，若 L 是可处理的，则 squeeze(L) 和 filter(L) 都是可处理的。

例 2.1.8.9. 该平坦布局

$$L = (4, 8) : (3, 3)$$

不可处理。特别地，这表明可处理平坦布局 L_1 与 L_2 的串接 $L_1 \star L_2$ 不一定可处理。

观察 2.1.8.10. 若 L 是可处理的平坦布局，且 stride(L) 的任何条目都不等于 0，则 L 可补。特别地，若 L 可处理，则 filter(L) 可补。

我们在本节最后列出平坦布局 L 可处理的一组等价条件。

命题 2.1.8.11. 设 L 是一个平坦布局。则下列条件等价。

1. L 可处理。 *2.* 排序(L) 可处理。 *3.* 过滤(L) 可处理。
4. 过滤(L) 可补。

证明。 假设 L 是一个平坦布局。

- **(**$1 \Leftrightarrow 2$**):** 这是由以下事实得出的：

$$\mathsf{sort}(\mathsf{sort}(L)) = \mathsf{sort}(L).$$

- **(**$1 \Leftrightarrow 3$**):** 这是由以下事实得出的：

$$\mathsf{sort}(\mathsf{filter}(L)) = \mathsf{filter}(\mathsf{sort}(L)).$$

- **(**$3 \Leftrightarrow 4$**):** 这是由以下事实得出的：如果

$$L = (s_1, \ldots, s_m) : (d_1, \ldots, d_m)$$

是一个平坦布局，且其中每个步幅条目 d_i 均非零，那么可处理性的定义就与可补性的定义一致。

□

2.2 嵌套元组

本节中，我们将介绍嵌套元组，它是定义具有完全一般性的布局所需的元组推广形式。

2.2.1 概形

嵌套元组 S 由其扁平化和其轮廓确定。前者是一个普通元组，后者描述了 S上的括号化模式。我们将轮廓精确定义如下。

定义 2.2.1.1. 一个轮廓 P 要么是

1. $P = *$，要么是
2. 一个元组 $P = (P_1, \ldots\ldots, P_r)$，其元素为轮廓 $P_1, \ldots\ldots, P_r$，其中 $r \geq 0$。

我们用 Profile 表示轮廓的集合。

例 2.2.1.2. 下面是一些轮廓的示例。

$$
\begin{aligned}
P_1 &= (*, *) \\
P_2 &= (*, (*, *)) \\
P_3 &= ((*, *), (*, *)) \\
P_4 &= ((*, *, *), (*, ())) \\
P_5 &= () \\
P_6 &= *
\end{aligned}
$$

我们来定义轮廓的一些重要属性。

定义 2.2.1.3. 假设 P 是一个轮廓。

- X 的秩为

$$
\mathsf{rank}(P) = \begin{cases} 1 & P = * \\ r & P = (P_1, \ldots, P_r) \text{ is a tuple of profiles.} \end{cases}.
$$

- P 的长度 为

$$
\mathsf{len}(P) = \begin{cases} 1 & P = * \\ \sum_{i=1}^{r} \mathsf{len}(P_i) & P = (P_1, \ldots, P_r) \text{ is a tuple of profiles.} \end{cases}
$$

- 深度为 P

$$
\mathsf{depth}(P) = \begin{cases} 0 & P = * \\ 1 + \mathsf{max}_{1 \leq i \leq r}(\mathsf{depth}(P_i)) & P = (P_1, \ldots, P_r) \text{ is a tuple of profiles.} \end{cases}
$$

例 2.2.1.4. 以下是一些轮廓示例，以及它们的秩、长度和深度：

$$
\begin{array}{lllll}
P = * & \mathsf{rank}(P) = 1, & \mathsf{len}(P) = 1, & \mathsf{depth}(P) = 0 \\
P = (*, *, *) & \mathsf{rank}(P) = 3, & \mathsf{len}(P) = 3, & \mathsf{depth}(P) = 1 \\
P = (((*, *), *, *), *, *), & \mathsf{rank}(P) = 3, & \mathsf{len}(P) = 6, & \mathsf{depth}(P) = 3 \\
P = ((((), ()), (*, (*, *))), & \mathsf{rank}(P) = 2, & \mathsf{len}(P) = 3, & \mathsf{depth}(P) = 3
\end{array}
$$

定义 2.2.1.5. 设 P 是一个秩(P)为 $=r$的轮廓。如果 $1 \le i \le r$，那么 P 的第 i个模式为

$$\mathsf{mode}_i(P) = \begin{cases} P & \mathsf{depth}(P) = 0 \text{ (hence } i = r = 1), \\ P_i & P = (P_1, \dots, P_r) \text{ has depth } \ge 1. \end{cases}$$

例 2.2.1.6. 若 $P = ((*,*),((*,(*,*))))$，则 P的模式为

$$\begin{aligned} \mathsf{mode}_1(P) &= (*,*) \\ \mathsf{mode}_2(P) &= (()) \\ \mathsf{mode}_3(P) &= (*,(*,*)). \end{aligned}$$

以下记号将很有用。

Nota式 2.2.1.7. 设 P 是一个深度为 >0的轮廓。对于任意 $1 \le j \le$ 秩(P)，我们记作

$$\begin{aligned} \mathsf{len}_j(X) &= \mathsf{len}(\mathsf{mode}_j(P)), \\ \mathsf{len}_{<j}(P) &= \sum_{i=1}^{j-1} \mathsf{len}_i(X), \\ \mathsf{len}_{\le j}(X) &= \mathsf{len}_{<j}(P) + \mathsf{len}_j(P) \end{aligned}$$

概形支持的最重要操作是替换：若 Q 是一个长度为 m的轮廓，且 $P_1, \dots, P_m$ 是轮廓，那么对于每个 $1 \le i \le m$，用轮廓 P_i 替换 Q 的第 i 个条目，即可得到一个新的轮廓 $(P_1, \dots, P_m)_Q$。更准确地说，定义如下。

定义 2.2.1.8. 设 Q 是一个长度为 m的轮廓，并设 $P_1, \dots, P_m$ 是轮廓。那么，对 $P_1, \dots, P_m$ 进行 Q-替换 所得的轮廓为

$$(P_1, \dots, P_m)_Q$$

定义如下。记深度$(Q) = d$ 和秩$(Q) = r$。

- 若 $d = 0$，则 $m = 1$，并定义
$$(P_1)_Q = P_1.$$
- 接下来，假设 $d > 0$，并且我们已为所有深度为 $< d$ 的轮廓 Q' 定义了 Q'-替换。我们可以写成
$$Q = (Q_1, \dots, Q_r)$$
其中，每个模式 $Q_i = \mathsf{mode}_i(Q)$ 的深度均为 $< d$。若对每个 $1 \le i \le r$，我们令
$$\ell_i = \mathsf{len}(P_1) + \cdots + \mathsf{len}(P_{i-1}),$$
则定义
$$(P_1, \dots, P_r)_Q = ((P_1, \dots, P_{\ell_2})_{Q_1}, \dots, (P_{\ell_r+1}, \dots, P_{\ell_{r+1}})_{Q_r}).$$

例 2.2.1.9. 若 $Q = (*,\ *)$ 且 $P_1 = (*,\ *)$，$P_2 = (*,\ *,\ *)$，

$$(P_1, P_2)_Q = ((*,*),(*,*,*)).$$

更一般地，如果 $Q = (*,\dots,*)$ 是深度为 $Q = 1$、长度为 $Q =$、秩为 $Q = r$ 的轮廓，那么

$$(P_1, \dots, P_r)_Q = (P_1, \dots, P_r)$$

是普通连接。

旁注 2.2.1.10. Q-替换有一种拟运算解释。轮廓的集合 Profile 具有一个（非对称）运算子的结构：该集合

$$\mathsf{Profile}(n) = \{P \in \mathsf{Profile} \mid \mathsf{len}(P) = n\}$$

构成了.....的集合 Profile 中的 n 元运算，并且如果 $n = m_1 + \cdots + m_r$，那么结构映射

$$\mathsf{Profile}(m_1) \times \cdots \times \mathsf{Profile}(m_r) \times \mathsf{Profile}(n) \longrightarrow \mathsf{Profile}(m_1 + \cdots + m_r)$$

$$(P_1, \ldots, P_r), Q \longmapsto (P_1, \ldots, P_r)_Q$$

由 Q 替换给出。我们还可以在这个非对称幺半群上构造余自由对称幺半群，这相当于为 n 元运算的集合赋予平凡的对称群作用。

2.2.2 基本定义

定义了轮廓及其基本性质后，我们现在可以定义嵌套元组。

定义 2.2.2.1。 如果 V 是一个集合，那么一个嵌套元组 X（其条目属于 V）是一个由以下部分构成的有序对 ($X^\flat$, P):

1. 一个元组 $X^\flat = (x_1, \ldots\ldots, x_m)$，其条目属于 V，称为 X 的扁平化，并且

2. 一个轮廓 $\mathsf{prof}(X) = P$，长度为 m，称为 X 的轮廓。

我们用 $\mathsf{Nest}(V)$ 表示所有条目属于集合 V 的嵌套元组所构成的集合。

例 2.2. **2.2.** 下面给出一些嵌套元组的示例，以及它们的扁平化结果和轮廓 .

$X = (2, (2, 2))$	$X^\flat = (2, 2, 2)$	$\mathsf{prof}(X) = (*, (*, *))$
$X = 25$	$X^\flat = (25)$	$\mathsf{prof}(X) = *$
$X = (((2, 2, 2), 8), 64)$	$X^\flat = (2, 2, 2, 8, 26)$	$\mathsf{prof}(X) = (((*, *, *), *), *)$
$X = ((), (32, ()), (4, 8))$	$X^\flat = (32, 4, 8)$	$\mathsf{prof}(X) = ((), (*, ()), (*, *))$

记号 2.2.2.3. 我们有时写作

$$X = (x_1, \ldots, x_m)_P$$

表示一个具有 $X^\flat = (x_1、\ldots\ldots、x_m)$ 和轮廓 $\mathsf{prof}(X) = P$ 的嵌套元组。

观察 2.2.2.4。 若 V 是任意集合，则根据定义，我们有一个拉回方块

$$\begin{array}{ccc} \mathsf{Nest}(V) & \xrightarrow{\mathsf{prof}(-)} & \mathsf{Profile} \\ {\scriptstyle (-)^\flat}\downarrow & \lrcorner & \downarrow{\scriptstyle \mathsf{len}(-)} \\ \mathsf{Tuple}(V) & \xrightarrow[\mathsf{len}(-)]{} & \mathbb{N}. \end{array}$$

注记 2.2.2.5。 根据概形的递归定义，我们也可以等价地将条目属于 V 的嵌套元组定义为以下两者之一

1. V 的一个元素，或

2. 一个由嵌套元组构成的元组，其条目位于 V 中。

我们来定义嵌套元组的一些重要属性。嵌套元组 X 的每个此类属性都会由其扁平化结果 $X^\flat$ 或其轮廓 $\mathrm{prof}(X)$继承。

定义 2.2.2.6. 假设 X 是一个条目取自 V的嵌套元组。

- X 的秩为
$$\mathsf{rank}(X) = \mathsf{rank}(P)$$
- X 的长度为
$$\mathsf{len}(X) = \mathsf{len}(P) = \mathsf{len}(X^\flat)$$
- X 的深度为
$$\mathsf{depth}(X) = \mathsf{depth}(P)$$
- 若 $V = \mathbb{Z}$，则 X 的大小为
$$\mathsf{size}(X) = \mathsf{size}(X^\flat).$$

例2.2.2.7。 下面给出一些整数嵌套元组的示例，以及它们的秩、长度、深度和大小：

$$\begin{array}{lllll}
X = 27 & \mathsf{rank}(X) = 1, & \mathsf{len}(X) = 1, & \mathsf{depth}(X) = 0, & \mathsf{size}(X) = 27 \\
X = (2, 10, 5) & \mathsf{rank}(X) = 3, & \mathsf{len}(X) = 3, & \mathsf{depth}(X) = 1, & \mathsf{size}(X) = 100 \\
X = (((3, 4), 2, 2), 8, 9), & \mathsf{rank}(X) = 3, & \mathsf{len}(X) = 6, & \mathsf{depth}(X) = 3, & \mathsf{size}(X) = 3096 \\
X = ((((), ()), (2, (5, 5))), & \mathsf{rank}(X) = 2, & \mathsf{len}(X) = 3, & \mathsf{depth}(X) = 3, & \mathsf{size}(X) = 50
\end{array}$$

例2.2.2.8。 深度为 0 的整数嵌套元组就是一个整数。

例2.2.2.9。 深度为 1 的整数嵌套元组就是一个整数元组。如果 X 是这样的嵌套元组，那么秩(X) = 长度(X)。

定义 2.2.2.10。 设 $X = (x_1, \ldots, x_m)_P$ 是一个秩为(X) $= r$的嵌套元组。如果 $1 \leq i \leq r$，那么 i第 个模式 X 为嵌套元组

$$\mathsf{mode}_i(X) = (x_{\mathsf{len}_{<i}(P)+1}, \ldots, x_{\mathsf{len}_{\leq i}(P)})_{\mathsf{mode}_i(P)}.$$

例2.2.2.11。 如果

$$X = ((3), 4, ((10, 10), 12)),$$

那么 X 的模式为

$$\begin{aligned}
\mathsf{mode}_1(X) &= (3) \\
\mathsf{mode}_2(X) &= 4 \\
\mathsf{mode}_3(X) &= ((10, 10), 12)
\end{aligned}$$

示例 2.2.2.12. 如果 $X =$(32, 5, 6,64)，那么 X的模式为

$$\begin{aligned}
\mathsf{mode}_1(X) &= 32 \\
\mathsf{mode}_2(X) &= 5 \\
\mathsf{mode}_3(X) &= 6 \\
\mathsf{mode}_4(X) &= 64
\end{aligned}$$

引入以下记号会很方便。

记号 2.2.2.13. 假设 X 是一个深度为 >0 的整数嵌套元组。对于任意 $1 \leq j \leq$秩(X)，我们记作

$$\begin{aligned}
\mathsf{len}_j(X) &= \mathsf{len}(\mathsf{mode}_j(X)),\\
\mathsf{len}_{<j}(X) &= \sum_{i=1}^{j-1} \mathsf{len}_i(X),\\
\mathsf{len}_{\leq j}(X) &= \mathsf{len}_{<j}(X) + \mathsf{len}_j(X)
\end{aligned}$$

类似地，我们记作

$$\begin{aligned}
\mathsf{size}_j(X) &= \mathsf{size}(\mathsf{mode}_j(X)),\\
\mathsf{size}_{<j}(X) &= \prod_{i=1}^{j-1} \mathsf{size}_j(X), \text{ and}\\
\mathsf{size}_{\leq j}(X) &= \mathsf{size}_{<j}(X) \cdot \mathsf{size}_j(X).
\end{aligned}$$

定义 2.2.2.14. 如果 $X = (x_1, \ldots, x_m)_P$是一个嵌套元组，且 $1 \leq i \leq m$，那么 X的第 i 个条目为

$$\mathsf{entry}_i(X) = \mathsf{entry}_i(X^\flat) = x_i.$$

例2.2.2.15. 如果

$$X = ((3), 4, ((10, 10), 12)),$$

那么 X 的条目为

$$\begin{aligned}
\mathsf{entry}_1(X) &= 3\\
\mathsf{entry}_2(X) &= 4\\
\mathsf{entry}_3(X) &= 10\\
\mathsf{entry}_4(X) &= 10\\
\mathsf{entry}_5(X) &= 12.
\end{aligned}$$

例2.2.2.16. 如果 $X = (32, 5, 6, 64)$，那么 X的条目为

$$\begin{aligned}
\mathsf{entry}_1(X) &= 32\\
\mathsf{entry}_2(X) &= 5\\
\mathsf{entry}_3(X) &= 6\\
\mathsf{entry}_4(X) &= 4.
\end{aligned}$$

例2.2.2.17. 如果 X 是深度为 1 的嵌套元组，那么对于所有 $1 \leq i \leq$，若 rank(X) =len(X)，则 mode$_i(X)$ = entry$_i(X)$。

观察 2.2.2.18. 如果 X 是由整数构成的嵌套元组，那么 X 的条目 是整数，而 X 的模式 本身是由整数构成的嵌套元组。

最后，我们引入嵌套元组全等的概念，用于表示嵌套元组何时具有相同的轮廓。

定义 2.2.2.19。 如果 X_1 和 X_2 是嵌套元组，我们称 X_1 和 X_2 全等，当且仅当

$$\mathsf{prof}(X_1) = \mathsf{prof}(X_2).$$

例2.2.2.20. 以下是一些嵌套元组 X_1 和 X_2 的示例，以及它们是否全等

$X_1 = 27$	$X_2 = 100$	全等
$X_1 = (2,2)$	$X_2 = (8, 64)$	全等
$X_1 = ((4,8), (4, 8))$	$X_2 = ((1, 1), (5, 10))$	全等
$X_1 = ((64,(8, 8)), (25,(5, 5)))$	$X_2 = ((2,(3, 5)), (7,(11, 13)))$	全等
$X_1 = 27$	$X_2 = (100)$	不全等
$X_1 = (2,2)$	$X_2 = (8, 64, 128)$	不全等
$X_1 = ((4,\ 8),\ (4,\ 8))$	$X_2 = (((1,\ 1),\ (5,\ 10)))$	不全等

2.2.3 替换

回顾 若 Q 是长度为 r 的轮廓，且 $P_1, \ldots, P_r$ 是轮廓，则我们定义了一个轮廓

$$(P_1, \ldots, P_r)_Q$$

称为 Q-替换，作用于 $P_1, \ldots, P_r$。该轮廓由 Q 得到，方法是将 Q的第 i个条目替换为轮廓 P_i。我们可以将此扩展为嵌套元组上的运算，如下所示。

定义 2.2.3.1. 设 $X_1, \ldots, X_m$ 是轮廓分别为 $P_1, \ldots, P_m$的嵌套元组，并设 Q是一个长度为 m的轮廓。我们将 Q-替换定义为

$$(X_1, \ldots, X_m)_Q$$

作用于 $X_1, \ldots, X_m$，得到一个扁平化结果为

$$(X_1, \ldots, X_m)^\flat_Q = X_1^\flat \star \cdots \star X_m^\flat$$

且轮廓为

$$(P_1, \ldots, P_m)_Q.$$

更一般地，如果 $X_1, \ldots, X_m$ 是嵌套元组，且 Y是一个长度为 m的嵌套元组，我们定义

$$(X_1, \ldots, X_m)_Y = (X_1, \ldots, X_m)_{\mathsf{prof}(Y)}.$$

例 2.2.3.2. 若 $(X_1, X_2, X_3) = (64,16, 4)$，且 $Q = (*, (*, *))$，则

$$(X_1, X_2, X_3)_Q = (64, (32, 4))$$

例 2.2.3.3. 如果 $(X_1,\ X_2,\ X_3,\ X_4) = ((2,\ 2),\ (3,\ 3),\ (5,\ 5),\ (7,\ 7))$ 且 $Q = ((*,\ *),\ (*,\ *))$，那么

$$(X_1, X_2, X_3, X_4)_Q = (((2, 2), (3, 3)), ((5, 5), (7, 7))).$$

例 2.2.3.4. 如果 $X = (12)$ 和 $Q = *$，那么

$$(X)_Q = 12.$$

例 2.2.3.5. 如果 $X_1 = 2$、 $X_2 = 2$、 $X_3 = (5,\ 5)$ 和 $Q = (*,\ *,\ *)$，那么

$$(X_1, X_2, X_3)_Q = (2, 2, (5, 5)) = (X_1, X_2, X_3).$$

更一般地，若 $X_1, \ldots, X_m$ 是任意嵌套元组，且 $P = (*, \ldots, *)$，那么

$$(X_1, \ldots, X_m)_Q = (X_1, \ldots, X_k)$$

就是 X_1、…、 X_m 的串接。

顺带一提 2.2.3.6. 嵌套元组的代换有一种运算子 解释。整数嵌套元组的集合 $\mathsf{Nest}(\mathbb{Z})$ 是运算子 $\mathsf{Profile}$ 上的一个代数，其结构映射由 Q-代换给出：

$$\mathsf{Nest}(\mathbb{Z}) \times \cdots \times \mathsf{Nest}(\mathbb{Z}) \times \mathsf{Profile}(n) \longrightarrow \mathsf{Nest}(\mathbb{Z})$$

$$(X_1, \ldots, X_m), Q \longmapsto (X_1, \ldots, X_m)_Q.$$

2.2.4 细化

本节将介绍嵌套元组上的一种重要关系，称为细化。直观地说，若 X' 和 X 是整数嵌套元组，那么如果可以通过将 X 中的每个条目替换为一个大小相同的嵌套元组，从而得到 X'，我们就说 X' 细化 X。更准确地说，定义如下。

定义 2.2.4.1。 若 X' 和 X 是嵌套元组，那么我们称 X' 细化 X，如果满足以下任一条件 r

1. X =大小(X')，或 2. (a) 深度(X'),深度(X) > 0，(b) 秩(X') =秩(X)，以及 (c) 对每个 $1 \le i \le$ 秩(X)，模式$_i$(X') 细化模式 $_i$(X)。

记号 2.2.4.2. 我们写作

$$X' \twoheadrightarrow X$$

表示 X' 细化 X。

例 2.2.4.3. 以下是嵌套元组细化的一些示例。

$$(2, (2, 2)) \twoheadrightarrow 8$$
$$((2, 2), (3, 3), (5, 5)) \twoheadrightarrow (4, 9, 25)$$
$$(64) \twoheadrightarrow 64$$
$$(8, ((2, 2, 2), ((1, 4), (2, 2)))) \twoheadrightarrow (8, (8, 8))$$

观察 2.2.4.4. 嵌套元组的细化具有自反性、传递性和反对称性，因此细化在正整数嵌套元组的集合上规定了一种偏序。

如果 X' 细化 X，那么我们可以认为 X' 是通过将 X中的每个条目 x_i 替换为某个大小为 x_i的嵌套元组 X'_i 而得到的。我们将嵌套元组 X'_i 称为相对于 X的 X' 的第 i个模式。更精确地说，我们有以下定义。

构造 2.2.4.5. 假设 X 是一个长度为 m的整数嵌套元组，并且假设 X' 细化 X。对于任意 $1 \le i \le m$，我们定义一个嵌套元组

$$X'_i = \mathsf{mode}_i(X', X),$$

称为相对于 X的第 i模式 X'，公式为

$$\mathsf{mode}_i(X', X) = \begin{cases} X' & \mathsf{depth}(X) = 0 \text{ (hence } i = \ell = 1) \\ \mathsf{mode}_{i-N}(\mathsf{mode}_j(X'), \mathsf{mode}_j(X)) & j \text{ is the largest integer such that} \\ & N \coloneqq \mathsf{len}_{<j}(X) < i. \end{cases}$$

例 2.2.4.6. 若 $X = ((4,\ 9),\ (25,\ 36))$，且 $X' = (((2,\ 2),\ (3,\ 3)),\ (25,\ (6,\ (2,\ 3))))$，则 X'细化X，并且 X' 相对于 X 的模式为

$$\begin{aligned} \mathsf{mode}_1(X', X) &= (2, 2) \\ \mathsf{mode}_2(X', X) &= (3, 3) \\ \mathsf{mode}_3(X', X) &= 25 \\ \mathsf{mode}_4(X', X) &= (6, (2, 3)). \end{aligned}$$

例 2.2.4.7. 若 X 是任意嵌套元组，则 X 细化 X，并且对于任意 $1 \le i \le \mathsf{len}(X)$，都有

$$\mathsf{mode}_i(X, X) = \mathsf{entry}_i(X).$$

例 2.2.4.8. 若 $X = X^\flat$ 是一个元组，且 X' 细化 X，则对于任意 $1 \le i \le \mathsf{len}(X)$，都有

$$\mathsf{mode}_i(X', X) = \mathsf{mode}_i(X').$$

例 2.2.4.9. 若 X' 是一个大小(X')为 $= N$ 的嵌套元组，则 X' 细化 N，并且 X' 相对于 N 的唯一模式为

$$\mathsf{mode}_1(X', N) = X'.$$

记号 2.2.4.10. 如果 $X' \twoheadrightarrow X$ 是细化，且 $1 \le i \le \mathsf{len}(X)$，那么我们记作

$$\begin{aligned} \mathsf{len}_i(X', X) &= \mathsf{len}(\mathsf{mode}_i(X', X)) \\ \mathsf{len}_{<i}(X', X) &= \sum_{j<i} \mathsf{len}_j(X', X) \\ \mathsf{len}_{\le i}(X', X) &= \sum_{j\le i} \mathsf{len}_j(X', X) \end{aligned}$$

定义 2.2.4.11. 假设 X' 细化 X，并记 $X'_i =$ 模式$_i(X', X)$。那么，X' 相对于 X 的扁平化是嵌套元组

$$\mathsf{flat}(X', X) = (X'_1, \ldots, X'_m).$$

例 2.2.4.12. 如果 $X' = (((2,2),(3,3)),((5,5),(7,7)))$ 且 $X = ((4,9),(25,49))$，那么

$$\mathsf{flat}(X', X) = ((2,2),(3,3),(5,5),(7,7)).$$

例 2.2.4.13。 如果 X 是任意嵌套元组，则相对于 X， X 的扁平化为

$$\mathsf{flat}(X, X) = X^\flat.$$

例 2.2.4.14。 如果 $X = X^\flat$ 是一个 Tuple，且 X' 细化了 X，则相对于 X ， X' 的扁平化为

$$\mathsf{flat}(X', X) = X'.$$

例 2.2.4.15. 若 X' 是大小为 size(X') $= N$的嵌套元组，则 X' 细化 N，而 X' 相对于 N 的扁平化为

$$\mathsf{flat}(X', N) = (N).$$

观察 2.2.4.16. 如果 X' 细化 X，那么 flat(X', X) 细化 $X^\flat$。

2.3 布局

在介绍完嵌套元组的必要背景知识后，我们将注意力转向布局。它们是平坦布局的推广，其中形状和步幅可以是嵌套元组，而不只是（平坦）元组。

2.3.1 基本定义

定义 2.3.1.1. 一个布局是一对

$$L = S : D$$

由正整数嵌套元组组成

$$\mathsf{shape}(L) = S$$

称为 L 的形状，以及一个非负整数嵌套元组

$$\mathsf{stride}(L) = D$$

称为 L 的步幅，使得 S 和 D 全等。

定义 2.3.1.2。 如果 $L = S : D$ 是一个布局，那么 L 的秩、长度、深度、大小和轮廓分别定义为 S 的秩、长度、深度、大小和轮廓。

例 2.3.1.3。 布局 $L = (3, (3, 2)) : (3, (1, 10))$ 可以表示如下。

$$L = \begin{array}{|c|c|c|c|c|c|} \hline 0 & 1 & 2 & 10 & 11 & 12 \\ \hline 3 & 4 & 5 & 13 & 14 & 15 \\ \hline 6 & 7 & 8 & 16 & 17 & 18 \\ \hline \end{array}$$

Exa例 2.3.1.4。 布局 $L = ((2, 2), (2, 2)) : ((1, 4), (2, 8))$ 可以表示如下 .

$$L = \begin{array}{|c|c|c|c|} \hline 0 & 2 & 8 & 10 \\ \hline 1 & 3 & 9 & 11 \\ \hline 4 & 6 & 12 & 14 \\ \hline 5 & 7 & 13 & 15 \\ \hline \end{array}$$

例 2.3.1.5。布局

$$L = 10 : 4$$

秩(L) 为 $= 1$，长度(L)为$= 1$，深度(L) 为 $= 0$，大小(L) 为 $= 10$，且 prof(L)为$= *$。

例 2.3.1.6. 布局

$$L = (7, (2, 10, 4), (3, 7)) : (1, (7, 14, 140), (560, 1680))$$

其秩为 $\mathsf{rank}(L) = 3$，$\mathsf{len}(L) = 6$，深度 $\mathsf{depth}(L) = 2$，大小 $\mathsf{size}(L) = 11760$，且 $\mathsf{prof}(L) = (*, (*, *, *), (*, *))$。

例 2.3.1.7. 该布局

$$L = ((2, 2, 2, (2, 2))) : ((1, 0, 8, (0, 16)))$$

其秩为 $\mathsf{rank}(L) = 1$，$\mathsf{len}(L) = 5$，深度 $\mathsf{depth}(L) = 3$，大小 $\mathsf{size}(L) = 32$，且 $\mathsf{prof}(L) = ((*, *, *, (*, *)))$。

例 2.3.1.8. 这一对

$$S : D = (2, (2, 2)) : (1, 2, 4)$$

不是布局，因为 S 和 D 不全等。

定义 2.3.1.9. 若 $L = S : D$ 是一个布局，那么对于任意满足 $\mathsf{rank}(L)$ 的 $1 \leq i \leq$，我们将L的第 i 个模式定义为布局

$$\mathsf{mode}_i(L) = \mathsf{mode}_i(S) : \mathsf{mode}_i(D),$$

对于任意 $1 \leq i \leq \mathsf{len}(L)$，我们将 L 的第 i个条目定义为布局

$$\mathsf{entry}_i(L) = \mathsf{entry}_i(S) : \mathsf{entry}_i(D).$$

例 2.3.1.10. 若 $L = \big((2, 2)\ 9\big) : \big((3, 6)\ 12\big)$，则 L 的模式为

$$\mathsf{mode}_1(L) = (2, 2) : (3, 6)$$
$$\mathsf{mode}_2(L) = 9 : 12$$

L 的条目为

$$\mathsf{entry}_1(L) = 2 : 3$$
$$\mathsf{entry}_2(L) = 2 : 6$$
$$\mathsf{entry}_3(L) = 9 : 12.$$

注记 2.3.1.11. 若 L 是一个布局，则 L 的模式也都是布局，而 L 的条目都是深度为 0 的布局。

注记 2.3.1.12. 平坦布局 L 恰好是深度为 1 的布局。另一方面，若 L 是一个布局，我们可以按如下方式得到平坦布局 $L^\flat$ 。

定义 2.3.1.13. 若 $L = S : D$ 是一个布局，我们将 L 的扁平化 定义为平坦布局

$$L^\flat = S^\flat : D^\flat.$$

例 2.3.1.14. $L = 10$ 的扁平化：4 是 $L^\flat = (10) : (4)$。

例 2.3.1.15. 扁平化

$$L = \big((2, 2, 2, (2, 2))\big) : \big((1, 0, 8, (0, 16))\big)$$

is

$$L^\flat = (2, 2, 2, 2, 2) : (1, 0, 8, 0, 16).$$

注记 2.3.1.16. 如果 L 是一个布局，那么 $\mathsf{len}(L)$ =秩$(L^\flat)$，并且对于任意 $1 \leq i \leq \mathsf{len}(L)$，我们有

$$\mathsf{entry}_i(L) = \mathsf{mode}_i(L^\flat).$$

我们可以使用上面的扁平化构造，将许多概念从平坦布局扩展到嵌套布局。例如：

构造 2.3.1.17 （布局函数）。 若 L 是一个嵌套布局，我们将 L 的布局函数 Φ_L 定义为

$$\Phi_L = \Phi_{L^\flat},$$

其中 $\Phi_{L^\flat}$ 是构造 2.1.2.19 的布局函数。类似地，若 N 满足 像$(\Phi_L) \subset [0, N)$，我们定义

$$\Phi_L^N = \Phi_{L^\flat}^N$$

为 Φ_L 经过包含映射 $[0, N) \subset \mathbb{Z}$ 的分解。

例 2.3.1.18. 若 $L = ((2,\ 2),\ 2) : ((3,\ 0),\ 10)$，则布局函数

$$\Phi_L : [0, 8) \to \mathbb{Z}$$

L 的函数为

$$\Phi_L \quad \begin{array}{cccccccc} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 \\ \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow \\ 0 & 3 & 0 & 3 & 10 & 13 & 10 & 13 \end{array}$$

给定一个布局 L，我们可以得到一个平坦布局 $L^\flat$ 和一个轮廓 P =prof(L)。反之，如果给定一个平坦布局 L 和一个轮廓 P，且其长度与 L 相同，那么我们可以按如下方式构造一个具有扁平化 L 和轮廓 P 的布局。

构造 2.3.1.19。 如果 L 是一个平坦布局，且 P 是一个轮廓，并且 $\mathsf{len}(P) = \mathsf{len}(L)$，那么我们可以定义

$$L = L_P$$

将其定义为形状为

$$\mathsf{shape}(L) = \mathsf{shape}(L)_P$$

且步幅为

$$\mathsf{stride}(L) = \mathsf{stride}(L)_P$$

其中 $(-)_P$ 是定义 P中的 2.2.1.8-替换运算。

例 2.3.1.20。 如果 $L = (8,\ 8,\ 8) : (1,\ 64,\ 8)$ 且 $P = (*,\ (*,\ *))$，那么

$$L_P = (8, (8, 8)), (1, (64, 8)).$$

例 2.3.1.21. 如果 $L = (128) : (2)$ 且 $P = *$，那么

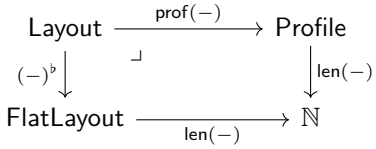
$$L_P = 128 : 2.$$

命题 2.3.1.22. 如果 L' 是平坦布局，且 P 是轮廓，其中 $\mathsf{len}(L') = \mathsf{len}(P)$，那么存在唯一布局 L，其扁平化为 $L^\flat = L'$，且其轮廓为 $\mathsf{prof}(L) = P$，即 $L = L'_P$。

证明。 这由嵌套元组的定义可知，因为一个嵌套元组由其扁平化结果和轮廓唯一确定。

□

观察 2.3.1.23。 前面的命题告诉我们，我们有一个拉回方块



我们可以如下将非退化性的概念扩展到嵌套情形。

定义 2.3.1.24。 设 L 是一个布局。如果对于所有 $1 \leq i \leq \mathsf{len}(L)$，以下条件都成立，则称 L 是非退化的：

$$\mathsf{entry}_i(\mathsf{shape}(L)) \quad \Rightarrow \quad \mathsf{entry}_i(\mathsf{stride}(L))$$

例 2.3.1.25。布局

$$L_1 = ((2,2),1) : ((1,2),0)$$
$$L_2 = ((8,8),(1,16)) : ((2,32),(0,128))$$

是非退化的，而布局

$$L_3 = ((2,2),1) : ((1,2),4)$$
$$L_4 = ((8,8),(1,16)) : ((2,32),(1024,128))$$

是退化的。

2.3.2 基本运算

在确立了布局的基本术语后，我们转而讨论布局所支持的运算。在本节中，我们定义一些基本运算，它们是构造更复杂运算所必需的，例如合并、补、复合、逻辑除法和逻辑乘积。

2.3.2.1 扁平化

若 L 是 一个布局，那么我们可以通过将 L的形状和步幅扁平化，得到一个平坦布局 $L^\flat$.

定义 2.3.2.1 节。 若 $L = S : D$ 是一个布局，我们将 L 的扁平化 定义为平坦布 t

$$L^\flat = S^\flat : D^\flat.$$

例 2.3.2.2. 的扁平化

$$L = ((2,2,2,(2,2))) : ((1,0,8,(0,16)))$$

is

$$L^\flat = (2,2,2,2,2) : (1,0,8,0,16).$$

例 2.3.2.3. $L = 10$ 的扁平化：4 是 $L^\flat = (10) : (4)$。

例 2.3.2.4. 假设 L 是一个布局。那么，$\mathsf{depth}(L) = 1$ 当且仅当 $L = L^\flat$。

2.3.2.2 拼接

我们可以通过拼接布局的形状并拼接其步长来拼接布局。

定义 2.3.2.5. 如果 $L = S : D$ 和 $L' = S' : D'$ 是布局，那么 L 和 L' 的串接 就是布局 (L, L')

$$(L, L') = (S, S') : (D, D').$$

更一般地，如果 $L_1, \ldots, L_k$ 是任意有限个布局的集合，且 $L_i = S_i : D_i$，那么 $L_1, \ldots, L_k$ 的串接就是布局

$$(L_1, \ldots, L_k) = (S_1, \ldots, S_k) : (D_1, \ldots, D_k).$$

注记 2.3.2.6. 嵌套元组（因而也包括布局）的串接不满足结合律。例如，取 $L_1 = 3 : 4$、 $L_2 = 2 : 2$，且 $L_3 = 5 : 1$。那么

$$\big(L_1, (L_2, L_3)\big) = \big(3, (2, 5)\big) : \big(4, (2, 1)\big) \neq \big((3, 2), 5\big) : \big((4, 2), 1\big) = \big((L_1, L_2), L_3\big).$$

此外，这些布局都不等于“三重”串接 $(L_1, L_2, L_3) = (3,2,5) : (4,2,1)$。不过，我们可以看到，这些布局的扁平化结果相同，因此它们的布局函数也相同。

例 2.3.2.7. 若 $L = (3,7,2) : (1,3,6)$ 且 $L' = (2, (2, (4, 3))) : (5, 3, (2, 2))$，则

$$(L, L') = ((3, 7, 2), (2, (2, (4, 3)))) : ((1, 3, 6), (5, (3, (2, 2))))$$

注记 2.3.2.8. 串接会增加布局的深度。更确切地说，我们有 e

$$\mathsf{depth}(L, L') = 1 + \max(\mathsf{depth}(L), \mathsf{depth}(L')).$$

注记 2.3.2.9. 当 L 和 L' 是平坦布局时，定义2.3.2.5 中的串接不同于定义 2.1.3.36中的平坦布局串接。相反，这些运算满足以下公式

$$L \star L' = (L, L')^\flat.$$

*Rema*秩 2.3.2.10. 若 L 是任意布局，其深度$(L) > 0$ 且秩$(L) = r$，则我们可以写作

$$L = (\mathsf{mode}_1(L), \ldots, \mathsf{mode}_r(L))$$

作为其各模式的串接。

例 2.3.2.11. 若

$$L = ((5, (7, 7)), 2, (4, 5)) : ((1, (35, 5)), 0, (1, 8))$$

那么 $L = (L_1, L_2, L_3)$，其中

$$\begin{aligned} L_1 &= \big(5, (7, 7)\big) : \big(1, (35, 5)\big), \\ L_2 &= 2 : 0, \text{ and} \\ L_3 &= (4, 5) : (1, 8). \end{aligned}$$

2.3.2.3 替换

回顾一下，如果 $X_1, \ldots, X_k$ 是嵌套元组，且 P 是一个轮廓，len(P) 为 $= k$，那么我们可以构造P-替换

$$(X_1, \ldots, X_k)_P$$

这是通过将 i P 的条目替换为嵌套元组 X_i 得到的。我们可以将这一构造从嵌套元组推广到布局，如下所示。

定义 2.3.2.12. 假设 $L = S : D$ 是一个布局，并假设 P 是一个轮廓，满足 len(P) $=$ rank(L)。我们定义

$$L_P = S_P : D_P$$

其中 S_P 和 D_P 分别是对 S 和 D（的模式）的 P-代换。

例 2.3.2.13. 如果 $P = (*, (*, *))$ 且 $L = (8, 8, 8) : (1, 8, 64)$，那么

$$L_P = (8, (8, 8)) : (1, (8, 64)).$$

例 2.3.2.14. 如果 $P = (*, (*, *)))$ 且

$$L = ((2, 2), (3, 3), (5, 5)) : ((2, 1), (12, 4), (180, 36)),$$

then

$$L_P = ((2, 2), ((3, 3), (5, 5))) : ((2, 1), ((12, 4), (180, 36))).$$

例 2.3.2.15. 如果 $L = (16) : (1)$ 且 $P = *$，那么

$$L_P = 16 : 1.$$

2.3.3 合并

回忆一下，如果 L 是平坦布局，那么 coal(L) 是唯一一个布局函数为 Φ_L 的最小秩平坦布局。我们可以在任意（嵌套）布局的情形下进行类似的构造。我们先定义合并布局的概念。

定义 2.3.3.1. 设 L 是一个布局。如果满足以下条件之一，则称 L 为合并的 。

1. $L = 1 : 0$,
2. depth(L) $= 0$ 且 shape(L) > 1，或
3. 深度(L) $= 1$，秩(L) > 1，且 L 按照定义 2.1.4.1 的意义是合并的。

例 2.3.3.2. 该布局

$$L = (2, (2, 2)) : (1, (16, 512))$$

不是合并的，因为深度(L) > 1。

例 2.3.3.3. 该布局

$$L = (64) : (2)$$

不是合并的，而布局

$$L' = 64 : 2$$

已合并。

例 2.3.3.4. 该布局

$$L = 1 : 8$$

未合并，而布局

$$L' = 1 : 0$$

已合并。

例 2.3.3.5。空布局

$$E = () : ()$$

未合并。

观察 2.3.3.6。 回想一下，如果布局 L是非退化的，则

$$\mathsf{entry}_i(\mathsf{shape}(L)) = 1 \quad \Rightarrow \quad \mathsf{entry}_i(\mathsf{stride}(L)) = 0.$$

如果 L 已合并，则 L是非退化的。

如果 L 是任意布局，我们可以按如下方式得到合并布局 coal(L)。

构造 2.3.3.7。 假设 L 是一个布局，记作

$$\mathsf{coal}^\flat(L^\flat) = (s_1, \ldots, s_m) : (d_1, \ldots, d_m).$$

1. 如果 $m > 1$，我们定义

$$\mathsf{coal}(L) = \mathsf{coal}^\flat(L^\flat)$$

2. 如果 $m = 1$，我们定义

$$\mathsf{coal}(L) = s_1 : d_1$$

3. 如果 $m = 0$，我们定义

$$\mathsf{coal}(L) = 1 : 0.$$

例 2.3.3.8。 如果 $E =$() : () 是空布局，那么

$$\mathsf{coal}(E) = 1 : 0.$$

例 2.3.3.9. 如果 $L =$(1,1) : (2, 4)，那么

$$\mathsf{coal}(L) = 1 : 0.$$

例 2.3.3.10. 如果 $L =$ (512) : (4)，那么

$$\mathsf{coal}(L) = 512 : 4.$$

例 2.3.3.11. 如果 $L =$(2,2, 2) : (1, 2, 4)，那么

$$\mathsf{coal}(L) = 8 : 1.$$

例 2.3.3.12. 如果 $L =$((2, 2, 2), (5, 5)) : ((1, 2, 4), (10,50))，那么

$$\mathsf{coal}(L) = (8, 25) : (1, 10).$$

注记 2.3.3.13。 如果 L 是一个布局，那么 coal(L) 的深度为 0 或 1。

命题 2.3.3.14。 如果 A和 B 是布局，那么

$$\Phi_A = \Phi_B \quad \Leftrightarrow \quad \mathsf{coal}(A) = \mathsf{coal}(B).$$

证明。 根据命题 2.1.4.18，我们有

$$\begin{aligned} \Phi_A = \Phi_B \quad &\Leftrightarrow \quad \Phi_{A^\flat} = \Phi_{B^\flat} \\ &\Leftrightarrow \quad \mathsf{coal}^\flat(A^\flat) = \mathsf{coal}^\flat(B^\flat) \\ &\Leftrightarrow \quad \mathsf{coal}(A) = \mathsf{coal}(B). \end{aligned}$$

□

定义 2.3.3.15。 若 L 是一个布局，则将 L 的复杂度定义为整数

$$\mathsf{complexity}(L) = \mathsf{len}(L) + \mathsf{depth}(L).$$

命题 2.3.3.16。 若 L 是一个布局，且大小$(L) > 1$，则$\mathsf{coal}(L)$ 是布局函数为 Φ_L 的唯一复杂度最小化布局。

证明。 假设 e L' 是一个与 L 具有相同布局函数的布局，并假设 coal(L') $\neq 1$: 0。于是

$$\mathsf{len}(L') \geq \mathsf{len}(\mathsf{coal}(L')) = \mathsf{len}(\mathsf{coal}(L)).$$

需要考虑两种情况。

- （情形 1）：假设 len($L\prime$) > 1。则 depth($L\prime$) $\geq 1 \geq$ depth(coal(L))。结合这些不等式，我们注意到

  $$\mathsf{complexity}(L') \geq \mathsf{complexity}(\mathsf{coal}(L)),$$

  当且仅当 L' =coal(L') =coal(L) 时，等号成立。

- （情形 2）：假设 len($L\prime$) $= 1$。则 $L\prime = (s) : (d)$，或 $L\prime = s : d$ 对于某些整数 $s > 1$ 和$d \geq 0$。无论哪种情形，我们都有 coal($L\prime$) $= s : d$，并且

  $$\mathsf{complexity}(L') \geq \mathsf{complexity}(\mathsf{coal}(L)),$$

  当且仅当 $L' = s : d =$ coal(L) 时，等号成立。

□

注记 2.3.3.17。 我们需要排除 size(L) $= 1$ 这一情形的唯一原因是：如果 size(L) $= 1$，那么 1 : 0 和空布局 () : () 是具有最小复杂度的不同布局，并且具有与 L 相同的布局函数（即平凡布局函数 $0 \mapsto 0$）。

2.3.4 相对合并

¯合并有一个重要的不变量，称为相对合并，记作 coal(L, S)。此操作¯还会接收一个额外输入，即由 shape(L) 细化的嵌套元组 S。在这种情况下，相对合并运算会尽可能简化布局 L，同时确保所得形状¯仍细化 S。

定义 2.3.4.1。 假设 $L = S : D$ 是一个布局，并且假设 S 是长度为 m、由 S 细化的某个嵌套元组。回想一下，对于任意 $1 \le i \le m$，我们可以考虑相对于 S 的 S 的第 i 个模式，记作

$$\mathsf{mode}_i(S, \bar{S}).$$

由于 S 和 D 全等，因此存在一个嵌套元组

$$\mathsf{mode}_i(D, \bar{S})$$

对应于模式$_i(S, S)$，并将 L 相对于 S 的第 i个模式定义为布局对应

$$\mathsf{mode}_i(L, \bar{S}) = \mathsf{mode}_i(S, \bar{S}) : \mathsf{mode}_i(D, \bar{S}).$$

例 2.3.4.2. 若 $S = (4, (9{,}25))$ 且

$$L = ((2, 2), ((3, 3), (5, (1, 5)))) : ((1, 2), ((6, 18), (90, (0, 450))))$$

then

$$\begin{aligned}
\mathsf{mode}_1(L, \bar{S}) &= (2, 2) : (1, 2) \\
\mathsf{mode}_2(L, \bar{S}) &= (3, 3) : (6, 18) \\
\mathsf{mode}_3(L, \bar{S}) &= (5, (1, 5)) : (90, (0, 450)).
\end{aligned}$$

观察 2.3.4.3. 设 $L = S : D$ 是一个布局，且设 S 是一个长度为 m、轮廓为 P 的嵌套元组，它由 S细化。若对于任意 $1 \le i \le m$，我们记

$$L_i = \mathsf{mode}_i(L, \bar{S}),$$

then

$$L = (L_1, \ldots, L_m)_P$$

是其相对模式的 P-替换。

定义 2.3.4.4. 假设 $L = S : D$ 是一个布局，并假设 S 是一个长度为 m 、轮廓为 P 的嵌套元组，且该轮廓由 S细化。若每个相对模式

$$\mathsf{mode}_i(L, \bar{S})$$

都是合并的。

观察 2.3.4.5. 在定义 2.3.4.4的设定下，若 L 在 S上合并，则 L 是非退化的。

例 2.3.4.6. 若 L 是一个布局，则 L 在形状(L)上合并，当且仅当 L 是非退化的，即

$$\mathsf{entry}_i(\mathsf{shape}(L)) = 1 \quad \Rightarrow \quad \mathsf{entry}_i(\mathsf{stride}(L)) = 0.$$

定义 2.3.4.7 (相对合并)。 假设 $L = S : D$ 是一个布局，并假设 S 是一个长度为 m 、轮廓为 P 的嵌套元组，且该轮廓由 S 细化。我们定义

$$\mathsf{coal}(L, \bar{S}) = (\mathsf{coal}(L_1), \ldots, \mathsf{coal}(L_m))_P.$$

注记 2.3.4.8. 在定义 2.3.4.7中，$\mathsf{coal}(L, S)$ 的形状细化 S。

引理 2.3.4.9。 若 $L = S : D$ 是一个布局，且 S 细化 S，则

$$\Phi_{\mathsf{coal}(L,\bar{S})} = \Phi_L.$$

证明。 如上，令

$$L_i = \mathsf{mode}_i(L, S)$$

表示相对于 S 的 L 的第 i 个模式，并令 $L_i = \mathsf{coal}(L_i)$。于是

$$\begin{aligned}
\Phi_{\mathsf{coal}(L,\bar{S})} &= \Phi_{(\bar{L}_1,\ldots,\bar{L}_m)_{\bar{S}}} \\
&= \Phi_{(\bar{L}_1,\ldots,\bar{L}_m)} \\
&= \Phi_{\mathsf{coal}((\bar{L}_1,\ldots,\bar{L}_m))} \\
&= \Phi_{\mathsf{coal}((L_1,\ldots,L_m))} \\
&= \Phi_{(L_1,\ldots,L_m)} \\
&= \Phi_{(L_1,\ldots,L_m)_{\bar{S}}} \\
&= \Phi_L.
\end{aligned}$$

□

命题 2.3.4.10。 设 A 和 B 是布局，并设 S 是长度为 m 的嵌套元组，使得 形状(A) 细化 S，且 形状(B) 细化 S。于是

$$\Phi_A = \Phi_B \quad \Leftrightarrow \quad \mathsf{coal}(A, \bar{S}) = \mathsf{coal}(B, \bar{S})$$

证明。 若 $\mathsf{coal}(A, S) = \mathsf{coal}(B, S)$，则根据引理 2.3.4.9，我们有

$$\Phi_A = \Phi_{\mathsf{coal}(A,\bar{S})} = \Phi_{\mathsf{coal}(B,\bar{S})} = \Phi_B.$$

反过来，假设 $\Phi_A = \Phi_B$。我们将论证 $\mathsf{coal}(A, S) = \mathsf{coal}(B, S)$。令 $P = \mathsf{prof}(S)$，并对任意 $1 \le i \le m$，令

$$\begin{aligned}
A_i &= \mathsf{mode}_i(A, \bar{S}) \\
B_i &= \mathsf{mode}_i(B, \bar{S}).
\end{aligned}$$

由于

$$\mathsf{coal}(A, \bar{S}) = (\mathsf{coal}(A_1), \ldots, \mathsf{coal}(A_m))_P$$

and

$$\mathsf{coal}(B, \bar{S}) = (\mathsf{coal}(B_1), \ldots, \mathsf{coal}(B_m))_P$$

只需证明对于所有 $1 \le i \le m$，$\mathsf{coal}(A_i) = \mathsf{coal}(B_i)$。根据逆字典序同构的结合律，我们可以将 A 的布局函数 Φ_A 写成

$$[0, \mathsf{size}(A)) \xrightarrow{\mathsf{colex}^{-1}} \prod_{j=1}^m [0, \mathsf{size}(A_j)) \xrightarrow{\prod \Phi_{A_j}} \prod_{j=1}^m \mathbb{Z} \xrightarrow{+} \mathbb{Z}$$

并且我们可以将 B 的布局函数 Φ_B 写作

$$[0, \mathsf{size}(B)) \xrightarrow{\mathsf{colex}^{-1}} \prod_{j=1}^m [0, \mathsf{size}(B_j)) \xrightarrow{\prod \Phi_{B_j}} \prod_{j=1}^m \mathbb{Z} \xrightarrow{+} \mathbb{Z}$$

对于固定的 $1 \le i \le m$，考虑子集

$$[0, \mathsf{size}(A_i)) \subset \prod_{j=1}^{m} [0, \mathsf{size}(A_j))$$

及其像

$$\mathsf{colex}([0, \mathsf{size}(A_i))) \subset [0, \mathsf{size}(A)).$$

由于对于所有 $1 \le j \le m$，$\mathsf{size}(A_j) = \mathsf{size}(B_j)$，这与该像相同

$$\mathsf{colex}([0, \mathsf{size}(B_j))) \subset [0, \mathsf{size}(B)) = [0, \mathsf{size}(B)).$$

将 Φ_A 限制到此子集上，得到 Φ_{A_i}；将 B 限制到此子集上，得到 Φ_{B_i}，因此可得 $\Phi_{A_i} = \Phi_{B_i}$，并且根据命题 2.3.3.14，我们有 $\mathsf{coal}(A_i) = \mathsf{coal}(B_i)$。由此我们推得

$$\mathsf{coal}(A, \bar{S}) = \mathsf{coal}(B, \bar{S}),$$

正如所需。 □

2.3.5 紧凑布局

我们可以轻松地将紧凑布局的概念扩展到嵌套情形。同样，就表示布局的标准网格图而言，若每个整数 $0 \le i < \mathsf{size}(L)$ 都恰好出现一次，则布局 L 是紧致的。更准确地说，我们有如下定义。

定义 2.3.5.1. 假设 L 是一个布局。若布局函数是紧致 n

$$\Phi_L^{\mathsf{cosize}(L)} : [0, \mathsf{size}(L)) \to [0, \mathsf{cosize}(L))$$

是一个同构。

例 2.3.5.2. 该布局

$$A = ((2,2),(2,2)) : ((1,4),(2,8)) =$$

0	2	8	10
1	3	9	11
4	6	12	14
5	7	13	15

是紧致的，而这些布局

$$B = ((2,2),(2,2)) : ((1,4),(2,32)) =$$

0	2	32	34
1	3	33	35
4	6	36	38
5	7	37	39

and

$$C = ((2,2),(2,2)) : ((1,4),(2,0)) =$$

0	2	0	2
1	3	1	3
4	6	4	6
5	7	5	7

并不紧致。

例 2.3.5.3。 以下布局是紧致的:

$$L_1 = (2,(2,2)) : (8,(1,4))$$
$$L_2 = ((8,1),(8,32)) : ((2,0),(16,128))$$
$$L_3 = 64 : 1$$

例 2.3.5.4。布局

$$L = (2,(2,2)) : (4,(8,16))$$

不是紧致的，因为整数 $1 \in [0,29) = [0,$余大小$(L))$ 不在 Φ_L 的像中。更一般地，如果 大小$(L) \neq$余大小(L)，那么 L 不是紧致的。

We c最后，我们列出布局 L 为紧致的一些等价条件 t.

命题 2.3.5.5。 设 L 是一个布局。那么以下条件等价。

1. L 是紧致的。*2.*
$L^\flat$ 是紧致的。*3.*
$\mathsf{coal}(L)$ 是紧致的。

证明。 这些条件等价，这是由以下事实得出的:

$$\Phi_L = \Phi_{L^\flat} = \Phi_{\mathsf{coal}(L)}.$$

□

2.3.6 补集

我们可以将补的概念轻松扩展到嵌套情形，如下所示。

定义 2.3.6.1。 假设 A 和 B 是布局。若串联布局 (A,B) 是紧致的，则称 B 为 A的补，并记作 $A \perp B$。

引理 2.3.6.2。 假设 A和 B 是布局。那么

$$A \perp B \quad \Leftrightarrow \quad A^\flat \perp B^\flat.$$

证明。 这可由以下观察得出：$(A,B)^\flat = A^\flat \star B^\flat$。 □

定义 2.3.6.3。 假设 A 是一个布局。若 $A^\flat$ 是可补的，我们就称 A 是可补的 。

引理 2.3.6.4. 假设 A 是一个布局。当且仅当 A 可补时， A 存在一个补 B 。

证明。 如果 A 可补，那么 $A^\flat$ 可补，因此存在一个平坦布局 B ，使得平坦连接 $A^\flat \star B$ 是紧致的。由此可知，串接 (A, B) 也是紧致的，因此 A 有补。反过来，假设存在一个布局 B ，使得 (A, B) 是紧致的。那么 $B^\flat$ 是 $A^\flat$的补，因此根据命题 2.1.6.21， $A^\flat$ 可补；由定义可知， A 也可补。

□

定义 2.3.6.5. 假设 A 是一个布局。如果 A 可补，那么我们定义

$$\mathsf{comp}(A) = \mathsf{coal}(\mathsf{comp}^\flat(A^\flat)),$$

如构造 2.1.6.16中所述。如果 N 是一个正整数，且 A 是 N-可补的，那么我们定义

$$\mathsf{comp}(A, N) = \mathsf{coal}(\mathsf{comp}^\flat(A^\flat, N))$$

如构造 2.1.6.29。

注记 2.3.6.6。 假设 A 是一个可补布局。那么我们几乎总有 $\mathsf{comp}(A)$ =$\mathsf{comp}^\flat(A^\flat)$。更准确地说，如果 $\mathsf{comp}^\flat(A^\flat)$ 的长度为 > 1，那么

$$\mathsf{comp}(A) = \mathsf{comp}^\flat(A^\flat),$$

如果 $\mathsf{comp}^\flat(A^\flat) = (s) : (d)$ 的长度为 1，那么

$$\mathsf{comp}(A) = s : d,$$

并且如果 $\mathsf{comp}^\flat(A^\flat) = 0 : 0$，那么

$$\mathsf{comp}(A) = 1 : 0.$$

定义 2.3.6.7。 假设 A 是一个布局，且 N 是一个正整数。我们称布局 B 为 A 的N-补 ，如果 $A \perp B$，并且

$$\mathsf{size}(A) \cdot \mathsf{size}(B) = N.$$

定义 2.3.6.8。 假设 A 是一个布局，且 N 是一个正整数。我们称 A 为 N-可补，如果平坦布局 $A^\flat$ 是 N-可补的，如定义 2.1.6.24所述。

命题 2.3.6.9. 假设 A 是一个布局。当且仅当A 是 N-可补的，才存在 A 的一个 N-补。

证明。 如果 B是一个 N-补，属于 A，那么 $B^\flat$是一个 N-补，属于 $A^\flat$，因此根据命题 2.1.6.32， $A^\flat$ 是 N-可补的，所以 A也是。反过来，如果 A是 N-可补的，那么$\mathsf{comp}(A, N)$ 是一个N-补，属于 A。

□

例 2.3.6.10. 如果 $A = ((4, 2), (2, 2)) : ((3, 24), (192, 96))$ 且 $N = 768$ ，那么

$$\mathsf{comp}(A, N) = (3, 2, 2, 2) : (1, 12, 48, 384).$$

例 2.3.6.11. 如果 $A = ((16,4), 64) : ((1, 16), 64)$ 且 $N = 4096$ 那么

$$\begin{aligned}\mathsf{comp}(A, N) &= \mathsf{coal}((\,) : (\,)) \\ &= 1 : 0.\end{aligned}$$

例 2.3.6.12. 如果 $A = (16,4),64) : ((1,16),64)$ 且 $N = 8192$ 那么

$$\begin{aligned}\mathsf{comp}(A,N) &= \mathsf{coal}((2) : (4096)) \\ &= 2 : 4096.\end{aligned}$$

例 2.3.6.13。 如果 $A = (16,\ 4),\ 64) : ((8,\ 1),\ 128)$ 且 $N = 16384$，那么

$$\begin{aligned}\mathsf{comp}(A,N) &= \mathsf{coal}((2,2) : (4,8192)) \\ &= (2,2) : (4,8192).\end{aligned}$$

2.3.7 复合

本节讨论布局上最重要的运算，即 复合。如果 A 和 B 是布局，那么 A 和 B 的复合是一个布局 $B \circ A$，其布局函数是 A 和 B 的布局函数的复合。更准确地说，有如下定义。

定义 2.3.7.1（布局的复合）。 设 A 和 B 是布局。 A 和 B 的复合 是满足以下性质的唯一布局 $B \circ A$。

1. $\mathsf{shape}(B \circ A)$ 细化 $\mathsf{shape}(A)$；2. $B \circ A$ 在 $\mathsf{shape}(A)$ 上是合并的；3. $\Phi_{B\circ A} = \Phi_B \circ \Phi_A^{\mathsf{size}(B)}$。

注记 2.3.7.2。 为了使 $B \circ A$ 存在，我们必须有

$$\mathsf{Image}(\Phi_A) \subseteq [0, \mathsf{size}(B)).$$

注记 2.3.7.3。 布局复合的定义中隐含着一个断言，即满足这三个条件的布局至多只有一个。这可由命题 2.3.4.10证明。我们可以将 弱复合定义为 A 与 B 的一个布局 C，使其满足条件 1 和 3（但不一定满足条件 2）；在这种情况下

$$B \circ A = \mathsf{coal}(C, \mathsf{shape}(A))$$

稍后我们将看到，在尝试计算布局的复合时，先计算 C 与 A 的 B任意弱复合，再对 形状(A) 进行合并，从而得到实际复合 $B \circ A$，是很有用的。注记 2.3.7.4。 注意，由观察 2.3.4.5可知，复合定义中的条件 2 意味着 $B \circ A$ 是非退化的。

例 2.3.7.5。 如果 $A = (3,\ 5) : (10,\ 2)$ 且 $B = (100) : (7)$，那么

$$B \circ A = (3,5) : (70,14).$$

例 2.3.7.6. 若 $A = (4) : (2)$ 且 $B = (2,2,6) : (12,6,1)$，则 A与 B的复合为

$$B \circ A = ((2,2)) : ((6,1)).$$

注记 2.3.7.7. 例 2.3.7.6 说明，平坦布局的复合 A 与 B 不一定是平坦布局。

例 2.3.7.8. 若 $A = ((2,4),8) : ((4,8),8)$，且 $B = (4,4,4,4) : (2,4,8,16)$，则

$$B \circ A = ((2,(2,2)),(2,4)) : ((4,(8,8)),(8,8)).$$

例 2.3.7.9. 若 $A = ((3,(2,2)),24) : ((3,(9,18)),72)$，且 $B = (9,8,3,8) : (24,3,1,384)$，则

$$B \circ A = ((3,(2,2)),(3,8)) : ((72,(3,6)),(1,384))$$

接下来，我们将推导一些用于计算布局复合的实用性质。

˜命题 2.3.7.10. 设 A 是一个布局，并设 B 和 B 是满足以下条件的布局：

- 大小$(B) \leq$ 大小$(\tilde{B})$，并且
- $\Phi_{\tilde{B}} \restriction_{\text{大小}(B)} = \Phi_B$。

若 A 和 B 可复合，则

$$B \circ A = \tilde{B} \circ A.$$

证明。设 A 和 B 可复合。那么余大小$(A) \leq$ 大小(B)，并且 $B \circ A$ 是˜ A 与 B 的复合，这一事实可由等式推出

$$\begin{aligned} \Phi_{\tilde{B}} \circ \Phi_A^{\mathsf{size}(\tilde{B})} &= (\Phi_{\tilde{B}}) \restriction_{\mathsf{size}(B)} \circ \Phi_A^{\mathsf{size}(B)} \\ &= \Phi_B \circ \Phi_A^{\mathsf{size}(B)}. \end{aligned}$$

□

推论 2.3.7.11。如果 A 和 B 是布局，那么当且仅当 A 和coal（B）可复合时，A 和 B 可复合，并且

$$B \circ A = \mathsf{coal}(B) \circ A.$$

现在我们已经介绍了布局复合的基本性质，接下来将注意力转向复合中最重要的两个实例，即逻辑除法和逻辑乘积。

2.3.8 逻辑除法

在本节中，我们定义布局的**逻辑除法**。作为一个引导性示例，考虑布局

$A = (4,\ 8) : (1,\ 4) =$

0	4	8	12	16	20	24	28
1	5	9	13	17	21	25	29
2	6	10	14	18	22	26	30
3	7	11	15	19	23	27	31

出于各种目的，我们可能希望对布局 A 进行平铺。例如，以下是通过各种布局 B对 A 进行平铺的结果。

$A =$

0	4	8	12	16	20	24	28
1	5	9	13	17	21	25	29
2	6	10	14	18	22	26	30
3	7	11	15	19	23	27	31

$B =$

0	4
1	5

$A =$

0	4	8	12	16	20	24	28
1	5	9	13	17	21	25	29
2	6	10	14	18	22	26	30
3	7	11	15	19	23	27	31

$B =$

0	4	8	12
2	6	10	14

$A =$

0	4	8	12	16	20	24	28
1	5	9	13	17	21	25	29
2	6	10	14	18	22	26	30
3	7	11	15	19	23	27	31

$B =$

0	4	16	20
2	6	18	22

使用这类分块布局时，我们希望用形如 (`tile coordinate`, `tile`) 的坐标对布局进行索引，其中 `tile` 指定我们正在处理哪个分块，`tile coordinate` 指定该分块内的坐标。例如，如果 A 和 B 的秩都是 2，我们希望将 $((i, j), (k, \ell))$ 写作 A 中第 (k, ℓ) 个分块的第 (i, j) 个条目的索引。$A \oslash B$ 的逻辑除法恰好提供了这种能力。

定义 2.3.8.1. 假设 A 和 B 是布局，并且假设

$$B^c = \mathsf{comp}(B, \mathsf{size}(A))$$

是补 数， B 相对于 $\mathsf{size}(A)$ 的补数。那么 A 除以 B 的逻辑除法是布局

$$\begin{aligned} A \oslash B &= A \circ (B, B^c) \\ &= (A \circ B, A \circ B^c). \end{aligned}$$

例 2.3.8.2. 如果 $A = (4, 8) : (1, 4)$，且 $B = (2, 2) : (1, 4)$，那么

$$A \oslash B = ((2, 2), (2, 4)) : ((1, 4), (2, 8)),$$

如图所示。

$A =$

0	4	8	12	16	20	24	28
1	5	9	13	17	21	25	29
2	6	10	14	18	22	26	30
3	7	11	15	19	23	27	31

$B =$

0	4
1	5

$A \oslash B =$

0	2	8	10	16	18	24	26
1	3	9	11	17	19	25	27
4	6	12	14	20	22	28	30
5	7	13	15	21	23	29	31

注记 2.3.8.3。 $A \oslash B$ 中每个条目的颜色 表示它所属的分块，而 $A \oslash B$ 中每个条目的不透明度表示它代表分块中的哪个条目。这就是为什么 $A \oslash B$ 的每一列颜色都相同，而 $A \oslash B$ 的每一行不透明度都相同。

例 2.3.8.4。 若 $A = (4,8) : (1,4)$，且 $B = (2,2) : (4,1)$，则

$$A \oslash B = ((2,2),(2,4)) : ((4,1),(2,8)),$$

如下面所示。

$A =$

0	4	8	12	16	20	24	28
1	5	9	13	17	21	25	29
2	6	10	14	18	22	26	30
3	7	11	15	19	23	27	31

$B =$

0	1
4	5

$A \oslash B =$

0	2	8	10	16	18	24	26
4	6	12	14	20	22	28	30
1	3	9	11	17	19	25	27
5	7	13	15	21	23	29	31

注记 2.3.8.5. 请注意前面两个示例之间的区别。在这两个示例中， A 的平铺完全相同，但每个分块的布局不同。在第一个示例中，分块采用列主序布局；而在第二个示例中，分块采用行主序布局。因此，进行逻辑除法时会得到不同的布局。

例 2.3.8.6。 如果 $A = (4,\ 8) : (1,\ 4)$ 且 $B = (2,\ 4) : (2,\ 4)$，那么

$$A \oslash B = ((2,4),(2,2)) : ((2,4),(1,16)).$$

例 2.3.8.7。 如果 $A = (4,\ 6) : (1,\ 40)$ 且 $B = 6 : 4$，那么

$$A \oslash B = (6,4) : (40,1).$$

Exa例 2.3.8.8。 如果 $A = (4,\ 6,\ 2,\ 4,\ 2,\ 5) : (36,\ 1,\ 18,\ 0,\ 0,\ 144)$ 且 $B = (4,\ 10) : (1,\ 192)$，那么

$$A \oslash B = (((4,(2,5)),(6,2,4)) : ((36,(0,144)),(1,18,0))$$

例 2.3.8.9。 如果 $A = (8,\ (4,\ 4))$ 且 $B = (2,\ (8,\ 16))$，那么

$$A \oslash B = ((2,2),(2,(4,4))) : ((4,8),(2,(8,16))).$$

2.3.9 逻辑乘积

本节定义布局的**逻辑乘积**。

定义 2.3.9.1. 假设 A和 B 是布局，并且假设

$$A^c = \mathsf{comp}\big(A, \mathsf{size}(A) \cdot \mathsf{cosize}(B)\big)$$

是相对于大小(A) ·余大小(B)而言的 A 的补。那么， A 和 B 的逻辑乘积 是布局

$$A \otimes B = (A, A^c \circ B).$$

观察 2.3.9.2. 根据命题2.3.7.10 和命题 2.3.7.11，如果令

$$\widetilde{A}^c = \mathsf{comp}(A, N)$$

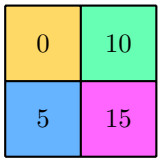
对于任意有效的 $N \geq$ 大小(A) ·余大小(B)，那么

$$A^c \circ B = \tilde{A}^c \circ B.$$

这意味着，在计算 $A \otimes B$时，我们可以将 A^c 取为 A的任意足够大的（已排序）补。

例 2.3.9.3. 如果 $A = (2,2) : (5,10)$ 和 $B = (3,5) : (5,1)$ 是布局

$A =$



$B =$

0	1	2	3	4
5	6	7	8	9
10	11	12	13	14

那么 $A \otimes B$ 就是该布局

$$A \otimes B = ((2,2),(3,5)) : ((5,10),(20,1))$$

如下图所示。

0	20	40	1	21	41	2	22	42	3	23	43	4	24	44
5	25	45	6	26	46	7	27	47	8	28	48	9	29	49
10	30	50	11	31	51	12	32	52	13	33	53	14	34	54
15	35	55	16	36	56	17	37	57	18	38	58	19	39	59

例 2.3.9.4. 若 $A = (3,3) : (6,1)$，且 $B = (10,12) : (24,2)$，则

$$A \otimes B = \big((3,3),(10,12)\big) : \big((6,1),(216,18)\big).$$

例 2.3.9.5. 若 $A = (2,10) : (1680,4)$，且 $B = (4,9) : (2,56)$，则

$$A \otimes B = ((2,10),((2,2),(3,3))) : ((1680,4),((2,40),(560,3360))).$$

例 2.3.9.6. 若 $A = (4,(2,2)) : (9,(1,3))$，且 $B = ((2,4),8) : ((1,4),2)$，则

$$A \otimes B = ((4,(2,2)),((2,4),8)) : ((9,(1,3)),((36,144),72)).$$

2.3.10 可处理布局

本节将定义一类性质尤为良好的布局，称为可处理的布局。我们将看到，可处理布局恰好是源自某个特定范畴 **Nest** 的布局。

定义 2.3.10.1. 我们称布局 L 是可处理的，如果平坦布局 $L^\flat$ 是可处理的，含义如定义 2.1.8.1所述。具体而言，L 是可处理的，当且仅当平坦布局

$$\mathsf{sort}(L^\flat) = (s_1,\dots,s_m) : (d_1,\dots,d_m)$$

满足以下条件：对于每个 $1 \le i < m$，都有

1. $d_i = 0$，或 2.
   $s_i d_i$ 整除 d_{i+1}。

例 2.3.10.2. 布局

$$L = (((12))) : (((17)))$$

是可处理的。更一般地，任何长度为 1 的布局 L 都是可处理的。

例 2.3.10.3. 布局

$$L = ((2,4),32) : ((1,2),8)$$

是可处理的。更一般地说，任何列优先布局都是可处理的。

例 2.3.10.4. 该布局

$$L = (2, (4, 32)) : (128, (32, 1))$$

是可处理的。更一般地，任何行主序布局 L 都是可处理的。

例 2.3.10.5. 该布局

$$L = ((3, 3), (1, 3), (3, 1, 3)) : ((81, 1), (0, 8), (3, 0, 27))$$

是可处理的。更一般地，任何紧凑布局都是可处理的。

例 2.3.10.6. 该布局

$$L = ((3, 7, 7)) : ((0, 15, 0))$$

是可处理的。更一般地，恰好包含一个非零步幅项的任何布局都是可处理的。

例 2.3.10.7. 该布局

$$L = (2, (2, (2, 2))) : (1, (2048, (16, 64)))$$

是可处理的。更一般地，任何可补布局都是可处理的。

例 2.3.10.8. 该布局

$$L = ((8, 8), (5, 5)) : ((8, 1), (10, 2))$$

并非可处理的。具体而言，这表明可处理布局 L_1 andL_2 的串接 (L_1, L_2) 不一定是可处理的。

第3章

布局类别

在深入探讨布局代数之后，我们现在转向这项工作的数学核心：将布局实现为适当定义的范畴中的态射。在此过程中，我们构建了一套布局图的图形演算，从而能够更直接地进行布局运算。

3.1 Tuple范畴

在本节中，我们定义一个范畴 **Tuple**，其对象是正整数组成的元组，其态射称为元组态射。每个元组态射 $f : S \to T$ 都编码一个平坦布局 L_f。元组态射的复合与布局复合相容，也就是说，如果 f 和 g 是可复合元组态射，那么

$$L_{g \circ f} = L_g \circ L_f.$$

我们定义一个实现函子（定理 3.1.4.4）

$$|\cdot| : \mathbf{Tuple} \to \mathbf{FinSet}$$

它通过以下公式还原 L_f 的布局函数

$$|f| = \Phi_{L_f}^{\mathsf{size}(T)}.$$

我们构建了一套“元组态射代数”，其中包含排序（第3.1.5.3节）、合并（第3.1.5.4节）、补（第3.1.5.6节）、拼接（第3.1.5.5节）、扁平除法（第3.1.5.7节）以及扁平积（第3.1.5.8节）等运算，这些运算与平坦布局上的相应运算相容。

3.1.1 基本定义

定义 3.1.1.1 定义。令**Fin**$_*$ 表示这样一个范畴：其对象是带基点有限集

$$\langle m \rangle_* = \{*, 1, 2, \ldots, m\}$$

对于 $m \geq 0$，其态射 $\alpha : \langle m \rangle_* \to \langle n \rangle_*$ 是满足 $\alpha(*) = *$ 的函数。我们称这些态射为带基点映射，或简称映射。

旁注 3.1.1.2. **Fin**$_*$ 是有限基点集合范畴**FinSet**$_*$ 的一个骨架 。

记号 3.1.1.3. 如果基点映射 $\alpha : \langle m \rangle_* \to \langle n \rangle_*$ 的陪域是已知的，我们有时记作

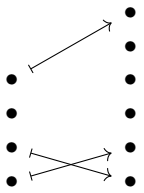
$$\alpha = (\alpha(1), \ldots, \alpha(m))$$

作为一个长度为 m 且条目属于 $\langle n \rangle_*$的元组。

例 3.1.1.4. 在 $\mathbf{Fin}_*$ 中有一个态射 $\alpha : \langle 4 \rangle_* \to \langle 6 \rangle_*$，定义如下:

$$\alpha = (2, 1, *, 6),$$

我们可以用下面的图表将其可视化。



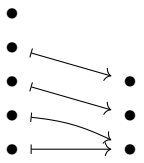
α

请注意，与条目 3 对应的圆点不支持箭头，这反映了它被映射到 $*$这一事实。

例 3.1.1.5. 在 $\mathbf{Fin}_*$ 中有一个态射 $\beta : \langle 5 \rangle_* \to \langle 3 \rangle_*$，定义如下:

$$\beta = (*, 1, 2, 3, *) \ ,$$

我们可以用下面的图表将其可视化。



β

例 3.1.1.6. 对于任意 $m \geq 0$，在 $\mathbf{Fin}_*$ 中，存在唯一一个形如 π 的态射: $\langle m \rangle_* \to \langle 0 \rangle_*$，即

$$\pi = (*, \ldots, *).$$

例 3.1.1.7. 对于任意 $n \geq 0$，在 $\mathbf{Fin}_*$ 中，存在唯一一个形如 δ 的态射: $\langle 0 \rangle_* \to \langle m \rangle_*$，即

$$\delta = ().$$

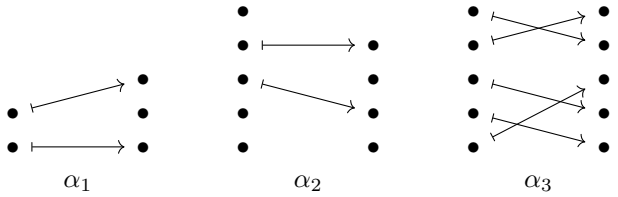
旁注 3.1.1.8. 范畴 $\mathbf{Fin}_*$ 是交换运算子的算子范畴，因此我们有时写作

$$\mathbf{Fin}_* = \mathbf{Comm}^{\otimes}.$$

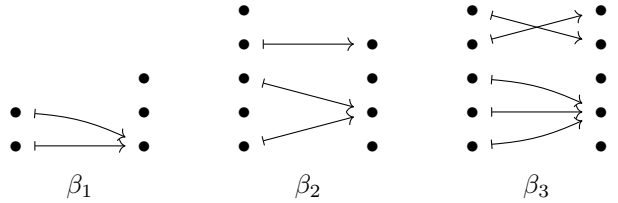
我们尤其关注可处理的 态射$\mathbf{Fin}_*$，其定义如下。

定义 3.1.1.9. 我们称基点映射 $\alpha : \langle m \rangle_* \to \langle n \rangle_*$ 是可处理的，如果对于任意 $j \in \langle n \rangle \subset \langle n \rangle_*$，原像 $\alpha^{-1}(j)$ 为空或仅包含一个元素。

例 3.1.1.10. 这些映射



是可处理的，而映射



则不可处理

注记 3.1.1.11. 若将态射 $\alpha : \langle m \rangle_* \to \langle n \rangle_*$ 在 $\mathbf{Fin}_*$ 中表示为一个 Tuple，即:

$$\alpha = (\alpha(1), \ldots, \alpha(m))$$

那么当且仅当 α 中每个正整数出现次数不超过一次时， α 才是可处理的带基点映射。

旁注 3.1.1.12. 宽子范畴

$$\mathbf{E}_0^{\otimes} \subset \mathbf{Comm}^{\otimes} = \mathbf{Fin}_*$$

由可处理的带基点映射构成，是 $\mathbf{E}_0$ 运算子的算子范畴。

定义 3.1.1.13。 令**Tuple**表示对象为正整数元组的范畴

$$S = (s_1, \ldots, s_m)$$

，其态射

$$f : (s_1, \ldots, s_m) \to (t_1, \ldots, t_n)$$

由一个可处理的带基点映射 $\alpha : \langle m \rangle_* \to \langle n \rangle_*$ 指定，并满足以下性质:

- 若 $1 \leq i \leq m$ 且 $\alpha(i) \neq *$，则 $s_i = t_{\alpha(i)}$ 。

我们称这样的态射 f 位于 α之上，并将 f 称为元组态射。

记号 3.1.1.14。 若 f: (s_1，……， s_m) $\to$ (t_1，……， t_n) 是一个位于 α之上的元组态射，则有时我们将 f 表示为

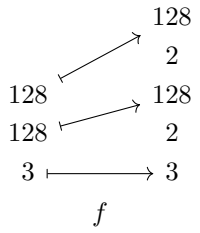
$$(s_1, \ldots, s_m) \xrightarrow[\alpha]{f} (t_1, \ldots, t_n).$$

我们所发展的布局图形演算基于**Tuple**中态射的自然可视化，如下例所示。

例 3.1.1.15. 元组态射

$$(3,128,128) \xrightarrow[(1,3,5)]{f} (3,2,128,2,128)$$

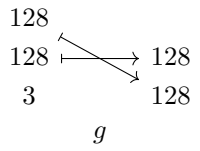
可以用下面的图表来表示。



例 3.1.1.16. 元组态射

$$(3,128,128) \xrightarrow[(*,2,1)]{g} (128,128)$$

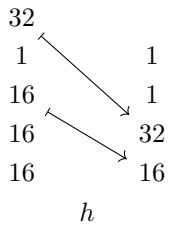
可以用下面的图表来表示。



例 3.1.1.17. 元组态射

$$(16,16,16,1,32) \xrightarrow[(*,*,1,*,2)]{h} (16,32,1,1)$$

可以用下面的图表来表示。



观察 3.1.1.18. 我们可以如下将范畴 **Tuple** 与一些熟知的运算子联系起来。令$\mathbb{Z}^{\mathsf{div}}_{>0}$ 表示正整数在整除关系下构成的偏序集，并将其视为一个对称幺半范畴，其中乘积由整数乘法给出。令 $(\mathbb{Z}^{\mathsf{div}}_{>0})^{\otimes}$ 表示 $\mathbb{Z}^{\mathsf{div}}_{>0}$ 的算子范畴。那么显然存在函子

$$\mathbf{Tuple} \to (\mathbb{Z}^{\mathsf{div}}_{>0})^{\otimes},$$

and

$$\mathbf{Tuple} \to \mathbf{E}_0^{\otimes},$$

使得该图表

$$\begin{array}{ccc} \mathbf{Tuple} & \longrightarrow & (\mathbb{Z}^{\mathsf{div}}_{>0})^{\otimes} \\ \downarrow & & \downarrow \\ \mathbf{E}_0^{\otimes} & \longrightarrow & \mathbf{Comm}^{\otimes} \end{array}$$

可交换。这表明 **Tuple** 是拉回算子的宽子范畴

$$\mathbf{Tuple} \subset \mathbf{E}_0^{\otimes} \times_{\mathbf{Comm}^{\otimes}} (\mathbb{Z}_{>0}^{\mathsf{div}})^{\otimes}$$

关于态射

$$(s_1, \ldots, s_m) \xrightarrow[\alpha]{f} (t_1, \ldots, t_n)$$

满足

$$\alpha(i) \neq 1 \quad \Rightarrow \quad s_i = t_{\alpha(i)}.$$

3.1.2 从元组态射到平坦布局

采用范畴 **Tuple** 进行研究的动因在于，每个元组态射 f 都编码了一个平坦布局L_f。此外，每个可处理布局 L 都能导出一个元组态射 f_L。我们将在定理 3.1.2.10中证明，这些构造在某种意义上互为逆映射，并且可处理布局恰好就是由元组态射编码的布局。

构造 3.1.2.1. 设

$$(s_1, \ldots, s_m) \xrightarrow[\alpha]{f} (t_1, \ldots, t_n)$$

是一个元组态射。我们定义 L_f 为形状为

$$\mathsf{shape}(L_f) = (s_1, \ldots, s_m)$$

是 f 的定义域，其步幅

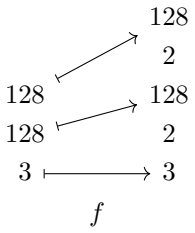
$$\mathsf{stride}(L_f) = (d_1, \ldots, d_m)$$

由以下公式定义

$$d_i = \begin{cases} 0 & \alpha(i) = * \\ \prod_{j<\alpha(i)} t_j & \alpha(i) \neq *. \end{cases}$$

我们将 L_f 称为由 编码的布局，或称为与 相关联的布局。

例 3.1.2.2。元组态射



例 3.1.1.15 编码的布局为

$$L_f = (3, 128, 128) : (1, 6, 1536).$$

注意，根据定理 3.1.2.1 中的公式计算步幅，实际上就是沿着从特定形状条目指向其目标条目的箭头，并将该条目下方的所有条目相乘（规定空乘积等于 1）。

例 3.1.2.3。元组态射

$$(3,128,128) \xrightarrow[(*,2,1)]{g} (128,128)$$

例3.1.1.16 编码了布局

$$L_g = (3,128,128) : (0,128,1).$$

例 3.1.2.4。元组态射

$$(16,16,16,1,32) \xrightarrow[(*,*,1,*,2)]{h} (16,32,1,1)$$

例3.1.1.17 编码了布局

$$L_h = (16,16,16,1,32) : (0,0,1,0,16).$$

我们已经了解了如何计算由元组态射 L_f 编码的平坦布局 f。另一方面，如果 L 是可处理的，那么我们可以反向进行，构造一个元组态射 f 来编码 L。回顾定义2.1.8.1 ，平坦布局 L 是可处理的当且仅当

$$\mathsf{sort}(L) = (s_1,\dots,s_m) : (d_1,\dots,d_m)$$

满足以下性质：

$$\text{If } 1 \le i < m, \text{ then } d_i = 0, \text{ or } s_i d_i \text{ divides } d_{i+1}.$$

构造 3.1.2.5。 假设 $L = (s_1,\dots,s_m) : (d_1,\dots,d_m)$ 是可处理的，并令

$$\mathsf{sort}(L) = (s'_1,\dots,s'_m) : (d'_1,\dots,d'_m),$$

因此存在某个置换 $\sigma \in \Sigma_m$ ，使得 $\mathsf{sort}(L) = L^\sigma$。换言之，对于每个 $1 \le i \le m$， $s'_i = s_{\sigma(i)}$ 且 $d'_i = d_{\sigma(i)}$。如果每个 d'_i 都非零，则令 $k = 0$。否则，令 k 为满足 $d'_k = 0$的最大整数。令 $\ell = 2(m-k)$，并令

$$(t'_1,\dots,t'_\ell) = \left(d'_{k+1}, s'_{k+1}, \frac{d'_{k+2}}{s'_{k+1}d'_{k+1}}, s'_{k+2}, \frac{d'_{k+3}}{s'_{k+2}d'_{k+2}}, \dots, \frac{d'_m}{s'_{m-1}d'_{m-1}}, s'_m\right).$$

我们定义

$$f'_L : (s_1,\dots,s_m) \to (t'_1,\dots,t'_\ell)$$

为位于映射 α 之上的元组态射：$\langle m\rangle_* \to \langle \ell\rangle_*$ ，其定义为

$$\alpha'(i) = \begin{cases} * & \sigma^{-1}(i) \le k \\ 2(\sigma^{-1}(i) - k) & k+1 \le \sigma^{-1}(i) \le m. \end{cases}$$

令 $J = \{j_1 < \cdots < j_n\} \subset \langle \ell\rangle$ 表示满足 j_i 为偶数或满足 $t_{j_i} \ne 1$的指标集合。令

$$(t_1,\dots,t_n) = (t'_{j_1},\dots,t'_{j_n}),$$

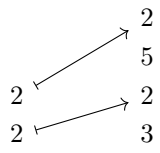
并令 $\iota : \langle n\rangle_* \to \langle \ell\rangle_*$ 为包含映射 $i \mapsto j_i$。根据构造，映射 α' 可分解为$\alpha' = \iota \circ \alpha$，我们将 L 的标准表示 定义为元组态射

$$(s_1,\dots,s_m) \xrightarrow[\alpha]{f_L} (t_1,\dots,t_n).$$

例 3.1.2.6. 如果

$$L = (2, 2) : (3, 30),$$

那么 L 是可处理的，且 L 的标准表示是元组态射



f_L

请注意，非正式地说，通过定理 f_L 计算 3.1.2.5 相当于

- 将陪域初始化为 (),
- 按递增顺序遍历 L 的非零步幅,
- 如果 d_j 是当前步幅，而 d_i 是此前遍历的步幅，则追加– (s_j)（若 $s_i d_i = d_j$），或– $\left(\begin{smallmatrix} d_j \\ s_i d_i \end{smallmatrix}, s_j\right)$（若 $s_i d_i < d_j$），并且

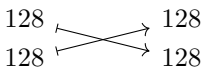
—

- 映射 $s_j \mapsto s_{j\circ}$

例 3.1.2.7. 如果

$$L = (128, 128) : (128, 1),$$

那么 L 是可处理的，且 L 的标准表示是元组态射

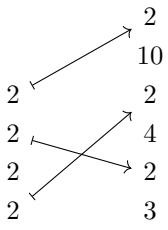


f_L

例 3.1.2.8. 如果

$$L = (2, 2, 2, 2) : (24, 0, 3, 480),$$

那么 L 是可处理的，且 L 的标准表示是元组态射



f_L

让我们说明，定理 f_L 3.1.2.5 的元组态射确实编码了布局 $L_\circ$

引理 3.1.2.9。 假设 L 是一个可处理的平坦布局，且 $f = f_L$ 是 L 的标准表示。那么由 f 编码的布局是

$$L_f = L.$$

证明。 假设 $L = (s_1, \ldots, s_m) : (d_1, \ldots, d_m)$ 可处理，并令

$$(s_1, \ldots, s_m) \xrightarrow[\alpha]{f} (t_1, \ldots, t_n)$$

是 L 的标准表示。显然

$$\mathsf{shape}(L_f) = (s_1, \ldots, s_m) = \mathsf{shape}(L).$$

我们需要检查 $\mathsf{stride}(L_f) = \mathsf{stride}(L)$。换言之，我们需要检查对于任意 $1 \leq i \leq m$，都有

$$d_i = \begin{cases} 0 & \alpha(i) = * \\ \prod_{j<\alpha(i)} t_j & \alpha(i) \neq *. \end{cases}$$

我们借用定理 3.1.2.5 的记号。若 $\alpha(i) = *$，则 $\alpha'(i) = *$，因此 $\sigma^{-1}(i) \leq k$。这意味着

$$d_i = d'_{\sigma^{-1}(i)} = 0.$$

Supp否则，设 $\alpha(i) \neq *$。那么 $\alpha'(i) \neq *$，因此 $k+1 \leq \sigma^{-1}(i) \leq m$。我们计算

$$\begin{aligned} \prod_{j<\alpha(i)} t_j = \prod_{\substack{j'<\alpha'(i) \\ t'_{j'} \neq 1}} t'_{j'} = \prod_{j'<\alpha'(i)} t'_{j'} = \prod_{j'<2(\sigma^{-1}(i)-k)} t'_{j'} &= d'_{k+1} \cdot \left(\prod_{v=1}^{\sigma^{-1}(i)-(k+1)} s'_{k+v} \frac{d'_{k+v+1}}{s'_{k+v} d'_{k+v}} \right) \\ &= d'_{\sigma^{-1}(i)} \\ &= d_i. \end{aligned}$$

□

我们已经证明，若 L 是可处理的平坦布局，则存在一个元组态射 f，它编码了 L。接下来，我们证明其逆命题，这意味着可处理的平坦布局恰好是由元组态射编码的布局。

命题 3.1.2.10。 设 L 是一个平坦布局。当且仅当 L 是可处理的，才存在一个编码 L的元组态射 f。

证明。 首先，假设 L是一个平坦布局，并且 $f: (s_1、\ldots\ldots、s_m) \to (t_1、\ldots\ldots、t_n)$ 是一个满足 $L_f = L$的元组态射。我们希望证明 L_f 是可处理的。令

$$\mathsf{sort}(L) = (s'_1, \ldots, s'_m) : (d_1, \ldots, d_m)$$

为 L的排序，并假设 $1 \leq i < m$。我们将论证 $d_i = 0$，或者 $s'_i d_i$ 整除 d_{i+1}。如果$d_i = 0$，则结论成立。否则，假设 $d_i \neq 0$。于是

$$d_i = \prod_{j<k} t_j$$

对于某个 $1 \leq k \leq n$ 满足 $s'_i = t_k$。由于 $d_{i+1} \geq d_i$，我们知道 $d_{i+1} \neq 0$，因此 d_{i+1} 具有以下形式

$$d_{i+1} = \prod_{j<\ell} t_j$$

对于某个 $1 \leq \ell \leq n$。需要考虑两种情形:

- (情形 1) 如果 $\ell > k$，那么

$$d_{i+1} = \prod_{j<\ell} t_j = \left(\prod_{j\le k} t_j\right)\left(\prod_{k<j<\ell} t_j\right) = s_i' d_i \left(\prod_{k<j<\ell} t_j\right),$$

因此 $s_i' d_i$ 整除 d_{i+1}。

- （情形 2）若 $\ell \le k$，则由于

$$\prod_{j<\ell} t_j = d_{i+1} \ge d_i = \prod_{j<k} t_j,$$

我们必有

$$t_\ell = \cdots = t_{k-1} = 1,$$

and

$$d_{i+1} = d_i.$$

特别地，我们有 $s_{i+1}' = t_\ell = 1$。但由于 sort(L_f) 已排序且 $d_{i+1} = d_i$，我们有$s_i' \le s_{i+1}' = 1$，因此 $s_i' = 1$。由此可推得

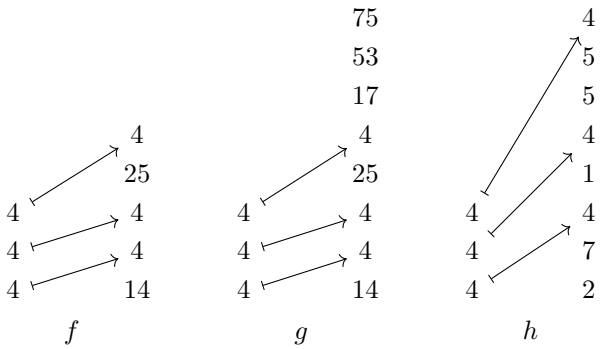
$$s_i' d_i = d_{i+1},$$

因此特别地， $s_i' d_i$ 整除 d_{i+1}。

我们得出结论： L 是可处理的。

接下来，假设 L 是可处理的。那么我们可以取 $f = f_L$ 作为 L的标准表示（参见构造 3.1.2.5），此时根据引理 3.1.2.9，有 $L = L_f$。 □

注记 3.1.2.11。 需要注意的是，有许多不同的元组态射会产生相同的布局。例如，下面所示的每个元组态射



都编码了布局

$$L_f = L_g = L_h = (4,4,4) : (14, 56, 5600).$$

其中， f 最简单： f 的像上方没有多余条目（与 g不同），而未被 f 命中的条目则是紧凑排列的（与 h不同）。为了精确描述 f 在这些态射中的简单性，我们引入标准形式这一概念。

定义 3.1.2.12. 假设

$$(s_1, \ldots, s_m) \xrightarrow[\alpha]{f} (t_1, \ldots, t_n)$$

是一个元组态射。如果满足以下条件，我们就称 f 具有标准形式：

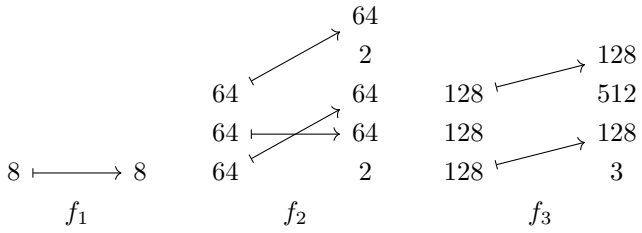
1. 如果 $n > 1$，那么 $n \in$ 像(α)。

2. 如果 $1 \le j < n$，那么

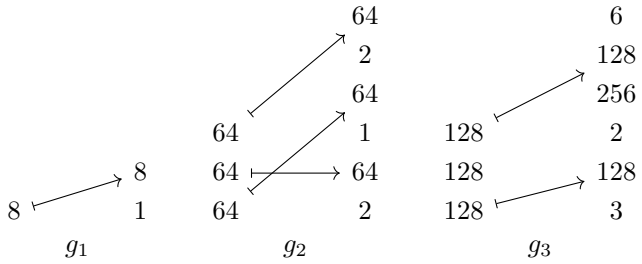
$$j \notin \mathsf{Image}(\alpha) \quad \Rightarrow \quad \begin{array}{c} t_j \neq 1, \text{ and} \\ j+1 \in \mathsf{Image}(\alpha) \end{array}$$

例 3.1.2.13. 注记 f 中的元组态射 3.1.2.11 具有标准形式，而 g 和 h 则没有。

例 3.1.2.14. 元组态射



具有标准形式，而元组态射

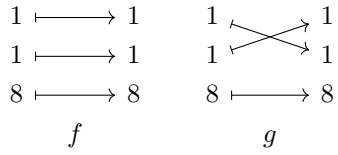


则没有。

例 3.1.2.15. 如果 L 是一个可处理布局，那么根据构造，L的标准表示 f_L 具有标准形式。

如果我们只考虑标准形式的元组态射，那么它们与可处理布局几乎一一对应。不过，有一种有问题的情况需要排除，接下来的例子将对此作出说明。

例 3.1.2.16. 考虑下面所示的元组态射 f 和 g 。



两者 f 和 g 都具有标准形式，并且

$$L_f = (8,1,1) : (1,8,8) = L_g.$$

这个例子说明，存在形如 $s_i = 1$ 和 $\alpha(i) \neq *$ 的条目，可能导致标准形式的表示元组态射不唯一。在布局方面，这对应于步幅为 $d_i \neq 0$ 的形状条目 $s_i = 1$ 。为了排除这类病态例子，我们引入非退化性的概念。

定义 3.1.2.17. 假设

$$(s_1,\dots,s_m) \xrightarrow[\alpha]{f} (t_1,\dots,t_n)$$

是一个元组态射，并且

$$L = (s_1,\dots,s_m) : (d_1,\dots,d_m)$$

是一个平坦布局。

1. 我们称 f 是非退化的，如果

$$s_i = 1 \quad \Rightarrow \quad \alpha(i) = *.$$

2. 我们称 L 是非退化的，如果

$$s_i = 1 \quad \Rightarrow \quad d_i = 0.$$

观察 3.1.2.18。 如果 f 是一个非退化元组态射，那么由 f 编码的布局 L_f 是非退化的。反之，如果 L 是一个非退化平坦布局，那么 L 的标准表示 f_L是非退化的。

观察 3.1.2.19。 限制到非退化平坦布局并不会真正损失一般性。如果 L 是任意平坦布局，那么 filter(L) 是一个非退化平坦布局，并且与 L具有相同的坐标函数和布局函数。

标准形式的非退化元组态射的本质性质在于，它们由其所编码的布局刻画。下面将对此作出精确说明。

引理 3.1.2.20. 设 f 和 g 是标准形式的非退化元组态射。若 $L_f = L_g$，则 $f = g$。

证明。设

$$(s_1,\dots,s_m) \xrightarrow[\alpha]{f} (t_1,\dots,t_n)$$

and

$$(s_1,\dots,s_m) \xrightarrow[\beta]{g} (u_1,\dots,u_p)$$

是满足以下条件的标准形式非退化元组态射:

$$L_f = (s_1,\dots,s_m) : (d_1,\dots,d_m) = L_g.$$

我们要证明 $f = g$。首先，我们将论证 $(t_1,\dots,t_n) = (u_1,\dots,u_p)$。令

$$X = \{t_1 \cdots t_j \mid 1 \le j \le n\}$$
$$Y = \{u_1 \cdots u_k \mid 1 \le k \le p\}$$

分别表示 $(t_1,\dots,t_n)$ 和 $(u_1,\dots,u_p)$ 的前缀乘积集合。我们断言 $X = Y$，因为这两个集合都等于

$$Z = \{d_i, s_i d_i \mid 1 \le i \le m \text{ and } d_i \neq 0\}.$$

我们来论证 $X = Z$。假设 $1 \le j \le n$。若存在某个 $i \in \langle m \rangle$，满足 $\alpha(i) = j$，那么$t_1 \cdots t_j = s_i d_i$。另一方面，如果 j不在 α 的像中，那么由于 f具有标准形式，存在某个 $i \in \langle m \rangle$，使得 $\alpha(i) = j+1$；在这种情况下， $t_1 \cdots t_j = d_i$。这证明了 $X \subseteq Z$。

反过来，如果 $1 \leq i \leq m$ 且 $d_i \neq 0$，那么 $d_i = t_1 \cdots t_{\alpha(i)-1}$ 且 $s_i d_i = t_1 \cdots t_{\alpha(i)}$，由此证明了 $Z \subseteq X$。我们推得 $X = Z$。同样的论证也证明了 $Y = Z$。

由于 f 和 g 都是非退化的标准形式，因此我们知道每个 t_j 和每个 u_k 都大于 1，这意味着

$$t_1 < t_1 t_2 < \cdots < t_1 \cdots t_n,$$
$$u_1 < u_1 u_2 < \cdots < u_1 \cdots u_p,$$

又由于 $X = Y$，可得 $n = p$，且对于每个 $1 \leq j \leq n$，都有 $t_1 \cdots t_j = u_1 \cdots u_j$。由此我们推得 $(t_1, \ldots, t_n) = (u_1, \ldots, u_p)$。

接下来，我们需要证明 $\alpha = \beta$。反设存在某个 $i \in \langle m \rangle$，使得$\alpha(i) \neq \beta(i)$。需要考虑两种情况。

- 如果 $\alpha(i) = * \neq \beta(i)$，那么
$$0 = d_i = t_1 \cdots t_{\beta(i)-1},$$
矛盾。情形 $\alpha(i) \neq * = \beta(i)$ 类似。
- 如果 $\alpha(i) \neq * \neq \beta(i)$，那么不失一般性，我们可以假设 $\alpha(i) < \beta(j)$，此时
$$d_i = t_1 \cdots t_{\alpha(i)-1} < t_1 \cdots t_{\beta(i)-1} = d_i,$$
矛盾。

我们推得 $\alpha = \beta$，因此 $f = g$。 □

现在我们可以证明对应定理，该定理将标准形式的非退化元组态射与非退化可处理平坦布局对应起来。

定理 3.1.2.21. 这些映射

$$f \longmapsto L_f$$

$$\left\{ \begin{array}{c} \text{非退化} \\ \text{元组态射的} \\ \text{标准形式} \end{array} \right\} \longleftrightarrow \left\{ \begin{array}{c} \text{非退化} \\ \text{可处理的平坦} \\ \text{布局} \end{array} \right\}$$

$$f_L \longleftarrow L$$

构造 *3.1.2.1* 和 *3.1.2.5* 确定标准形式的非退化元组态射与非退化可处理平坦布局之间的一一对应关系。

证明。 我们要证明，构造 $f \mapsto L_f$ 和 $L \mapsto f_L$ 在限制到指定形式的元组态射和布局时互为逆映射。若 L 是一个非退化、可处理的平坦布局，则根据引理 3.1.2.9 可得 $L_{f_L} = L$。接下来，假设 f 是一个标准形式的非退化元组态射，且 $L = L_f$ 是由 f编码的布局。由于 f 和 f_{L_f} 都是标准形式的非退化元组态射，且由这些元组态射编码的布局相等，因此根据引理3.1.2.20 可得 $f = f_{L_f}$。

□

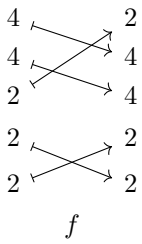
3.1.3 示例

本节将介绍若干重要的元组态射族，并描述它们所生成的平坦布局。

例 3.1.3.1（恒等态射）。 如果某个元组态射 f 满足$f = \mathrm{id}_S$，其中 S是某个元组，则称其为恒等态射。若 $f = \mathrm{id}_S$ 是恒等态射，则 L_f 是形状为 S的列主序布局。例如，下面给出了一个恒等态射 f 及其相关布局 L_f。

$$\begin{array}{ccc} 4 \longmapsto 4 & & \\ 4 \longmapsto 4 & & \\ 2 \longmapsto 2 & & \\ 2 \longmapsto 2 & \rightsquigarrow & L_f = (2,2,2,4,4) : (1,2,4,8,32) \\ 2 \longmapsto 2 & & \\ f & & \end{array}$$

例 3.1.3.2（同构）。 如果存在一个元组态射 $g\colon T \to S$，使得 $g \circ f = \mathrm{id}_S$ 且 $f \circ g = \mathrm{id}_T$，则元组态射 $f\colon S \to T$ 是同构。若 f 是同构，则其相关布局 L_f 是紧致的。例如，下面给出了一个同构 f 及其相关布局 L_f。



$$\rightsquigarrow \qquad L_f = (2,2,2,4,4) : (2,1,64,4,16)$$

f

.

观察 3.1.3.3. 请注意，如果一个元组态射

$$(s_1, \ldots, s_m) \xrightarrow[\alpha]{f} (t_1, \ldots, t_n)$$

是同构，则 $\alpha : \langle m \rangle_* \to \langle m \rangle_*$ 是双射，因此 $\alpha \mid_{\langle m \rangle} \in \Sigma_m$ 是一个置换。反过来，如果 $\sigma \in \Sigma_m$ 是一个置换，且$(s_1, \ldots, s_m)$是一个正整数元组，那么我们可以构造同构

$$(s_1, \ldots, s_m) \xrightarrow[\sigma_*]{f} (s_{\sigma(1)}, \ldots, s_{\sigma(m)}).$$

由此可得，定义域为 $(s_1, \ldots, s_m)$ 的元组同构 f 与 Σ_m 中的置换一一对应。

例 3.1.3.4（投影）。 假设 $S = (s_1, \ldots, s_m)$ 是一个形状，并且假设

$$\{i_1 < \cdots < i_r\} \subset \langle m \rangle$$

是某个子集。令

$$(s_1, \ldots, s_m) \xrightarrow[\alpha]{p} (s_{i_1}, \ldots, s_{i_r})$$

为位于映射 α 之上的元组态射，其

$$\alpha(x) = \begin{cases} j & x = i_j \\ * & \text{else.} \end{cases}$$

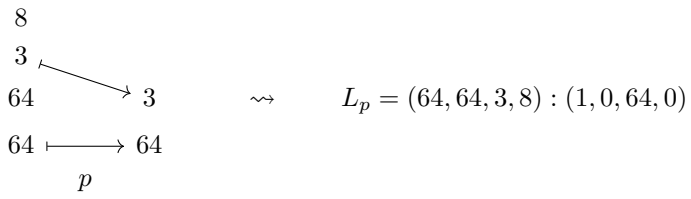
我们将 $p(s_1, \ldots, s_m)$ 到 $(s_{i_1}, \ldots, s_{i_r})$ 的投影称为。由 p 编码的布局 is

$$L_p = (s_1, \ldots, s_m) : (d_1, \ldots, d_m),$$

其中

$$d_i = \begin{cases} s_{i_1} \cdots s_{i_{j-1}} & i = i_j \text{ for some } 1 \le j \le r \\ 0 & \text{otherwise.} \end{cases}$$

例如，下面给出了一个将 $(64, 64, 3, 8)$ 投影到 $(64, 3)$ 的投影 p，以及其相关布局。



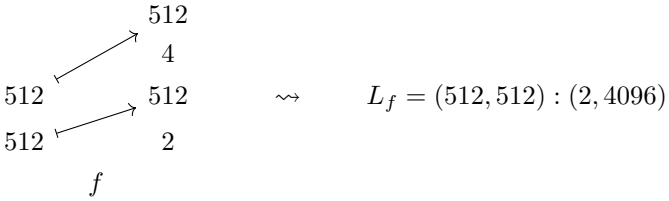
例 3.1.3.5（膨胀）。设 $S = (s_1, \ldots, s_m)$ 是一个形状，并设 $c_1, \ldots, c_m$ 是正整数。元组态射

$$(s_1, \ldots, s_m) \xrightarrow[(*,2,*,4,\ldots,*,2m)]{f} (c_1, s_1, \ldots, c_m, s_m)$$

称为 $(s_1, \ldots, s_m)$ 按 $(c_1, \ldots, c_m)$ 膨胀。与此态射相关的布局 L_f 为 $L_f = (s_1, \ldots, s_m) : (d_1, \ldots, d_m)$，其中

$$d_i = \prod_{j<i} c_j s_j.$$

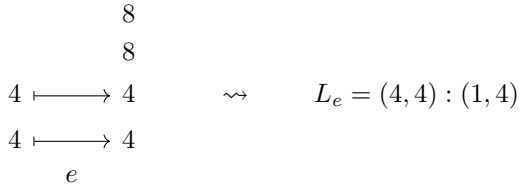
例如, 因此，这里是将 $f(512, 512)$ 按 $(2, 4)$ 进行膨胀后得到的结果，以及其相关布局 .



例 3.1.3.6 (扩张)。 设 $S = (s_1, \ldots, s_m)$ 是一个正整数元组，并设 $1 \le i \le m'$，使得 $S' = (s_1, \ldots, s_{m'})$ 整除 S。那么，元组态射

$$(s_1, \ldots, s_{m'}) \xrightarrow[(1,2,\ldots,m')]{e} (s_1, \ldots, s_{m'}, \ldots, s_m)$$

称为 S' 到 S的扩张 。 e 编码的布局是形状为（$s_1, \ldots, s_{m'}$）的列主序布局。例如，下面是将 $S' = (4, 4)$扩张为 $S = (4, 4, 8, 8)$的示例。



扩张的一个重要性质是：如果 $f\colon S \to T$ 是任意元组态射，并且 $e\colon T \to T'$ 是一个扩张，那么

$$L_{e\circ f} = L_f.$$

换句话说，与扩张 f 后复合，不会改变 f所编码的布局。

例 3.1.3.7（限制）。假设

$$(s_1, \ldots, s_m) \xrightarrow[\alpha]{f} (t_1, \ldots, t_n)$$

是一个元组态射，并假设

$$I = \{i_1 < \cdots < i_r\} \subset \langle m \rangle$$

是一个指标子集。那么元组态射

$$(s_{i_1}, \ldots, s_{i_r}) \xrightarrow[\alpha\circ\iota]{f|_I} (t_1, \ldots, t_n)$$

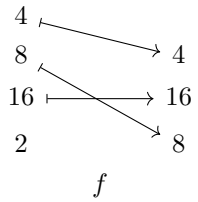
称为 f 到 I的限制。如果 f 所编码的布局是

$$L_f = (s_1, \ldots, s_m) : (d_1, \ldots, d_m),$$

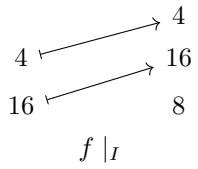
那么 $f\,|_I$ 所编码的布局是

$$L_{f|_I} = (s_{i_1}, \ldots, s_{i_r}) : (d_{i_1}, \ldots, d_{i_r}).$$

例如，下面是一个元组态射 f 的限制 $f\,|_I$，其中 $I = \{2, 4\}$。



f $\rightsquigarrow$ $L_f = (2, 16, 8, 4) : (0, 8, 1, 128)$



$f\,|_I$ $\rightsquigarrow$ $L_{f|_I} = (16, 4) : (8, 128)$

例 3.1.3.8（条目包含）。前述构造的一个重要特例如下。若 $f\colon (s_1, \ldots, s_m) \to (t_1, \ldots, t_m)$ 是一个元组态射，且 $1 \le i \le m$，则 i 个条目 f_i 的f 为

$$(s_i) \xrightarrow[(i)]{f_i} (t_1, \ldots, t_n)$$

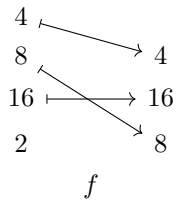
如果 f 编码的布局是

$$L_f = (s_1, \ldots, s_m) : (d_1, \ldots, d_m),$$

那么由 f_i 编码的布局是

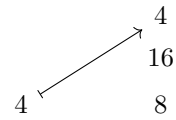
$$L_{f_i} = (s_i) : (d_i).$$

例如，这里有一个元组态射 f，以及它的第四个条目 f_4。



f

$\rightsquigarrow \qquad L_f = (2, 16, 8, 4) : (0, 8, 1, 128)$



f_4

$\rightsquigarrow \qquad L_{f_4} = (4) : (128)$

注记 3.1.3.9。 给定一个$\langle n\rangle_* \in \mathbf{Fin}_*$，对于每个 $i \in \langle n\rangle$，存在一个态射 $\varphi_i : \langle 1\rangle_* \to \langle n\rangle_*$，将其映至 $* \mapsto *$和 $1 \mapsto i$。对于一个位于 $\alpha : \langle m\rangle_* \to \langle n\rangle_*$之上的元组态射 $f : (s_1, \ldots, s_m) \to (t_1, \ldots, t_n)$，第 i 个条目位于复合映射 $\alpha \circ \varphi_i : \langle 1\rangle_* \to \langle n\rangle_*$之上。

例 3.1.3.10（分解）。 假设

$$(s_1, \ldots, s_m) \xrightarrow[\alpha]{f} (t_1, \ldots, t_n)$$

是一个元组态射，并假设

$$J = \{j_1 < \cdots < j_\ell\} \subset \langle n\rangle$$

是一个子集，使得 Image(α) $\subseteq J \cup \{*\}$。若将 $\iota : \langle \ell\rangle_* \to \langle n\rangle_*$ 记作映射 $k \mapsto j_k$，则 α可分解为 $\alpha = \iota \circ \bar{\alpha}$，其中映射 $\bar{\alpha} : \langle m\rangle_* \to \langle \ell\rangle_*$唯一确定；我们将将 f 经 J分解定义为元组态射

$$(s_1, \ldots, s_m) \xrightarrow[\bar{\alpha}]{f|^J} (t_{j_1}, \ldots, t_{j_\ell}).$$

如果 f 编码的布局是

$$L_f = (s_1, \ldots, s_m) : (d_1, \ldots, d_m),$$

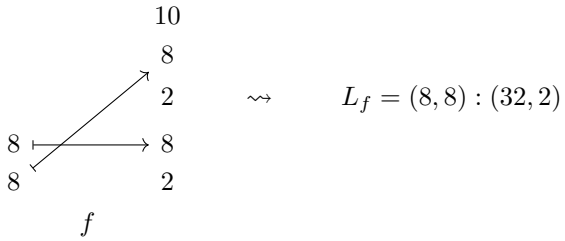
那么 $f\,|^J$ 编码的布局是

$$L_{f|^J} = (s_1, \ldots, s_m) : (d'_1, \ldots, d'_m),$$

其中

$$d'_i = \frac{d_i}{\left(\prod_{k<\alpha(i) \text{ and } k\notin J} t_j\right)}.$$

例如，这是元组态射 f 的分解 $f|^J$，其中 $J=\{2,\ 4,\ 5\}$。



$$\rightsquigarrow \qquad L_f=(8,8):(32,2)$$

$$\rightsquigarrow \qquad L_{f|^J}=(8,8):(8,1)$$

注记 3.1.3.11。 分解有一种范畴解释。借用例 3.1.3.10中的记号，我们可以观察到，存在一个元组态射 $i:(t_{j_1},\dots,t_{j_\ell})\to(t_1,\dots,t_n$，它位于 ι之上，并且 $f|^J$ 是 f 沿着 i的拉回:

$$\begin{array}{ccc}(s_1,\dots,s_m) & \xrightarrow{f|^J} & (t_{j_1},\dots,t_{j_\ell}) \\ \downarrow{\scriptstyle \mathsf{id}} & \lrcorner & \downarrow{\scriptstyle i} \\ (s_1,\dots,s_m) & \xrightarrow[f]{} & (t_1,\dots,t_n)\end{array}$$

3.1.4 元组态射的实现

如我们所见，元组态射 $f:\ S\to T$ 编码了一个平坦布局 L_f。在本节中，我们将构造一个实现函子

$$|\cdot|:\mathbf{Tuple}\to\mathbf{FinSet}.$$

使这种编码显式化。实现函子 $|\cdot|$ 将元组态射 f 映射为 L_f 的布局函数 $|f|$。为了构造实现函子 $|\cdot|$，我们先构造一个辅助函子

$$F:\mathbf{Tuple}\to\mathbf{FinSet}$$

我们将在构造中使用它。

构造 3.1.4.1. 我们定义一个函子

$$F:\mathbf{Tuple}\to\mathbf{FinSet}$$

如下。

- 对于 **Tuple** 中的对象 $S=(s_1,\dots,s_m)$，我们定义

$$FS=[0,S)=\prod_{i=1}^{m}[0,s_i).$$

- 对于 **Tuple** 中位于 α 之上的态射 $f\colon (s_1,\dots,s_m) \to (t_1,\dots,t_n)$，我们将 Ff 定义为映射

$$[0,S) \xrightarrow{Ff} [0,T)$$

由……给出

$$(Ff)(x_1,\dots,x_m) = (y_1,\dots,y_n)$$

其中

$$y_j = \begin{cases} x_i & \text{there exists } 1 \le i \le m \text{ with } \alpha(i) = j, \\ 0 & \text{else.} \end{cases}$$

On我们很容易验证 $F(g\circ f) = Fg\circ Ff$and $F\mathrm{id}_S = \mathrm{id}_{FS}$，因此 F实际上是一个函子 r.

例 **3.1.4.2.** 设 $f\colon (4,4) \to (4,4,4)$ 是位于 $\alpha = (1,3)$ 之上的元组态射。该 n

$$Ff : [0,(4,4)) \to [0,(4,4,4))$$

由……给出

$$(Ff)(x_1,x_2) = (x_1,0,x_2).$$

例 3.1.4.3. 设 $g\colon (3,256,256,512) \to (3,256,256)$ 是位于 $\beta = (*,3,2,*)$ 之上的元组态射。那么

$$Fg : [0,(3,256,256,512)) \to [0,(3,256,256))$$

由下式给出

$$(Fg)(x_1,x_2,x_3,x_4) = (0,x_3,x_2).$$

构造 3.1.4.4. 我们定义一个函子

$$|\cdot| : \textbf{Tuple} \to \textbf{FinSet}$$

如下。

- 对于 **Tuple** 中的对象 $S = (s_1,\dots,s_m)$，我们定义

$$|S| = [0,\mathsf{size}(S)) = \{0,1,\dots,\mathsf{size}(S)-1\}.$$

- 对于元组态射 $f\colon S \to T$，我们定义

$$|f| = \mathsf{colex}_T \circ Ff \circ \mathsf{colex}_S^{-1}$$

（回顾定理 2.1.2.18）。

如果 $f\colon S \to T$和 $g\colon T \to U$ 是可复合元组态射，那么

$$\begin{aligned} |g\circ f| &= \mathsf{colex}_U \circ F(g\circ f) \circ \mathsf{colex}_S^{-1} \\ &= \mathsf{colex}_U \circ Fg \circ Ff \circ \mathsf{colex}_S^{-1} \\ &= \mathsf{colex}_U \circ Fg \circ \mathsf{colex}_T^{-1} \circ \mathsf{colex}_T \circ Ff \circ \mathsf{colex}_S^{-1} \\ &= |g| \circ |f| \end{aligned}$$

并且如果 $f = \mathsf{id}_S$ 是恒等态射，那么

$$\begin{aligned}|\mathsf{id}_S| &= \mathsf{colex}_S \circ F\mathsf{id}_S \circ \mathsf{colex}_S^{-1} \\ &= \mathsf{colex}_S \circ \mathsf{id}_S \circ \mathsf{colex}_S^{-1} \\ &= \mathsf{colex}_S \circ \mathsf{colex}_S^{-1} \\ &= \mathsf{id}_{|S|}\ ,\end{aligned}$$

因此 $|\cdot|$ 确实指定了一个函子。接下来，我们注意到，对于一个态射 $f \text{ in } \mathbf{Tuple}$，映射 $|f|$ 是 L_f的布局函数。这使我们可以轻松推导出，**Tuple** 中态射的复合与平坦布局的复合相容（见推论3.1.4.6）。

引理 3.1.4.5。 如果 $f:\ S \to T$ 是一个元组态射，那么 f 的实现 $|f|$ 是 L_f的布局函数:

$$|f| = \Phi_{L_f}^{\mathsf{size}(T)}$$

证明。 令 $S = (s_1, \ldots, s_m)$、 $T = (t_1, \ldots, t_n)$，并令

$$L_f = (s_1, \ldots, s_m) : (d_1, \ldots, d_m)$$

表示与 f关联的布局，其步长 d_i 由以下公式定义

$$d_i = \begin{cases} 0 & \alpha(i) = * \\ \prod_{j<\alpha(i)} t_j & \text{else.} \end{cases}$$

通过与 colex_S进行前复合: $\prod_{i=1}^m [0, s_i) \to [0, \text{大小}(S))$，只需证明对于任意 $(x_1, \ldots, x_m) \in \prod_{i=1}^m [0, s_i)$，我们有

$$(\mathsf{colex}_T \circ Ff)(x_1, \ldots, x_m) = (x_1, \ldots, x_m) \cdot (d_1, \ldots, d_m).$$

对于一般输入 $(x_1, \ldots, x_m) \in \prod_{i=1}^m [0, s_i)$，我们有

$$(Ff)(x_1, \ldots, x_m) = (y_1, \ldots, y_n)$$

其中，若 $\alpha(i) = j$，则 y_j 等于 x_i；否则为 0。由此可得

$$\begin{aligned}(\mathsf{colex}_T \circ Ff)(x_1, \ldots, x_m) &= (y_1, \cdots, y_n) \cdot (1, t_1, \ldots, t_1 \cdots t_{n-1}) \\ &= \sum_{j=1}^{n} y_j \cdot t_1 \cdots t_{j-1} \\ &= \sum_{i=1}^{m} x_i d_i \\ &= (x_1, \ldots, x_m) \cdot (d_1, \ldots, d_m),\end{aligned}$$

如所需。 □

推论 3.1.4.6. 如果 f 和 g 是非退化的可复合元组态射，那么

$$L_{g \circ f} = L_g \circ L_f$$

证明。 设 $f : S \to T$ 和 $g:\ T \to U$ 是 **Tuple** 中分别位于 α 和 β 之上的态射。记作 $S = (s_1, \ldots, s_m)$ 和 $T = (t_1, \ldots, t_n)$。我们需要验证

1. 形状($L_{g\circ f}$) 细化 形状(L_f)：这是成立的，因为 L_f 和 $L_{g\circ f}$ 的形状都等于 S。

2. $L_{g\circ f}$ 在形状 shape(L_f) 上合并：这是因为元组态射 $g\circ f$ 是非退化的，因此布局 $L_{g\circ f}$ 也是非退化的。

3. $\Phi_{L_{g\circ f}} = \Phi_{L_g} \circ \Phi^{\mathsf{size}(L_g)}{}_{L_f}$：根据引理 3.1.4.5，可得

$$\begin{aligned}\Phi^{\mathsf{size}(U)}_{L_{g\circ f}} &= |g\circ f| \\ &= |g| \circ |f| \\ &= \Phi^{\mathsf{size}(U)}_{L_g} \circ \Phi^{\mathsf{size}(T)}_{L_f}.\end{aligned}$$

再与包含映射 $[0, \mathsf{size}(U)) \subset \mathbb{Z}$ 后复合，并注意到 $\mathsf{size}(T) = \mathsf{size}(L_g)$，即可得出结果。

□

3.1.5 元组态射上的运算

我们的下一个目标是构建“元组态射代数”，其中包含合并、补、复合、扁平除法和平坦积等运算。我们将证明，这些运算中的每一种都与平坦布局上的相应运算相兼容。

3.1.5.1 和

元组态射 f 和 g 的和 $f\oplus g$ 是通过连接 f 和 g的定义域与陪域得到的。为了精确定义这一运算，我们先定义 $\mathbf{Fin}_*$ 中态射上的相应运算。

定义 3.1.5.1. </style}设： $m\rangle_* \to \langle n\rangle_*$ 和 $\beta : \langle p\rangle_* \to \langle q\rangle_*$ 是 $\mathbf{Fin}_*$ 中的态射。我们将 α 与 β 的和 定义为态射

$$\alpha\oplus\beta : \langle m+p\rangle_* \to \langle n+q\rangle_*$$

由下式给出

$$(\alpha\oplus\beta)(x) = \begin{cases}\alpha(x) & 1\le x\le m \\ n+\beta(x-m) & m+1\le x\le m+p \\ * & x=*.\end{cases}$$

此运算满足结合律，因此对于 $\mathbf{Fin}_*$ 中任意有限集合的态射 $\alpha_1,\dots,\alpha_k$，我们可以考虑它们的和 $\alpha_1\oplus\cdots\oplus\alpha_k$。

注记 3.1.5.2. 如果 α 和 β 是可处理的带基点映射，那么 $\alpha\oplus\beta$ 是可处理的。

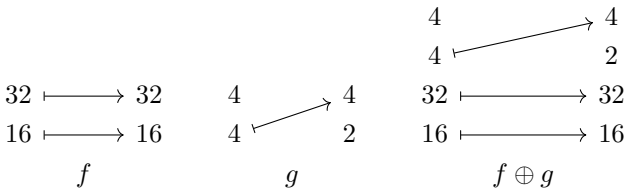
现在我们可以定义**Tuple**中的态射和。

定义 3.1.5.3. 设 $f: S\to T$ 和 $g: U\to V$ 是分别位于 α 和 β之上的元组态射。我们将 f 和 g 的和定义为以下元组态射

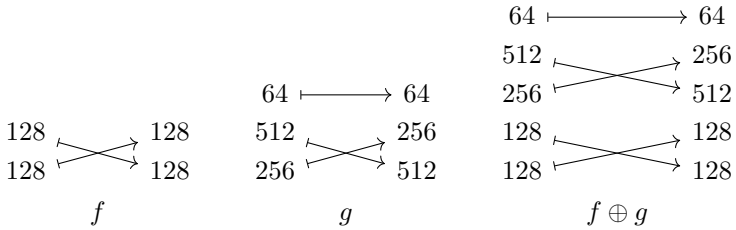
$$f\oplus g : S\star U \to T\star V$$

位于 $\alpha\oplus\beta$之上。此运算满足结合律，因此对于 **Tuple** 中任意有限个态射 f_1，……，f_k，我们可以考虑它们的和 $f_1\oplus\cdots\oplus f_k$。

例 3.1.5.4。 下面给出一个元组态射 f 与 g的和 $f \oplus g$ 的例子。



Ex例 3.1.5.5. 这是元组态射 $f \oplus g$ 和 f 之和的另一个例子 g.



注记 3.1.5.6. 元组态射之和有一种范畴解释：若 $f : S \to T$ 和$g : U \to V$ 是元组态射，则

$$f \oplus g : S \star U \to T \star V$$

是箭头范畴 Ar(**Tuple**)中 f 和 g 的余积。

3.1.5.2 挤压

我们常常需要从元组中移除所有整数 1。这可以通过挤压 函子来实现。

定义 3.1.5.7. 我们定义一个函子

$$\mathbf{Tuple} \xrightarrow{\mathsf{squeeze}(-)} \mathbf{Tuple}$$

如下。若 $S = (s_1, \dots, s_m)$ 是 **Tuple** 中的一个对象，我们定义

$$\mathsf{squeeze}(S) = (s_{i_1}, \dots, s_{i_k})$$

其中，$\{i_1 < \cdots < i_k\} \subset \langle m \rangle$ 是满足 $s_{i_j} \neq 1$ 的指标。若 $f : (s_1, \dots, s_m) \to (t_1, \dots, t_n)$ 是一个元组态射，我们定义

$$\mathsf{squeeze}(f) : \mathsf{squeeze}(S) \to \mathsf{squeeze}(T)$$

为元组态射

$$\mathsf{squeeze}(f) = (f\mid_I)\mid^J$$

其中， $f\mid_I$ 是 f 到.....的限制

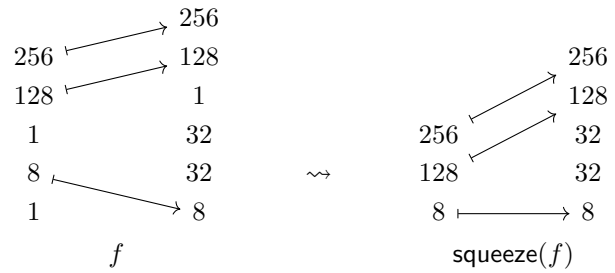
$$I = \{i \in \langle m \rangle \mid s_i \neq 1\}$$

如定义 3.1.3.7中所述，且其中 $(f\mid_I)\mid^J$ 是 经过

$$J = \{j \in \langle n \rangle \mid t_j \neq 1\},$$

如定义 3.1.3.10。

例 3.1.5.8。 以下是态射 f 及其对应的态射挤压(f)的示例。



例 3.1.5.9. 如果 $f\colon (s_1,\ldots,s_m) \to (t_1,\ldots,t_n)$ 是一个元组态射，那么

$$f = \mathsf{squeeze}(f) \quad \Leftrightarrow \quad \text{no } s_i,\, t_j \text{ is equal to } 1.$$

命题 3.1.5.10. 如果 f 是一个元组态射，那么

$$L_{\mathsf{squeeze}(f)} = \mathsf{squeeze}(L_f).$$

证明。 假设 $f\colon (s_1,\ldots,s_m) \to (t_1,\ldots,t_m)$ 是一个元组态射，并令

$$L_f = (s_1,\ldots,s_m) : (d_1,\ldots,d_m)$$

为与 f 相关联的平坦布局。令 $I = \{i_1 < \cdots < i_{m'}\} \subset \langle m\rangle$ 表示具有 $s_{i_k} \neq 1$ 的指标子集。那么

$$\begin{aligned} L_{f|_I} &= (s_{i_1},\ldots,s_{i_k}) : (d_{i_1},\ldots,d_{i_k}) \\ &= \mathsf{squeeze}(L_f). \end{aligned}$$

令 $J = \{j_1 < \cdots < j_{n'}\} \subset \langle n\rangle$ 表示具有 $t_{j_k} \neq 1$ 的指标子集，因此 $\mathsf{squeeze}(f) = (f|_I)|^J$。令 β 表示 $\mathsf{squeeze}(f)$ 所在的映射。那么

$$L_{\mathsf{squeeze}(f)} = L_{(f|_I)|^J} = (s_{i_1},\ldots,s_{i_k}) : (d'_{i_1},\ldots,d'_{i_k})$$

其中

$$\begin{aligned} d'_{i_k} &= \frac{d_{i_k}}{\left(\prod_{\ell<\beta(k) \text{ and } \ell\notin J} t_\ell\right)} \\ &= d_{i_k} \end{aligned}$$

因为对于任意 $\ell \notin J$， $t_\ell = 1$。因此我们得出

$$\begin{aligned} L_{\mathsf{squeeze}(f)} &= (s_{i_1},\ldots,s_{i_k}) : (d_{i_1},\ldots,d_{i_k}) \\ &= \mathsf{squeeze}(L_f). \end{aligned}$$

□

观察 3.1.5.11. 如果 f 是元组态射，那么

$$\mathsf{squeeze}(\mathsf{squeeze}(f)) = \mathsf{squeeze}(f),$$

so

$$\mathbf{Tuple} \xrightarrow{\mathsf{squeeze}(-)} \mathbf{Tuple}$$

是一个幂等函子。

3.1.5.3 排序

排序操作 $f \mapsto \mathrm{sort}(f)$ 置换 f 的定义域，使得到的态射成为已排序的，其含义如下。

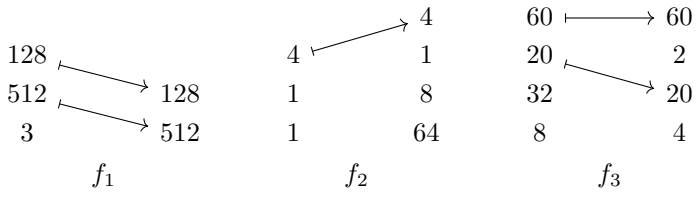
定义 3.1.5.12. 我们称一个元组态射

$$(s_1, \ldots, s_m) \xrightarrow[\alpha]{f} (t_1, \ldots, t_n)$$

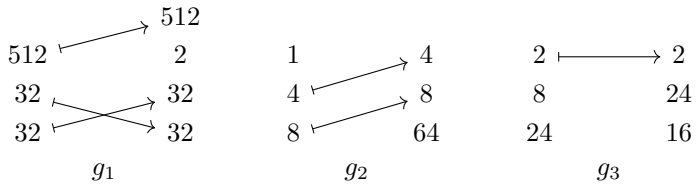
是已排序的，如果对于任意 $1 \leq i, j \leq m$，以下条件成立。

1. 如果 $\alpha(i) = * \neq \alpha(j)$，那么 $i < j$。
2. 如果 $\alpha(i) = * = \alpha(j)$，那么
$$i \leq j \quad \Rightarrow \quad s_i \leq s_j.$$
3. 如果 $\alpha(i) \neq * \neq \alpha(j)$，那么
$$i \leq j \quad \Rightarrow \quad \alpha(i) \leq \alpha(j).$$

例 3.1.5.13. 下方所示的态射 f_1、f_2 和 f_3 已排序，



而下方所示的态射 g_1、g_2 和 g_3



are 未排序。态射 g_1、g_2 和 g_3 分别违反条件 3、1 和 2。

Pr命题 3.1.5.14. 如果 f 是一个已排序的元组态射，那么平坦布局 L_f 已排序 .

证明。设

$$(s_1, \ldots, s_m) \xrightarrow[\alpha]{f} (t_1, \ldots, t_n)$$

已排序，并考虑布局

$$L_f = (s_1, \ldots, s_m) : (d_1, \ldots, d_m).$$

假设 $1 \leq i < m$。我们要证明 $d_i < d_{i+1}$，或者 $d_i = d_{i+1}$ 和 $s_i \leq s_{i+1}$。需要考虑两种情况。

- （情形 1）假设 $\alpha(i) = *$，因此 $d_i = 0$。如果 $\alpha(i+1) = *$，那么 $d_{i+1} = 0$；又因为 f 已排序，所以有 $s_i \leq s_{i+1}$。如果 $\alpha(i+1) \neq *$，那么 $d_{i+1} \geq 1 > 0 = d_i$。

- （情形 2）假设 $\alpha(i) \neq *$，此时 $\alpha(i+1) \neq *$，且 $\alpha(i) < \alpha(i+1)$。那么

$$d_i = \prod_{j<\alpha(i)} t_j \leq \prod_{j<\alpha(i+1)} = d_{i+1},$$

其中等号成立当且仅当 $s_i = t_{\alpha(i)} = 1$，这意味着 $s_i \leq s_{i+1}$。

我们得出 L_f 已排序。 □

接下来，我们在 **Tuple** 上定义排序(–)运算。如果 f 是一个元组态射，那么排序(f)将通过对 f 前复合一个适当的置换 g得到。

构造 3.1.5.15。假设

$$(s_1, \ldots, s_m) \xrightarrow[\alpha]{f} (t_1, \ldots, t_n)$$

是一个元组态射。我们如下定义置换 $\sigma \in \Sigma_m$。令

$$P = \{i \in \langle m \rangle \mid \alpha(i) = *\},$$
$$Q = \{i \in \langle m \rangle \mid \alpha(i) \neq *\},$$

于是 $\langle m \rangle$ 是 P 与 Q的不交并。我们通过以下方式定义 P 的线性序：如果

1. $s_{i_1} < s_{i_2}$，或 2.
$s_{i_1} = s_{i_2}$ 和 $i_1 \leq i_2$。

如果 $\alpha(i_1) \leq \alpha(i_2)$，则我们将 Q 上的线性序定义为 $j_1 \preceq_Q j_2$。如果满足以下条件，我们将 $\langle m \rangle$ 上的线性序定义为 $i_1 \preceq i_2$：

1. $i_1 \in P$ 和 $i_2 \in Q$，2. i_1,
$i_2 \in P$ 和 $i_1 \preceq_P i_2$，或 3. i_1,
$i_2 \in Q$ 和 $i_1 \preceq_Q i_2$。

令 σ 为与 $\langle m \rangle$上的线性序 $\preceq$ 相关联的置换，并令 σ^{-1} 为其逆映射。映射σ_*^{-1}:$\langle m \rangle_* \to \langle m \rangle_*$ 可由一个元组态射覆盖

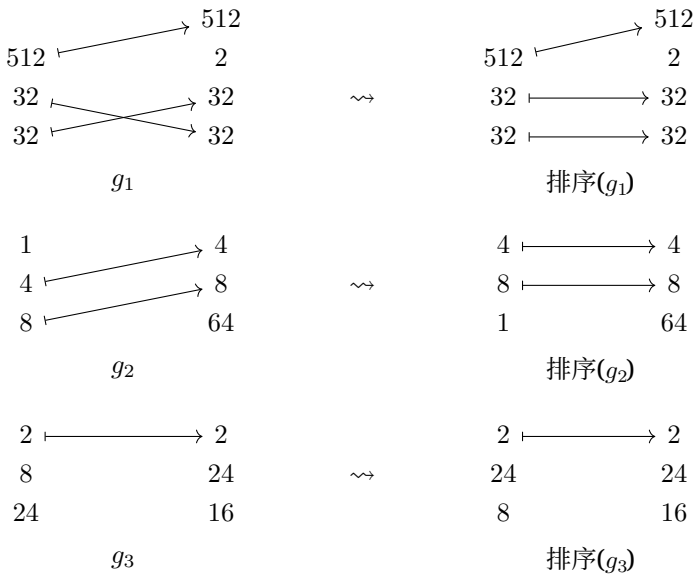
$$g : (s_{\sigma^{-1}(1)}, \ldots, s_{\sigma^{-1}(m)}) \to (s_1, \ldots, s_m),$$

并且我们将 sort(f) 定义为复合

$$\mathsf{sort}(f) = f \circ g.$$

例 3.1.5.16. 态射的排序 g_1， g_2，以及 g_3 例 3.1.5.13 如下所示

。



引理 3.1.5　.17. 假设 $f:\ S \to T$ 是一个元组态射。那么，当且仅当 排序$(f) = f$时，f 是已排序的。.

证明。 我们对 排序$(-)$ 的构造保证，对于任意元组态射 排序(f) f，结果都是已排序的。特别地，如果 $f =$ 排序(f)，那么 f 就是已排序的。反过来，如果 f 是已排序的，那么构造 $\sigma \in \Sigma_m$3.1.5.15 中的置换是恒等置换，因此 $g = \mathsf{id}_S$，于是

$$\mathsf{sort}(f) = f \circ \mathsf{id}_S = f.$$

□

命题 3.1.5.18。 如果 f 是元组态射，那么

$$L_{\mathsf{sort}(f)} = \mathsf{sort}(L_f).$$

证明。 沿用构造中的记号 3.1.5.15，我们有 排序$(f) = f \circ g$ 其中

$$g : (s_{\sigma^{-1}(1)}, \ldots, s_{\sigma^{-1}(m)}) \to (s_1, \ldots, s_m)$$

位于 σ_*^{-1} 之上: $\langle m \rangle_* \to \langle m \rangle_*$。 如果 $L_f = (s_1, \ldots, s_m) : (d_1, \ldots, d_m)$，那么

$$\begin{aligned} L_{\mathsf{sort}(f)} &= (s'_1, \ldots, s'_m) : (d'_1, \ldots, d'_m) \\ &= (s_{\sigma^{-1}(1)}, \ldots, s_{\sigma^{-1}(m)}) : (d_{\sigma^{-1}(1)}, \ldots, d_{\sigma^{-1}(m)}). \end{aligned}$$

由于 $L_{\mathsf{sort}(f)}$ 的模式是 L_f的模式的一个置换，只需证明 $L_{\mathsf{sort}(f)}$ 已排序。设 $1 \le i < m$。先假设 $\sigma^{-1}(i) \in P$，于是 $d'_i = d_{\sigma^{-1}(i)} = 0$。若 $\sigma^{-1}(i+1) \in P$，则 $d'_{i+1} = d_{\sigma^{-1}(i+1)} = 0$。根据 σ的构造，我们有 $s'_i = s_{\sigma^{-1}(i)} \le s_{\sigma^{-1}(i+1)} = s'_{i+1}$。反之，若 $\sigma^{-1}(i+1) \in Q$，则 $d'_{i+1} = d'_{\sigma^{-1}(i+1)} > 0 = d'_i$。接着假设 $\sigma^{-1}(i) \in Q$。那么根据 σ的构造，我们有 $\sigma^{-1}(i+1) \in Q$ 且 $\alpha(\sigma^{-1}(i)) < \alpha(\sigma^{-1}(i+1))$，并且有

$$\begin{aligned} d'_i = d_{\sigma^{-1}(i)} &= \prod_{j < \alpha(\sigma^{-1}(i))} t_j \\ &\le \prod_{j < \alpha(\sigma^{-1}(i+1))} t_j \\ &= d_{\sigma^{-1}(i+1)} \\ &= d'_{i+1}, \end{aligned}$$

其中，当且仅当 $t_{\alpha(\sigma^{-1}(i))} = \ldots = t_{\alpha(\sigma^{-1}(i+1))-1} = 1$时等号成立。特别地，我们有 $s_i = s_{\sigma^{-1}(i)} = t_{\alpha(\sigma^{-1}(i))} = 1$，因此 $s'_i \leq s'_{i+1}$。由此可知 $L_{\mathrm{sort(f)}}$ 已排序，所以 $L_{\mathsf{sort}(f)} = \mathsf{sort}(L_f)$。 □

注记 3.1.5.1 9. 操作 sort(–) 不是函子性的。例如，考虑元组态射 s

$$(2,3) \xrightarrow[(2,1)]{f} (3,2) \qquad \text{and} \qquad (10,25) \xrightarrow[(2,1)]{g} (25,10)$$

那么 f 和 g 可与 $g \circ f = \mathsf{id}_{(25,10)}$复合，但排序后的态射

$$(10,25) \xrightarrow[(1,2)]{\mathsf{sort}(f)} (10,25) \qquad \text{and} \qquad (25,10) \xrightarrow[(1,2)]{\mathsf{sort}(g)} (25,10)$$

不可复合。

3.1.5.4 合并

我们首先引入合并元组态射的概念。

定义 3.1.5.20。设 $f\colon S \to T$ 是位于 α 之上的元组态射。若满足以下条件，我们称 f 是合并的：

1. $S =$挤压（S），并且
2. 对于任意 $1 \leq i < \mathsf{len}(S)$，以下条件中恰有一个成立：

   （a） $\alpha(i) = * \neq \alpha(i+1)$，（b） $\alpha(i) \neq * = \alpha(i+1)$，
   （c） $\alpha(i) > \alpha(i+1)$，或（d） $\alpha(i) < \alpha(i+1)$，并且存在满足 $t_j > 1$ 的 $\alpha(i) < j < \alpha(i+1)$。

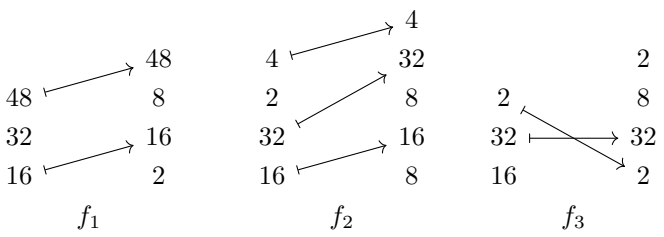
例 3.1.5.21。如果存在某个 $1 \leq i < \mathsf{len}(S)$，使得 $\alpha(i+1) = \alpha(i) + 1$，则 f 不是合并的。

注记 3.1.5.22。如果 $f\colon S \to T$ 是满足 $f =$ 挤压（f）的元组态射，则当且仅当对于任意 $1 \leq i < \mathsf{len}(S)$，以下条件之一成立时，$f$ 是合并的：

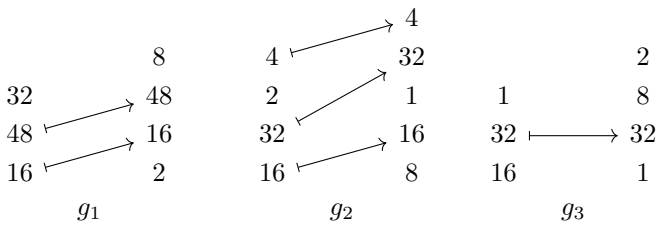
1. $\alpha(i) = * \neq \alpha(i+1)$、
2. $\alpha(i) \neq * = \alpha(i+1)$、
3. $\alpha(i) > \alpha(i+1)$,

或 4. $\alpha(i+1) \neq \alpha(i) + 1$。

例 3.1.5.23. 这些态射



是合并的，而这些态射



没有合并。

命题 3.1.5.24。 设 f 是一个元组态射。当且仅当 L_f 是合并的时， f 是合并的。

证明。 设 $f\colon (s_1, \ldots, s_m) \to (t_1, \ldots, t_n)$ 是一个元组态射，并令

$$L_f = (s_1, \ldots, s_m) : (d_1, \ldots, d_m)$$

为由 f 编码的布局。

首先设 f 是合并的。那么，形状为 shape(L_f) 的 domain(f) 中没有任何条目等于 1。设 $1 \leq i < m$。我们要证明 $s_i d_i$ 不等于 d_{i+1}。如果 $d_i = 0$，那么 $\alpha(i) = *$，于是我们有

$$\begin{aligned} s_i d_i = d_{i+1} \quad &\Leftrightarrow \quad d_{i+1} = 0 \\ &\Leftrightarrow \quad \alpha(i+1) = * \end{aligned}$$

但根据 f 是合并的这一假设，我们有 $\alpha(i+1) \neq *$，因此 $s_i d_i \neq d_{i+1}$。如果 $d_i \neq 0$，那么 $\alpha(i) \neq *$。如果 $\alpha(i+1) = *$，那么 $d_{i+1} = 0$，所以 $s_i d_i \neq d_{i+1}$。如果 $\alpha(i+1) < \alpha(i)$，那么 $d_i \geq d_{i+1}$，并且由于 $s_i \neq 1$，我们有 $s_i d_i > d_{i+1}$。最后，如果 $\alpha(i) < \alpha(i+1)$，那么

$$\begin{aligned} s_i d_i = s_i \cdot \left(\prod_{j < \alpha(i)} t_j \right) &= \prod_{j \leq \alpha(i)} t_j \\ &< \prod_{j < \alpha(i+1)} t_j \\ &= d_{i+1}. \end{aligned}$$

我们得出结论： L_f 是合并的。

接下来假设布局 L_f是合并的。那么，定义域(f) =形状(L_f)中没有任何条目等于 1。假设 $1 \leq i < m$。如果 $\alpha(i) = *$，那么 $d_i = 0$；又因为 L_f 是合并的，我们必须有 $d_{i+1} \neq s_i d_i = 0$，因此 $\alpha(i+1) \neq *$。假设 $\alpha(i) \neq *$，且 $\alpha(i) < \alpha(i+1)$。由于 L_f是合并的，我们有 $s_i d_i \neq d_{i+1}$。但是如果我们写成

$$s_i d_i = \prod_{j \leq \alpha(i)} t_j,$$

and

$$d_{i+1} = \prod_{j < \alpha(i+1)} t_j,$$

这意味着 $\prod_{\alpha(i) < j < \alpha(i+1)} t_j \neq 1$。特别地，存在某个 $\alpha(i) < j < \alpha(i+1)$，使得 $t_j > 1$。我们得出结论： f 是合并的。

□

接下来，我们定义作用于元组态射上的 coal(–) 运算。

构造 3.1.5.25. 假设 f 是一个元组态射。我们定义态射 coal(f) 如下:

1. 首先，我们设置 $g =$ 挤压(f)，并将 β 写作: $\langle m\rangle_* \to \langle n\rangle_*$，表示 g 所处的映射。

2. 接下来，我们在 $\langle m\rangle$ 上定义等价关系 $\sim$，其中 $i \sim i'$ 当且仅当以下任一条件成立:

   (a) 对于 $i \le i'' \le i'$，$\beta(i'') = *$；或者 (b)
   对于 $i \le i'' \le i'$，$\beta(i'') = \beta(i) + (i'' - i)$。

   商集 $\langle m\rangle/\sim$ 按 $[i_1] \le [i_2]$ 排序，其中 $i_1 \le i_2$，因此我们可以将该商集与 $\langle \bar{m}\rangle$对应起来，其中 $\bar{m}$ 是 $\langle m\rangle/\sim$的大小。

3. 接下来，在 $\langle n\rangle$ 上定义等价关系 $\sim$，其中 $j \sim j'$ 当且仅当存在 $i \in \langle m\rangle$ 使得

$$\beta(i + (j'' - j)) = \beta(i) + (j'' - j)$$

   对所有 $j \le j'' \le j'$。当 $j_1 \le j_2$ 时，商集 $\langle n\rangle/\sim$ 按 $[j_1] \le [j_2]$ 排序，因此我们可以将此商集与 $\langle \bar{n}\rangle$ 视为相同，其中 $\bar{n}$ 是 $\langle n\rangle/\sim$的大小。

4. 接下来，我们观察到映射 $\beta : \langle m\rangle_* \to \langle n\rangle_*$ 可下降为一个映射

$$\bar{\beta} : \langle \bar{m}\rangle_* \to \langle \bar{n}\rangle_*$$

   由 $\beta([i]) = [\beta(i)]$ 给出。

5. coal(f) 的定义域 $S = (\bar{s}_1, \dots, \bar{s}_{\bar{m}})$ 定义为

$$\bar{s}_i = \prod_{i' \in I} s_{i'}$$

   若 $i \in \langle \bar{m}\rangle$ 对应于等价类 $I \in \langle m\rangle/\sim$。coal($f$) 的陪域 $\bar{T} = (\bar{t}_1, \dots, \bar{t}_{\bar{n}})$ 定义为

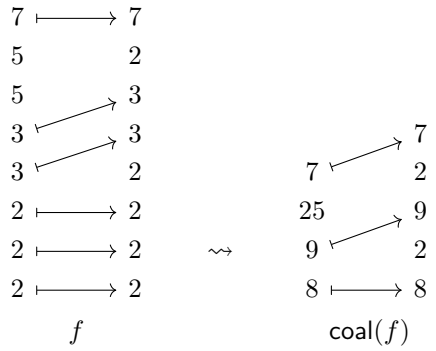
$$\bar{t}_j = \prod_{j' \in J} t_{j'}$$

   如果 $j \in \langle \bar{n}\rangle$ 对应于等价类 $J \in \langle n\rangle/\sim$。于是我们定义

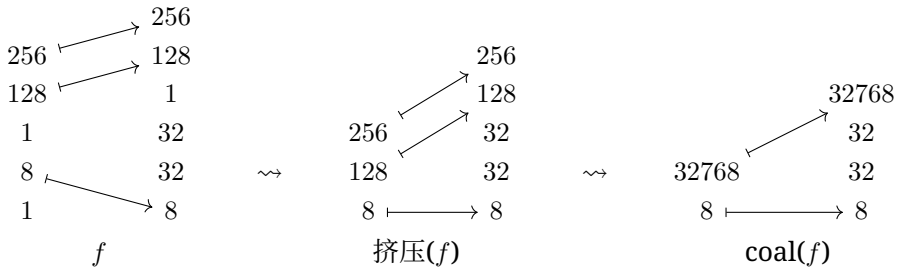
$$\mathsf{coal}(f) : \bar{S} \to \bar{T}$$

   为位于 β 之上的元组态射。

例 3.1.5.26. 下面是元组态射 f 及其合并态射 coal(f)的一个例子。



例 3.1.5.27. 我们可以将态射 f （例 3.1.5.8 ）合并如下



命题 3.1.5.28. 如果 f 是一个元组态射，那么

1. $\mathsf{coal}(f)$ 是合并的，并且
2. $L_{\mathsf{coal}(f)} = \mathsf{coal}(L_f)$.

证明。 首先，我们将说明 $\mathsf{coal}(f)$ 是合并的。这直接来自我们的构造，因为应用挤压会消除所有等于 1 的模式，而在构造中取商集则会合并所有满足 $\alpha(i+1) = \alpha(i) + 1$ 的相邻模式。

接下来，我们将证明 $L_{\mathsf{coal}(f)} = \mathsf{coal}(L_f)$。根据命题 2.1.4.18 和命题3.1.5.24，只需证明 $\Phi_{\mathsf{coal}(f)} = \Phi_f$。显然，对 f 应用挤压(–) 不会影响相关布局函数，因此我们需要说明，在构造中取商集不会改变相关布局的布局函数。这是因为我们的商集可以分步构造，每一步都会合并满足以下条件之一的相邻模式：$\alpha(i) = * = \alpha(i+1)$，或 $\alpha(i+1) = \alpha(i) + 1$。这分别对应于将形如$s_i, s_{i+1}\colon 0,0$ 的相邻模式替换为 $s_i s_{i+1}\colon 0$，以及将 $s_i, s_{i+1}\colon d_i, s_i d_i$ 替换为 $s_i s_{i+1}\colon d_i$。这两种操作都不会改变布局的布局函数，因此我们得出 $\Phi_{L_{\mathsf{coal}(f)}} = \Phi_{\mathsf{coal}(L_f)}$，正如所需。

□

3.1.5.5 拼接

接下来，我们将定义元组态射上的拼接运算。该运算可用于满足下文所述“不相交”条件的元组态射。

定义 3.1.5.29. 假设 $\alpha : \langle m\rangle_* \to \langle n\rangle_*$ 和 $\beta : \langle p\rangle_* \to \langle n\rangle_*$ 是 $\mathbf{Fin}_*$ 中具有相同陪域的态射。若 α 和 β 满足图像不相交，则称它们具有不相交像。

$$\mathsf{Image}(\alpha) \cap \mathsf{Image}(\beta) = \{*\}.$$

构造 3.1.5.30. 若 $\alpha : \langle m\rangle_* \to \langle n\rangle_*$ 和 $\beta : \langle p\rangle_* \to \langle n\rangle_*$ 的像不相交，则可良定义一个态射

$$\alpha \star \beta : \langle m+p\rangle_* \to \langle n\rangle_*$$

由下式给出

$$(\alpha \star \beta)(i) = \begin{cases} * & i = * \\ \alpha(i) & 1 \le i \le m \\ \beta(i-m) & m+1 \le i \le m+p. \end{cases}$$

此运算满足结合律，因此对于$\cdots \star \alpha_k$，我们可以考虑$\cdots \star \alpha_k$任意一组在$\mathbf{Fin}_*$ 中的态射 $\alpha_1, \ldots, \alpha_k$，其像两两不相交。

注记 3.1.5.31。若 α 和 β 是可处理的带基点映射，且 α 和 β 的像不相交，则 $\alpha \star \beta$ 是可处理的。

定义 3.1.5.32。设

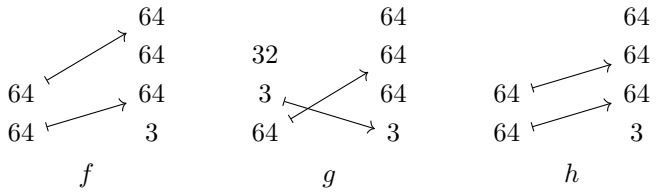
$$f : S \to T$$

and

$$g : U \to T$$

分别是覆盖 α 和 β 的元组态射。我们称 f 和 g 具有不相交像，如果态射 α 和 β 具有不相交像。

例 3.1.5.33。考虑下方所示的元组态射 f、g 和 h。



那么 f 和 g 具有不相交像，而 h 和 g 不具有不相交像。

构造 3.1.5.34。设

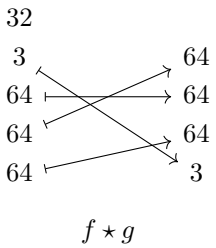
$$f : S \to T, \text{ and } g : U \to T$$

分别是覆盖 α 和 β 的元组态射，并且 f 和 g 具有不相交像。我们将 f 和 g 的串接定义为态射

$$f \star g : S \star U \to T$$

覆盖 $\alpha \star \beta$。此运算满足结合律，因此对于任意有限个像两两不相交的态射 f_i，我们可以考虑 $f_1 \cdots f_k$。

例 3.1.5.35. 如果 f 和 g 是 **Tuple** 中例3.1.5.33里的态射，那么 f 和 g 的拼接态射如下所示。



$f \star g$

例 3.1.5.36. 假设 $f\colon (s_1, \ldots, s_m) \to (t_1, \ldots, t_n)$ 是一个元组态射，并且对于任意 $1 \leq i \leq m$，令

$$f_i : (s_i) \to (t_1, \ldots, t_n)$$

表示 i 是 f 的第几个条目，如例 3.1.3.8所示。于是我们可以写成

$$f = f_1 \star \cdots \star f_m$$

作为其条目的串接。

引理 3.1.5.37. 设 $f_1\colon\ S_1 \to T$ 和 $f_2\colon\ S_2 \to T$ 是像不相交的元组态射。若 $g\colon\ T \to U$ 是任意元组态射，则

$$g \circ (f_1 \star f_2) = (g \circ f_1) \star (g \circ f_2).$$

证明。假设 f_1、f_2 和 g 分别位于 α_1 上：$\langle m_1\rangle_* \to \langle n\rangle$、$\alpha_2 : \langle m_2\rangle_* \to \langle n\rangle$，以及 $\beta : \langle n\rangle \to \langle p\rangle$。所讨论的两个映射具有相同的定义域和相同的陪域，因此只需证明

$$\beta \circ (\alpha_1 \star \alpha_2) = (\beta \circ \alpha_1) \star (\beta \circ \alpha_2).$$

我们计算

$$\begin{aligned}
(\beta \circ (\alpha_1 \star \alpha_2))(i) &= \beta((\alpha_1 \star \alpha_2)(i)) \\
&= \begin{cases} \beta(*) & i = * \\ \beta(\alpha_1(i)) & 1 \le i \le m_1 \\ \beta(\alpha_2(i - m_1)) & m_1 + 1 \le i \le m_1 + m_2 \end{cases} \\
&= \begin{cases} * & i = * \\ (\beta \circ \alpha_1)(i) & 1 \le i \le m_1 \\ (\beta \circ \alpha_2)(i - m_1) & m_1 + 1 \le i \le m_1 + m_2 \end{cases} \\
&= ((\beta \circ \alpha_1) \star (\beta \circ \alpha_2))(i).
\end{aligned}$$

□

命题 3.1.5.38。 假设 f_1、.....、f_k 是 **Tuple** 中的态射，它们具有相同的陪域且像两两不相交。那么布局 L_{f_1}、.....、L_{f_k} 满足

$$L_{f_1 \star \cdots \star f_k} = L_{f_1} \star \cdots \star L_{f_k}.$$

证明。 首先，我们证明该结论对 $k = 2$ 成立。假设

$$f = (s_1, \ldots, s_m) \to (t_1, \ldots, t_n), \text{ and } g : (u_1, \ldots, u_p) \to (t_1, \ldots, t_n)$$

像不相交，并记

$$L_f = (s_1, \ldots, s_m) : (d_1, \ldots, d_m), \text{ and } L_g = (u_1, \ldots, u_p) : (d'_1, \ldots, d'_p).$$

那么布局 $L_{f \star g}$ 定义为

$$L_{f \star g} = (s_1, \ldots, s_m, u_1, \ldots, u_p) : (e_1, \ldots, e_{m+m'})$$

其中

$$\begin{aligned}
e_i &= \prod_{j < (\alpha \star \beta)(i)} t_j \\
&= \begin{cases} \displaystyle\prod_{j < \alpha(i)} t_j & 1 \le i \le m \\ \displaystyle\prod_{j < \beta(i-m)} t_j & m + 1 \le i \le m + m'. \end{cases} \\
&= \begin{cases} d_i & 1 \le i \le m \\ d'_{i-m} & m + 1 \le i \le m + m'. \end{cases}
\end{aligned}$$

至此， $k = 2$ 时该结果的证明结束。一般情形可由元组态射串接的结合律以及平坦布局串接的结合律推出。 □

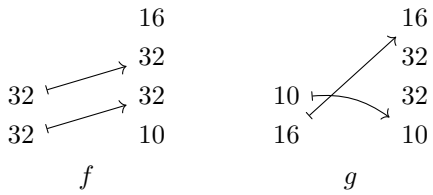
3.1.5.6 补

我们首先定义互补元组态射的概念。

定义 3.1.5.39. 假设 $f\colon S \to T$ 和 $g\colon U \to T$ 是元组态射。若 g 是 f 的补，则称

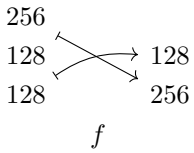
1. f 和 g 的像不相交，且
2. 串接
$$f \star g : S \star U \xrightarrow{\cong} T$$
是一个同构。

例 3.1.5.40. 若 f 和 g 是下图所示的态射



那么 g 是 f 的一个补。

例 3.1.5.41. 如果 f 是下图所示的态射



那么 f 不存在补。

接下来，我们证明互补元组态射会导出互补平坦布局。

命题 3.1.5.42. 如果 $f\colon S \to T$ 是一个元组态射，且 g 是 f 的一个补，那么 L_g 是 L_f 的大小(T)-补。

证明. 记 $S =$ 为 domain(f)，$U =$ 为 domain(g)，且 $T =$ 为 codomain(f) $=$ codomain(g)。首先，我们注意到

$$\begin{aligned}\mathsf{size}(L_f) \cdot \mathsf{size}(L_g) &= \mathsf{size}(L_f \star L_g)\\ &= \mathsf{size}(L_{f\star g})\\ &= \mathsf{size}(S \star U)\\ &= \mathsf{size}(T).\end{aligned}$$

接下来，我们注意到 $f \star g$ 是一个同构，因此

$$|f \star g| = \Phi_{L_{f\star g}}^{\mathsf{size}(T)}$$

其中我们使用了 $\Phi_{L_{f\star g}}^{\mathsf{size}(T)}$ 引理3.1.4.5中的识别。 □

命题 3.1.5.43。 如果 f 是单射元组态射，那么

$$\mathsf{coal}^\flat(L_{f^c}) = \mathsf{comp}^\flat(L_f, \mathsf{size}(T)).$$

证明。 由命题 3.1.5.42，我们知道 L_{f^c}是 大小(T)-补，补于 L_f。由于 f^c已排序，因此 L_{f^c} 也已排序，并且由命题 2.1.6.33可得:

$$\mathsf{coal}^\flat(L_{f^c}) = \mathsf{comp}^\flat(L_f, \mathsf{size}(T)),$$

sin由于这两个布局都是 L_f 的平坦、已排序、合并补对象，且大小相同 . □

命题 3.1.5.44。 若 $f\colon (s_1, \ldots, s_m) \to (t_1, \ldots, t_n)$ 是一个标准形式的单射元组态射，则

$$L_{f^c} = \mathsf{comp}^\flat(L_f).$$

证明。 记

$$L_f = (s_1, \ldots, s_m) : (d_1, \ldots, d_m)$$

表示由 f 编码的布局。由命题 3.1.5.42，我们知道 L_{f^c}是一个 大小(T)-补，补于L_f。其中

$$\begin{aligned}\mathsf{size}(T) = t_1 \cdots t_n &= (t_1 \cdots t_{n-1})t_n \\ &= d_m s_m.\end{aligned}$$

根据构造， f^c已排序，因此 L_{f^c}也已排序。此外，由于 f具有标准形式，可知 f^c是合并的。由命题 2.1.6.23，我们可推得

$$L_{f^c} = \mathsf{comp}^\flat(L_f).$$

□

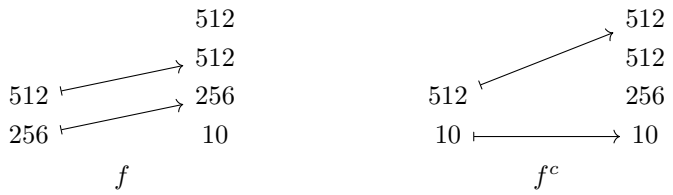
定义 3.1.5.45. 设 f 是一个位于 α 之上的元组态射：$\langle m\rangle_* \to \langle n\rangle_*$。若 α 是单射，则称 f 是可补的 。

构造 3.1.5.46. 设 $f\colon (s_1, \ldots, s_m) \to (t_1, \ldots, t_n)$ 是一个可补的元组态射。令 $j_1 < \cdots < j_{n-m}$ 表示 $\langle n\rangle$ 中不属于 α 的像的指标集合。我们将 f 的补 定义为以下元组态射

$$f^c : (t_{j_1}, \ldots, t_{j_k}) \to (t_1, \ldots, t_n)$$

位于映射 补$(\alpha) : \langle n-m\rangle_* \to \langle n\rangle_*$ 给定为 $k \mapsto j_k$。根据构造，我们可以观察到 f^c是 f 的一个补，其含义见定义 3.1.5.39

例 3.1.5.47. 下面是一个态射 f 及其补 f^c的例子。



命题 3.1.5.48. 若 f 是元组态射，且 g 是 f的补，则

$$\mathsf{sort}(g) = f^c.$$

证明。 假设 f 位于 α 之上：$\langle m\rangle_* \to \langle n\rangle_*$，sort($g$) 位于 β 之上：$\langle n-m\rangle_* \to \langle n\rangle_*$，且 f^c 位于α^c之上：$\langle n-m\rangle_* \to \langle n\rangle_*$。于是 β 和 α^c 是具有相同像的递增映射，即

$$\mathsf{Image}(\beta) = \langle n\rangle \setminus \mathsf{Image}(\alpha) = \mathsf{Image}(\alpha^c),$$

因此 $\beta = \alpha^c$，进而 sort(g) $= f^c$。 □

命题 3.1.5.49. 设 f 是一个元组态射。那么 f 存在补，当且仅当 f是可补的，符合定义*3.1.5.45*。

证明。 若 f在一个非单射映射 α之上，则对于任意满足以下条件的态射 f^*： f和 f^*具有不相交像，态射 $f \star f^*$ 在一个非单射映射之上，因此 $f \star f^*$ 不是同构。反之，若 f 在一个单射映射之上，则构造 3.1.5.46 中的态射 f是其补。

□

命题 3.1.5.50. 若 f 是可补的元组态射，则

$$\mathsf{sort}(f) = (f^c)^c.$$

证明。 两个映射都是递增的、单射的，且具有相同的像，因此它们相等。 □

3.1.5.7 平坦除法

本节定义元组态射上的除法运算。

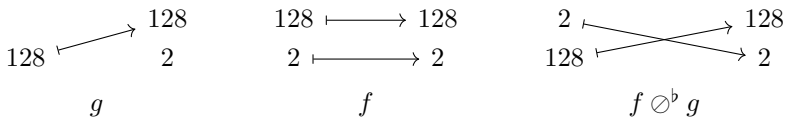
定义 3.1.5.51。 如果 f 和 g 是元组态射，则当 g 和 f 可复合时，我们称 g 整除 f。换句话说，

$$\mathsf{codomain}(g) = \mathsf{domain}(f).$$

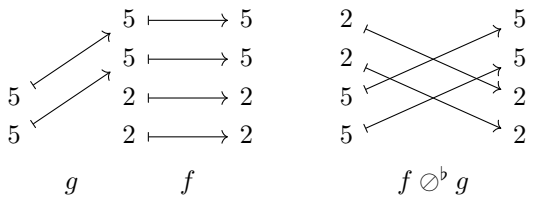
定义 3.1.5.52。 设 $g : S \to T$ 且 $f : T \to U$ 是元组态射。 f 除以 g 的平坦除法 是元组态射

$$f \oslash^\flat g = f \circ (g \star g^c).$$

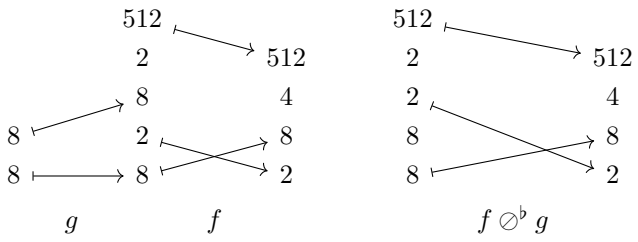
例 3.1.5.53。 下面给出元组态射 f 和 g 以及它们的平坦商 $f \oslash^\flat g$ 的一个例子。



例 3.1.5.54. 这是一个关于元组态射 f 和 g 及其平坦商集$f \oslash^\flat g$的例子。



例 3.1.5.55. 这是一个关于元组态射 f 和 g 及其平坦商集$f \oslash^\flat g$的例子。



命题 3.1.5.56. 若 f 和 g 是非退化的可复合元组态射，则

$$\mathsf{coal}^\flat(L_{f \oslash^\flat g}) = \mathsf{coal}^\flat(L_f \oslash^\flat L_g)$$

证明。 由命题 3.2.6.20，可得

$$\mathsf{coal}^\flat(L_{g^c}) = \mathsf{comp}^\flat(L_g, \mathsf{size}(L_f)),$$

并且我们计算得

$$\begin{aligned}
\mathsf{coal}^\flat(L_f \oslash^\flat L_g) &= \mathsf{coal}^\flat(L_f \circ (L_g \star \mathsf{comp}(L_g, \mathsf{size}(L_f)))) \\
&= \mathsf{coal}(L_f \circ (L_g \star L_{g^c})) \\
&= \mathsf{coal}(L_f \circ L_{g \star g^c}) \\
&= \mathsf{coal}(L_{f \circ (g \star g^c)}) \\
&= \mathsf{coal}(L_{f \oslash^\flat g}).
\end{aligned}$$

□

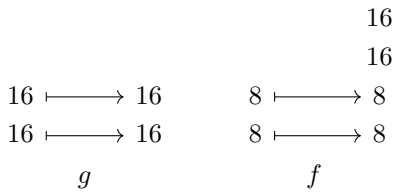
3.1.5.8 扁平积

本节中，我们定义元组态射上的乘积运算。

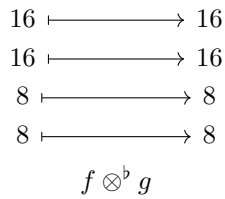
定义 3.1.5.57. 设 f 和 g 是元组态射。若陪域(g) $=$ 定义域(f^c)，则称 f 和 g 为乘积可容许。若 f 和 g 乘积可容许，则将 f 和 g的平坦积定义为

$$f \otimes^\flat g = f \star (f^c \circ g).$$

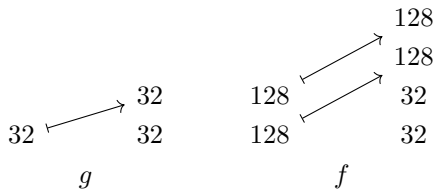
例 3.1.5.58. 如果 f 和 g 是下图所示的元组态射



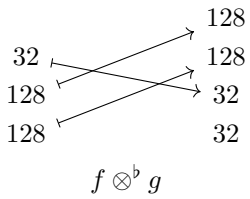
那么 f 和 g 是乘积可容许的，而 $f \otimes^\flat g$ 是下图所示的元组态射。



例 3.1.5.59. 如果 f 和 g 是下图所示的元组态射



那么， f 和 g 是乘积可容许的，而 $f \otimes^\flat g$ 是下方所示的元组态射。



引理 3.1.5.60。 如果 f 和 g 是乘积可容许的，且 g 是单射，那么 $f \otimes^\flat g$ 是单射，并且

$$(f \otimes^\flat g)^c = f^c \circ g^c.$$

证明。 元组态射 $(f \otimes^\flat g)^c$和 $f^c \circ g^c$都是单射、递增的，并且具有相同的陪域，因此只需证明它们具有相同的像。$(f \otimes^\flat g)^c = (f \star (f^c \circ g))^c$ 的像由不在 f的像中且不在 $f^c \circ g$的像中的那些条目组成。 f^c的像由不在 f的像中的那些条目组成，因此复合 $f^c \circ g^c$的像由不在 f的像中且不在 $f^c \circ g$的像中的那些条目组成。

□

命题 3.1.5.61。 假设 f 和 g 是乘积可容许的，且 g 和 h 是乘积可容许的。那么

1. $f \otimes^\flat g$ 和 h 是乘积可容许的，
2. f 和 $g \otimes^\flat h$ 是乘积可容许的，且 *3.* $(f \otimes^\flat g) \otimes^\flat h = f \otimes^\flat (g \otimes^\flat h)$ 。

证明。 利用引理 3.1.5.37和引理3.1.5.60，我们计算

$$\begin{aligned} f \otimes^\flat (g \otimes^\flat h) &= f \star (f^c \circ (g \otimes^\flat h)) \\ &= f \star (f^c \circ (g \star (g^c \circ h))) \\ &= f \star ((f^c \circ g) \star (f^c \circ (g^c \circ h))) \\ &= f \star ((f^c \circ g) \star ((f^c \circ g^c) \circ h)) \\ &= f \star (f^c \circ g) \star ((f \otimes^\flat g)^c \circ h) \\ &= (f \otimes^\flat g) \star ((f \otimes^\flat g)^c \circ h) \\ &= (f \otimes^\flat g) \otimes^\flat h. \end{aligned}$$

□

命题 3.1.5.62。 设 f 和 g 是非退化元组态射，且 f 和 g 是乘积可容许的。则

$$L_{f \otimes^\flat g} = L_f \otimes^\flat L_g.$$

证明。设 $f\colon\ S \to T$ 和 $g\colon\ U \to V$ 是乘积可容许的，并令

$$L_f^* = \mathsf{comp}^\flat(L_f, \mathsf{size}(L_f) \cdot \mathsf{cosize}(L_g)).$$

由于 f 是单射，且 g 的陪域是 f^c的定义域，因此可得

$$\mathsf{size}(L_f) \cdot \mathsf{cosize}(L_g) \leq \mathsf{size}(S) \cdot \mathsf{size}(V) = \mathsf{size}(T).$$

利用这一事实，以及

$$\Phi_{\mathsf{comp}(L_f, \mathsf{size}(T))} = \Phi_{L_{f^\mathsf{c}}},$$

我们有

$$\begin{aligned} L_f^* \circ L_g &= \mathsf{comp}(L_f, \mathsf{size}(T)) \circ L_g \\ &= L_{f^\mathsf{c}} \circ L_g. \end{aligned}$$

利用这一事实，我们计算

$$\begin{aligned} L_f \otimes^\flat L_g &= L_f \star (L_f^* \circ L_g) \\ &= L_f \star (L_{f^\mathsf{c}} \circ L_g) \\ &= L_f \star L_{f^\mathsf{c} \circ g} \\ &= L_{f \star (f^\mathsf{c} \circ g)} \\ &= L_{f \otimes^\flat g} \end{aligned}$$

□

3.2 范畴 Nest

在上一节中，我们引入了一个范畴**Tuple**，其态射编码了平坦可处理布局。本节中，我们将引入一个范畴 **Nest**，其态射编码了具有任意嵌套结构的可处理布局。

3.2.1 基本定义

回想一下，对于嵌套元组 S，我们用 $S^\flat$ 表示 S的扁平化。例如，若 $S = (64, (8, 8))$，则 $S^\flat = (64, 8, 8)$。

定义 3.2.1.1。令 **Nest**表示这样一个范畴：其对象是正整数的嵌套元组，且其中的态射

$$f : S \to T$$

在 **Nest**中由一个元组态射指定

$$f^\flat : S^\flat \to T^\flat.$$

换句话说，

$$\mathsf{Hom}_{\mathbf{Nest}}(S, T) = \mathsf{Hom}_{\mathbf{Tuple}}(S^\flat, T^\flat).$$

具体而言，**Nest** 中的态射 $f : S \to T$ 由一个可处理的带基点映射 $\alpha : \langle \mathsf{len}(S) \rangle_* \to \langle \mathsf{len}(T) \rangle_*$ 指定，并满足以下性质：

- If $1 \le i \le \mathsf{len}(S)$ and $\alpha(i) \neq *$, then $\mathsf{entry}_i(S) = \mathsf{entry}_{\alpha(i)}(T)$.

我们称这样的态射 f 覆盖 α，并将 f 称为嵌套元组态射。

记号　**注 3.2.1.2**。若 $f : S \to T$ 是一个覆盖 α的嵌套元组态射，我们将 f 表示为

$$S \xrightarrow[\alpha]{f} T$$

例 3.2.1.3。以下是一些嵌套元组态射的示例。

$$(64, (8, 8)) \xrightarrow[(1,2,3)]{f} (64, 8, 8)$$

$$((2, 2), 2) \xrightarrow[(*,5,2)]{g} (10, 2, 2, (3, 2, 3))$$

$$64 \xrightarrow[(2)]{h} ((64, 64), 512).$$

观察 3.2.1.4。若 X 是一个集合，记 X^{ind} 为 X上的粗糙范畴。该范畴的对象是 X的元素，且任意两个对象之间都有唯一的（同构）态射。根据**Nest**的定义，我们有一个拉回方块

$$\begin{array}{ccc} \mathbf{Nest} & \xrightarrow{\mathsf{prof}(-)} & \mathsf{Profile}^{\mathsf{ind}} \\ {\scriptstyle (-)^\flat}\downarrow & \lrcorner & \downarrow{\scriptstyle \mathsf{len}(-)} \\ \mathbf{Tuple} & \xrightarrow[\mathsf{len}(-)]{} & \mathbb{N}^{\mathsf{ind}} \end{array}$$

我们可以将其视为对拉回方块的范畴化 2.2.2.4。

例 3.2.1.5。 假设 S 是一个长度为 m的嵌套元组。如果 $1 \leq i \leq m$，那么存在一个嵌套元组态射

$$\mathsf{entry}_i(S) \to S$$

位于映射 $\langle 1 \rangle_* \to \langle m \rangle_*$ 之上，该映射由 $1 \mapsto i$ 给出。例如，若 $S = (64,(8,8))$ 且 $i = 1$，则有一个嵌套元组态射

$$64 \xrightarrow[(1)]{} (64,(8,8)).$$

例 3.2.1.6. 假设 S 是秩为 r 的嵌套元组。若 $1 \leq i \leq r$，则存在一个规范的嵌套元组态射

$$\mathsf{mode}_i(S) \to S$$

位于映射 $\langle \mathsf{len}_i(S) \rangle_* \to \langle \mathsf{len}(S) \rangle_*$ 之上，该映射由 $j \mapsto j + \mathsf{len}_{<i}(S)$ 给出。例如，若 $S = (64,(8,8))$，则有一个嵌套元组态射

$$(8,8) \xrightarrow[(2,3)]{} (64,(8,8)).$$

观察 3.2.1.7. 存在函子关联范畴 **Nest** 和 **Tuple**。首先，存在一个包含函子

$$\mathbf{Tuple} \xrightarrow{\subset} \mathbf{Nest}$$

它将元组态射 $f : S \to T$ 视为嵌套元组态射。接下来，存在一个扁平化函子

$$\mathbf{Nest} \xrightarrow{(-)^\flat} \mathbf{Tuple}$$

它将嵌套元组态射 $f\colon S \to T$映射为底层的元组态射 $f^\flat\colon S^\flat \to T^\flat$。复合

$$\mathbf{Tuple} \xrightarrow{\subset} \mathbf{Nest} \xrightarrow{(-)^\flat} \mathbf{Tuple}$$

是**Tuple**上的恒等函子，因此**Tuple** 是**Nest**的可回缩子范畴。此外，这些函子构成范畴的伴随等价。

注记 3.2.1.8。 人们或许希望考虑某个范畴**C** ，其态射编码可处理布局，但它与**Tuple**不 等价。作者考察过几个这样的例子，但将其研究留待未来工作。

3.2.2 从嵌套元组态射到布局

范畴**Nest** 的关键特点是，若 $f : S \to T$ 是一个嵌套元组态射，那么 f 就编码了一个布局 L_f。该布局通过为平坦布局 $L_{f^\flat}$ 赋予 S的嵌套轮廓而得到。更准确地说，我们有如下构造。

构造 3.2.2.1。**设**

$$f : S \to T$$

是一个嵌套元组态射，并假设 $P = \mathsf{prof}(S)$。我们将 L_f 定义为布局

$$L_f = (L_{f^\flat})_P$$

其中 $(-)_P$ 是 P-替换操作，见定义 2.3.1.19。我们将 L_f 称为由 编码的布局 f。

构造 3.2.2.2。假设

$$(s_1, \ldots, s_m)_P \xrightarrow[\alpha]{f} (t_1, \ldots, t_n)_Q$$

是一个嵌套元组态射。我们将 L_f 定义为形状为

$$\mathsf{shape}(L_f) = (s_1, \ldots, s_m)_P$$

的布局，其中 f的定义域为，步幅

$$\mathsf{stride}(L_f) = (d_1, \ldots, d_m)_P$$

的条目由以下公式定义

$$d_i = \begin{cases} 0 & \alpha(i) = * \\ \prod_{j<\alpha(i)} t_j & \alpha(i) \neq *. \end{cases}$$

我们将 L_f 称为由 f编码的布局。

例 3.2.2.3. 由以下内容编码的布局

$$((8,8),(4,4)) \xrightarrow[(1,4,3,2)]{f} (8,4,4,8)$$

is

$$L_f = ((8,8),(4,4)) : ((1,128),(32,8)).$$

例 3.2.2.4. 由以下内容编码的布局

$$(128,(4,4,2)) \xrightarrow[(3,1,2,*)]{g} ((4,4),128)$$

is

$$L_g = (128,(4,4,2)) : (16,(1,4,0)).$$

观察 3.2.2.5. 扁平化函子

$$\textbf{Nest} \xrightarrow{(-)^\flat} \textbf{Tuple}$$

与布局的扁平化相容，也就是说，如果 f 是一个嵌套元组态射，那么

$$(L_f)^\flat = L_{f^\flat}.$$

如果 L 是一个可处理布局，那么我们可以构造一个嵌套元组态射，如下所示，它将 L 编码为

构造 3.2.2.6. 设 L 是一个可处理布局。我们将 L的标准表示定义为嵌套元组态射

$$f_L : S \to T$$

其中 $(f_L)^\flat = f_{L^\flat}$ 是 $L^\flat$ 的标准表示， S =shape(L) 是 L 的形状，而 T 是 $f_{L^\flat}$ 的陪域。

例 3.2.2.7. 若

$$L = (32, (2, 2)) : (192, (24, 3))$$

那么 L的标准表示为

$$(32, (2, 2)) \xrightarrow[(6,4,2)]{f_L} (3, 2, 4, 2, 4, 32).$$

引理 3.2.2.8. 若 L 是可处理布局，且 $f = f_L$ 是 L的标准表示，则

$$L_f = L.$$

证明. 我们有

$$(L_f)^\flat = L_{f^\flat} = L^\flat$$

and

$$\mathsf{shape}(L_f) = \mathsf{shape}(L).$$

□

命题 3.2.2.9. 设 L 是一个布局。则存在一个编码 L的嵌套元组态射 f，当且仅当 L 是可处理的。

证明. 首先假设 $L = L_f$对于某个嵌套元组态射 f。那么 $(L_f)^\flat = L_{f^\flat}$，并且根据命题 3.1.2.10，我们知道 $L^\flat$ 是可处理的，因此 L也是可处理的。反过来，若 L 是可处理的，则可取 $f = f_L$ 为 L的标准表示，并根据引理 3.2.2.8，我们有 $L_f = L$。 □

为了建立可处理布局与某些嵌套元组态射之间的一一对应关系，我们引入嵌套元组态射的标准形式 这一概念。

定义 3.2 .2.10. 设 $f : S \to T$ 是一个嵌套元组态射。我们称 f 具有标准形式 i f

1. $f^\flat$ 具有标准形式，如定义 3.1.2.12中所述，并且
2. T 是平坦的。

例 3.2.2.11. 嵌套元组态射

$$((2, 2), (3, 3)) \xrightarrow[(4,6,2,3}]{f} (10, 3, 3, 2, 10, 2)$$

具有标准形式。

例 3.2.2.12. 嵌套元组态射

$$((2, 2), (3, 3)) \xrightarrow[(4,6,2,3}]{f} ((10, 3, 3), (2, 10, 2))$$

不具有标准形式，因为 g 的陪域不是平坦的。

与平坦情形一样，为了在标准形式的嵌套元组态射与可处理布局之间建立一一对应，我们需要排除非退化嵌套元组态射和非退化布局。为此，我们作如下定义。

定义 3.2.2.13. 设

$$S \xrightarrow[\alpha]{f} T$$

是一个嵌套元组态射，并且假设

$$L = S : D$$

是一个布局。

1. 我们称 f 是非退化的，如果

$$\mathsf{entry}_i(S) = 1 \quad \Rightarrow \quad \alpha(i) = *.$$

2. 我们称 L 是非退化，如果

$$\mathsf{entry}_i(S) = 1 \quad \Rightarrow \quad \mathsf{entry}_i(D) = 0.$$

注记 3.2.2.14. 如果 f 是一个嵌套元组态射，那么 f 是非退化的，当且仅当 $f^\flat$ 是非退化的。如果 L 是一个布局，那么 L 是非退化的，当且仅当 $L^\flat$ 是非退化的。

命题 3.2.2.15. 这些映射

$$f \longmapsto L_f$$

$$\left\{ \begin{array}{c} \textit{Non-degenerate} \\ \textit{nested tuple morphisms} \\ \textit{of standard form} \end{array} \right\} \longleftrightarrow \left\{ \begin{array}{c} \textit{Non-degenerate} \\ \textit{tractable layouts} \end{array} \right\}$$

$$f_L \longmapsfrom L$$

构造*3.2.2.2* 和*3.2.2.6* 确定了标准形式的嵌套元组态射与可处理布局之间的一一对应。

证明。 我们已在命题 3.2.2.9 中证明，若 L 是可处理布局，且 $f = f_L$ 是 L的标准形式，则 $L_f = L$。接下来，假设 f 具有标准形式，并令 $L = L_f$ 为由 f编码的布局。我们要证明 f 等于 f_L 的标准表示 L。根据命题 3.1.2.21，我们知道 $f^\flat$ 等于 $f_{L^\flat}$ 的标准表示 $L^\flat$，并且由于

$$\mathsf{domain}(f) = \mathsf{shape}(L) = \mathsf{domain}(f_L),$$

and

$$\mathsf{codomain}(f) = \mathsf{codomain}(f^\flat) = \mathsf{codomain}(f_{L^\flat}) = \mathsf{codomain}(f_L),$$

由此可得 $f = f_L$。 □

3.2.3 示例

本节列出一些重要的嵌套元组态射族。

例 3.2.3.1（重新加括号）。 假设 S_1 和 S_2 是扁平化结果相同的嵌套元组

$$S_1^\flat = S_2^\flat.$$

然后存在一个重新加括号同构

$$\mathsf{id}_{S_1}^{S_2} : S_1 \xrightarrow{\cong} S_2$$

覆盖恒等态射。这些态射具有传递性，即

$$\mathsf{id}_{S_2}^{S_3} \circ \mathsf{id}_{S_1}^{S_2} = \mathsf{id}_{S_1}^{S_3},$$

并且与恒等态射相容，即

$$\mathsf{id}_S^S = \mathsf{id}_S.$$

如果 $f = \mathsf{id}_{S_1}^{S_2}$ 是一个重新加括号同构，那么 L_f 是形状为 S_1 的列主序布局。

例 3.2.3.2（展平）。 作为前一个例子的特殊情形，如果 S 是任意嵌套元组，那么我们有一个展平同构

$$\mathsf{id}_S^{S^\flat} : S \xrightarrow{\cong} S^\flat$$

以及一个反扁平化同构

$$\mathsf{id}_{S^\flat}^S : S^\flat \xrightarrow{\cong} S$$

观察 3.2.3.3. 如果 $f : S \to T$ 是一个嵌套元组，那么 f 等于以下复合

$$S \xrightarrow{\mathsf{id}_S^{S^\flat}} S^\flat \xrightarrow{f^\flat} T^\flat \xrightarrow{\mathsf{id}_{T^\flat}^T} T.$$

换句话说，我们有一个规范分解

$$f = \mathsf{id}_{T^\flat}^T \circ f^\flat \circ \mathsf{id}_S^{S^\flat}.$$

例 3.2.3.4(条目)。假设

$$S \xrightarrow[\alpha]{f} T$$

是一个嵌套元组态射。假设 $1 \leq i \leq \mathsf{len}(S)$，并记 $j = \alpha(i)$。那么我们将嵌套元组态射

$$\mathsf{entry}_i(S) \xrightarrow[(j)]{\mathsf{entry}_i(f)} T$$

称为 f 的第 i 个条目。由条目 $_i(f)$ 编码的布局是

$$L_{\mathsf{entry}_i(f)} = \mathsf{entry}_i(L_f).$$

例 3.2.3.5 (条目包含)。 作为前一个例子的特例，如果 S 是一个嵌套元组且 $1 \leq i \leq \mathsf{len}(S)$，我们可以取 $f = \mathsf{id}_S$，此时

$$\mathsf{entry}_i(\mathsf{id}_S) : \mathsf{entry}_i(S) \longrightarrow S$$

是 S 的第 i 个条目的包含映射。

例 3.2.3.6(模式)。假设

$$S \xrightarrow[\alpha]{f} T$$

是一个嵌套元组态射。假设 $1 \leq i \leq \mathrm{rank}(S)$，并且，写作

$$N = \mathsf{len}_{<i}(S)$$
$$\ell = \mathsf{len}_i(S).$$

那么我们将嵌套元组态射称为

$$\mathsf{mode}_i(S) \xrightarrow[(N+1,\ldots,N+\ell)]{\mathsf{mode}_i(f)} T$$

S 的第 i 个模式。由 $\mathsf{mode}_i(L_f)$ 编码的布局为

$$L_{\mathsf{mode}_i(f)} = \mathsf{mode}_i(L_f).$$

例 3.2.3.7 （模态包含）。 作为上一个例子的一个特例，我们可以取 $f = \mathrm{id}_S$，此时

$$\mathsf{mode}_i(\mathsf{id}_S) : \mathsf{mode}_i(S) \to S$$

是 S 的第 i 个模式的包含映射。我们有时将此映射记为

$$\mathsf{incl}_i(S) = \mathsf{mode}_i(\mathsf{id}_S).$$

3.2.4 嵌套元组态射的实现

在平坦情形中，我们构造了一个实现函子

$$\mathbf{Tuple} \xrightarrow{|\cdot|} \mathbf{FinSet}$$

它将元组态射 f 映射为 L_f的布局函数。我们可以通过与扁平化函子前复合，将其扩展为一个实现函子

$$\mathbf{Nest} \xrightarrow{|\cdot|} \mathbf{FinSet}$$

通过与扁平化函子 $\mathbf{Nest} \to \mathbf{Tuple}$ 前复合。

定义 3.2.4.1。我们定义实现函子

$$\mathbf{Nest} \xrightarrow{|\cdot|} \mathbf{FinSet}$$

为复合

$$\mathbf{Nest} \xrightarrow{(-)^\flat} \mathbf{Tuple} \xrightarrow{|\cdot|} \mathbf{FinSet}$$

引理 3.2.4.2。 若 $f:\ S \to T$ 是一个嵌套元组态射，则 f 的实现 $|f|$ 是 L_f的布局函数：

$$|f| = \Phi_{L_f}^{\mathsf{size}(T)}.$$

证明。 由 3.1.4.5 可立即得出，因为

$$|f| = |f^\flat| = \Phi_{L_{f^\flat}}^{\mathsf{size}(T)} = \Phi_{L_f}^{\mathsf{size}(T)}$$

□

3.2.5 细化

本节将从范畴论视角重新审视嵌套元组的细化。回顾第2.2.4 节，嵌套元组 S' 细化 S，记作

$$S' \longrightarrow\!\!\!\!\!\rightarrow S$$

如果 S' 可以通过将 S 的每个条目替换为大小相同的某个嵌套元组而得到。例如，

$$(2,(2,2)) \twoheadrightarrow 8,$$

and

$$((2,2),(3,3),(5,5)) \twoheadrightarrow (4,9,25).$$

如果 $\mathsf{len}(S) = m$ 且 $\mathsf{prof}(S) = P$，那么我们可以写作

$$S' = (S'_1, \ldots, S'_m)_P$$

作为相对模式的 P-替换

$$S'_i = \mathsf{mode}_i(S', S).$$

我们将普通连接

$$(S'_1, \ldots, S'_m) = \mathsf{flat}(S', S)$$

称为相对于 S 对 S' 进行扁平化。

令 **Ref** 表示在细化关系下由嵌套正整数组成的偏序集范畴，因此 **Ref** 中的态射是一个细化 $S' \twoheadrightarrow S$。若 S 是一个嵌套元组，令

$$\mathbf{Ref}(S) = \{S' \mid S' \text{ refines } S\}$$

表示 细化 S 的嵌套元组构成的偏序集。等价地，**Ref**(S) 是切片范畴 **Ref**/ $_S$.

构造 3.2.5.1. [相对模态嵌入] 假设 $S' \twoheadrightarrow S$ 是一个细化，并记

$$S'_i = \mathsf{mode}_i(S', S)$$

为 S'相对于 S 的模式。那么 S' 和 $(S'_1, \ldots, S'_m)$ 具有相同的扁平化，因此我们有一个重新加括号同构

$$\mathsf{id}^{S'}_{(S'_1,\ldots,S'_m)} : (S'_1, \ldots, S'_m) \xrightarrow{\cong} S'$$

并定义

$$\mathsf{incl}_i(S', S) : S'_i \to S'$$

为复合

$$S'_i \xrightarrow{\mathsf{incl}_i((S'_1,\ldots,S'_m))} (S'_1, \ldots, S'_m) \xrightarrow{\mathsf{id}^{S'}_{(S'_1,\ldots,S'_m)}} S'$$

第 i个模的 包含映射（S'_1，……，S'_m）与重新加括号同构（S'_1，……，S'_m）$\cong S'$.

例 3.2.5.2。 如果 $S = (4, (9, 25))$ 且 $S' = ((2, 2), ((3, 3), 25))$，那么 S' 细化 S，且 $\mathsf{incl}_2(S', S)$ 是嵌套元组态射

$$(3,3) \xrightarrow[(3,4)]{\mathsf{incl}_2(S',S)} ((2,2),((3,3),25)).$$

构造 3.2.5.3。 [相对模式] 设 $f'\colon S' \to T'$ 是一个嵌套元组态射，并设 S' 细化 S。我们将 i 态射的第 th 个模式 f' 相对于 S 定义为，记作

$$\mathsf{mode}_i(f', S) = f' \circ \mathsf{incl}_i(S', S) : S'_i \to T'$$

为复合

$$S'_i \xrightarrow{\mathsf{incl}_i(S',S)} S' \xrightarrow{f'} T'$$

特别地，我们有

$$\mathsf{mode}_i(\mathsf{id}_{S'}, S) = \mathsf{incl}_i(S', S).$$

例 3.2.5.4。 设 $S = (4, (9, 25))$ 和 $S' = ((2, 2), ((3, 3), 25))$，从而 S' 细化 S。若 f' 是嵌套元组态射

$$((2,2),((3,3),25)) \xrightarrow[(1,3,2,*,4)]{f'} (2,3,2,25).$$

则 $\mathsf{mode}_2(f', S)$ 是嵌套元组态射

$$(3,3) \xrightarrow[(2,*)]{\mathsf{mode}_2(f',S)} (2,3,2,25).$$

构造 3.2.5.5 （拉回）。 设 $f\colon S \to T$ 是位于 α 之上的嵌套元组态射，并设 $T' \twoheadrightarrow T$ 是一个细化。令

$$T'_j = \mathsf{mode}_j(T', T)$$

表示相对于 T 的 T' 的第 j 个模式，并对任意 $1 \le i \le \mathsf{len}(S)$，设

$$S'_i = \begin{cases} \mathsf{entry}_i(S) & \alpha(i) = * \\ T'_j & \alpha(i) = j. \end{cases}$$

我们将沿 f 对 T' 的拉回定义为嵌套元组

$$S' = f^*T' = \mathsf{sub}(S, (S'_1, \ldots, S'_m)).$$

对于任意 $1 \le i \le m$，令

$$f'_i : S'_i \to T'$$

若 $\alpha(i) = *$，则该映射为平凡映射；若非如此，则为包含映射

$$\mathsf{incl}_j(T', T) : S'_i = T'_j \to T'$$

若 $\alpha(i) = j$。映射 $f'_1, \ldots, f'_m$ 的像互不相交，因此我们构造串接

$$(f'_1, \ldots, f'_m) : (S'_1, \ldots, S'_m) \to T'.$$

我们将 $f' = T'^* f$ 定义为复合映射

$$S' \xrightarrow{\mathsf{id}_{S'}^{(S'_1,\ldots,S'_m)}} (S'_1, \ldots, S'_m) \xrightarrow{(f'_1,\ldots,f'_m)} T'.$$

我们将 f' 称为沿着 T 对 f 的拉回，并将这样的拉回表示为一个方形图

$$\begin{array}{ccc} S' & \xrightarrow{f'} & T' \\ \downarrow & \lrcorner & \downarrow \\ S & \xrightarrow[f]{} & T. \end{array}$$

例 3.2.5.6。 设 $f\colon (64,\ 32) \to (4,\ 64,\ 4,\ 32)$ 位于 $\alpha = (2,\ 4)$ 之上。那么我们得到一个拉回方块

$$\begin{array}{ccc} ((16,4),(16,2)) & \xrightarrow{f'} & ((2,2),(16,4),(2,2),(16,2)) \\ \downarrow & \lrcorner & \downarrow \\ (64,32) & \xrightarrow[f]{} & (4,64,4,32) \end{array}$$

其中 f' 位于 $\alpha' = (3,\ 4,\ 7,\ 8)$ 之上。

例 3.2.5.7。 设 S 是一个具有扁平化的嵌套元组

$$S^\flat = (s_1, \ldots, s_m),$$

并设 $S' \twoheadrightarrow S$ 是一个具有相对扁平化的细化

$$(S'_1, \ldots, S'_m).$$

那么，沿着反扁平化同构对 $S' \twoheadrightarrow S$ 作拉回

$$\mathsf{id}^S_{(s_1,\ldots,s_m)} : (s_1, \ldots, s_m) \to S$$

就是重新加括号同构

$$\begin{array}{ccc} (S'_1, \ldots, S'_m) & \xrightarrow{\mathsf{id}^{S'}_{(S'_1,\ldots,S'_m)}} & S' \\ \downarrow & \lrcorner & \downarrow \\ (s_1, \ldots, s_m) & \xrightarrow[\mathsf{id}^S_{(s_1,\ldots,s_m)}]{} & S. \end{array}$$

Ex例 3.2.5.8。 设 $S' \twoheadrightarrow S$ 是一个细化，并考虑第 i 个条目的包含映射

$$s_i \to S.$$

那么， $S' \twoheadrightarrow S$ 沿着 $s_i \to S$ 的拉回就是第 i个相对模式包含映射

$$\begin{array}{ccc} S'_i & \xrightarrow{\mathsf{incl}_i(S',S)} & S' \\ \downarrow & \lrcorner & \downarrow \\ s_i & \longrightarrow & S. \end{array}$$

O观察 3.2.5.9。 上述拉回构造定义了一个反变函子

$$\mathbf{Nest}^{\mathrm{op}} \longrightarrow \mathbf{Cat}$$

$$\begin{array}{ccc} S & \longmapsto & \mathbf{Ref}(S) \\ {\scriptstyle f}\downarrow & & \uparrow{\scriptstyle f^*} \\ T & \longmapsto & \mathbf{Ref}(T) \end{array} \qquad \begin{array}{ccc} f^*T' & \leftarrow & f^*T'' \\ & & \uparrow \\ T' & \leftarrow & T'' \end{array}$$

拉回的关键性质是， f'的布局函数等于 f的布局函数。

引理 3.2.5.10。假设

$$\begin{array}{ccc} S' & \xrightarrow{f'} & T' \\ \downarrow & \lrcorner & \downarrow \\ S & \xrightarrow[f]{} & T \end{array}$$

是一个拉回方块，其中 f 位于 α之上。令

$$f'_i : S'_i \to T$$

表示相对于 S的 f' 的第 i个模式，并令

$$(L_f)^\flat = (s_1, \ldots, s_m) : (d_1, \ldots, d_m).$$

那么对于任意 $1 \leq i \leq m$，我们有

$$\mathsf{coal}(L_{f'_i}) = s_i : d_i.$$

证明。 假设 $1 \leq i \leq m$。如果 $\alpha(i) = *$，那么 f'_i 是平凡映射，因此

$$L_{f'_i} = s_i : 0 = s_i : d_i.$$

特别地，$\mathsf{coal}(L_{f'_i}) = s_i : 0 = s_i : d_i$。接下来假设 $\alpha(i) = j \neq *$。根据 f'的构造，我们有

$$f'_i = \mathsf{incl}_j(T', T) : T'_j \to T'.$$

它位于映射 α'_i之上，该映射由 $t \mapsto \mathsf{len}_{<j}(T') + t$给出。对于每个 $1 \leq t < \mathsf{len}(T_j)$，我们有$\alpha'_i(t) = \alpha'_i(t+1)$，因此 $L_{f'_i}$是一个大小为 $\mathsf{size}(T'_j) = t_j = s_i$的列主序布局。这意味着 $\mathsf{coal}(L_{f'_i})$ 是一个形如以下形式的深度 0 布局

$$\mathsf{coal}(L_{f'_i}) = s_i : e$$

for 某个整数 $e \geq 0$。我们断言 $e = d_i$。如果我们将 $t'_{j'} =$ 条目$_{j'}(T')$ 写作，那么我们有

$$\begin{aligned} e = \mathsf{entry}_1(\mathsf{stride}(L_{f'_i})) &= \prod_{j' < \alpha'_i(1)} t'_{j'} \\ &= \prod_{j' \leq \mathsf{len}_{<j}(T', T)} t'_{j'} \\ &= \prod_{j' < j} \mathsf{size}(T'_{j'}) \\ &= \prod_{j' < j} t_{j'} \\ &= d_i. \end{aligned}$$

□

命题 3.2.5.11。如果

$$\begin{array}{ccc} S' & \xrightarrow{f'} & T' \\ \downarrow & \lrcorner & \downarrow \\ S & \xrightarrow[f]{} & T \end{array}$$

是一个拉回方块，那么 $\Phi_{L_f} = \Phi_{L_{f'}}$。

证明。 我们首先固定记号。设 $m = \mathsf{len}(S)$，并设

$$\begin{aligned} S^\flat &= (s_1, \ldots, s_m), \\ S'_i &= \mathsf{mode}_i(S', S), \\ T'_j &= \mathsf{mode}_j(T', T), \\ (L_f)^\flat &= (s_1, \ldots, s_m) : (d_1, \ldots, d_m). \end{aligned}$$

考虑重新加括号同构

$$\mathsf{id}^{S'}_{(S'_1,\ldots,S'_m)} : (S'_1, \ldots, S'_m) \to S'$$

此映射与 f' 的复合是串接 (f'_1、……、f'_m)，其中若 $\alpha(i) = *$，则 f'_i 是平凡映射，而相对模式包含为

$$\mathsf{incl}_i(T', T) : S'_i = T'_j \to T'$$

否则。利用引理 3.2.5.10，以及 $L_{f'} = L_{(f'_1,\ldots,f'_m)}$ 这一事实，我们计算得

$$\begin{aligned} \mathsf{coal}(L_{f'}) &= \mathsf{coal}(L_{(f'_1,\ldots,f'_m)}) \\ &= \mathsf{coal}((L_{f'_1}, \ldots, L_{f'_m})) \\ &= \mathsf{coal}((\mathsf{coal}(L_{f'_1}), \ldots, \mathsf{coal}(L_{f'_m}))) \\ &= \mathsf{coal}((s_1, \ldots, s_m) : (d_1, \ldots, d_m)) \\ &= \mathsf{coal}(L_f). \end{aligned}$$

由命题 2.3.3.14，我们推得 $\Phi_{L_{f'}} = \Phi_{L_f}$。 □

构造 3.2.5.12 （推前）。 设 $f : S \to T$ 是一个位于 α 之上的嵌套元组态射，并设 $S' \twoheadrightarrow S$ 是一个细化。令

$$S'_i = \mathsf{mode}_i(S', S)$$

表示相对于 S 的 S' 的第 i 个模式，并且对任意满足 $1 \le j \le \mathsf{len}(T)$ 的情况，定义

$$T'_j = \begin{cases} \mathsf{entry}_j(T) & j \notin \mathsf{Image}(\alpha) \\ S'_i & \alpha(i) = j. \end{cases}$$

我们将的推前沿着 f 定义为嵌套元组

$$T' = f_* S' = \mathsf{sub}(T, (T'_1, \ldots, T'_n)).$$

对于任意 $1 \le i \le m$，令

$$f'_i : S'_i \to T'$$

若 $\alpha(i) = *$，则为平凡映射；否则为相对模式包含映射

$$\mathsf{incl}_j(T', T) : S'_i = T'_j \to T'$$

若 $\alpha(i) = j$。态射 f'_1，……，f'_m 的像互不相交，因此我们可以进行串接

$$(f'_1, \ldots, f'_m) : (S'_1, \ldots, S'_m) \to T'.$$

我们将 $f' = S'_* f$ 定义为复合

$$S' \xrightarrow{\mathsf{id}_{S'}^{(S'_1,\ldots,S'_m)}} (S'_1,\ldots,S'_m) \xrightarrow{(f'_1,\ldots,f'_m)} T'.$$

我们将 f' 称为 f 沿着 T 的推前。我们将这样的推前表示为

$$\begin{array}{ccc} S' & \xrightarrow{f'} & T' \\ \downarrow & \lrcorner & \downarrow \\ S & \xrightarrow[f]{} & T \end{array}$$

例 3.2.5.13. 如果 $f\colon (64,\ 32) \to (4,\ 64,\ 4,\ 32)$ 位于 $\alpha = (2,\ 4)$ 之上，那么我们有一个推前方块

$$\begin{array}{ccc} ((16,4),(16,2)) & \xrightarrow{f'} & (4,(16,4),4,(16,2)) \\ \downarrow & \lrcorner & \downarrow \\ (64,32) & \xrightarrow[f]{} & (4,64,4,32) \end{array}$$

拉回的关键性质是，f' 的布局函数与 f 的布局函数相同。

引理 3.2.5.14. 假设

$$\begin{array}{ccc} S' & \xrightarrow{f'} & T' \\ \downarrow & \lrcorner & \downarrow \\ S & \xrightarrow[f]{} & T \end{array}$$

是一个推前方块，其中 f 位于 α 之上。令

$$f'_i : S'_i \to T$$

表示相对于 S 而言的 f' 的第 i 个模式，并令

$$(L_f)^\flat = (s_1,\ldots,s_m) : (d_1,\ldots,d_m).$$

那么对于任意 $1 \le i \le m$，我们有

$$\mathsf{coal}(L_{f'_i}) = s_i : d_i.$$

证明。 证明与引理 3.2.5.10 的证明完全相同 □

命题 3.2.5.15. 如果

$$\begin{array}{ccc} S' & \xrightarrow{f'} & T' \\ \downarrow & \lrcorner & \downarrow \\ S & \xrightarrow[f]{} & T \end{array}$$

是一个推前方块，那么 $\Phi_{L_f} = \Phi_{L_{f'}}$。

证明。 证明与命题 3.2.5.11 中的证明相同。 □

观察　**命题 3.2.5.16.** 上述定义的推前构造指定了一个协变函　or

$$\mathbf{Nest} \longrightarrow \mathbf{Cat}$$

$$\begin{array}{ccc} S & \longmapsto & \textbf{引用}(S) \\ {\scriptstyle f}\downarrow & & \downarrow{\scriptstyle f_*} \\ T & \longmapsto & \mathbf{Ref}(T) \end{array} \qquad \begin{array}{c} S'' \to S' \\ \downarrow \\ f_*S'' \to f_*S' \end{array}$$

观察 3.2.5.17. 若 $f : S \to T$ 是嵌套元组的一个同构，则

$$\mathbf{Ref}(T) \xrightarrow{f^*} \mathbf{Ref}(S)$$

and

$$\mathbf{Ref}(S) \xrightarrow{f_*} \mathbf{Ref}(T)$$

是范畴的逆同构。具体而言，

$$(f^{-1})^* = f_* \quad \text{and} \quad (f^{-1})_* = f^*.$$

观察 3.2.5.18. 若 S_1 和 S_2 是嵌套元组，且 $\mathrm{flat}(S_1) = \mathrm{flat}(S_2)$，则存在一个规范嵌套元组同构 $S_1 \cong S_2$，因此也存在一个典范范畴同构

$$\mathbf{Ref}(S_1) \cong \mathbf{Ref}(S_2).$$

我们还需要明确一个概念，称为 相互细化。这一概念的重要性将在第 4 章中变得清晰，届时我们会在布局组合算法中使用这一概念。

定义 3.2.5.19。 设 T 和 U 是嵌套元组。$(T,\ U)$ 的一个相互细化是如下形式的图表

$$\begin{array}{ccc} T' & \rightarrowtail & U' \\ \downarrow & & \downarrow \\ T & & U \end{array}$$

明确地说，这是一对嵌套元组 $(T\prime U')$，满足

1. T' 细化 T；2. $U\prime$细化 U；3. $T\prime$ 整除 U'。

除了相互细化的定义外，我们还需要以下事实。

引理 3.2.5.20。 设 T 和 U 是嵌套元组。那么，(T, U)（T, U）的相互细化与 （$T^\flat, U^\flat$）的相互细化之间存在一一对应关系。

证明。 如果 $(T\prime U')$ 是 (T, U) 的相互细化，那么沿着反扁平化同构 $\mathrm{id}^T_{T^\flat}$ 和 $\mathrm{id}^U_{U^\flat}$ 拉回后，可得到一个相互细化

$$\begin{array}{ccc} (\mathrm{id}^T_{T^\flat})^* T\prime & \rightarrowtail & (\mathrm{id}^U_{U^\flat})^* U\prime \\ \downarrow & & \downarrow \\ T^\flat & & U^\flat \end{array}$$

$(T^\flat, U^\flat)$。反过来，如果 $((T^\flat)', (U^\flat)')$ 是 $T^\flat, U^\flat$ 的相互细化，那么沿着展平同构 $\mathsf{id}_T^{T^\flat}$ 和 $\mathsf{id}_U^{U^\flat}$ 拉回后，可得到一个相互细化

$$\begin{array}{ccc} (\mathsf{id}_T^{T^\flat})^*(T^\flat)' & \rightarrowtail & (\mathsf{id}_U^{U^\flat})^*(U^\flat)' \\ \downarrow & & \downarrow \\ T & & U \end{array}$$

$(T^\flat,\ U^\flat)$ 的。 □

3.2.6 嵌套元组态射的运算

接下来，我们要构建一种“嵌套元组态射代数”。由于我们已经为元组态射构建了这样的“代数”，因此只需为各种运算的输出配备适当的轮廓，就可以将其扩展到嵌套情形。

3.2.6.1 拼接

接下来，我们在嵌套元组态射上定义一种拼接运算，该运算与布局的串接相容，即

$$L_{(f,g)} = (L_f, L_g).$$

我们通过串接 f 和 g 的定义域来拼接嵌套元组态射 f 和 g。为了使其定义良好，我们需要 f 和 g 满足不相交条件，具体如下所述。

定义 3.2.6.1。设 f 和 g 是具有相同陪域的嵌套元组态射。如果 f和 g 具有不相交像，则称它们 $f^\flat$ 和 $g^\flat$ 具有不相交像，如定义 3.1.5.32所述。

例 3.2.6.2。如果

$$f : (3, (512, 512)) \to (2, 512, 2, 512)$$

在 $(*, 2, 4)$ 之上，且

$$g : (2, 2) \to (2, 512, 2, 512)$$

在 $(1, 3)$ 之上，那么 f 和 g 的像不相交。

例 3.2.6.3。如果

$$f : (2, (32, 64)) \to (32, (2, 2, 2), 64)$$

在 $\alpha = (3, 1, 5)$ 之上，且

$$g : ((2, 2)) \to (32, (2, 2, 2), 64)$$

在 $\beta = (2, 4)$ 之上，那么 f 和 g 的像不相交。

构造 3.2.6.4。假设 $f : S \to T$ 和 $g : U \to T$ 是分别位于 α 和 β 之上的嵌套元组态射，并且 f 和 g 的像不相交。我们将 f 和 g 的串接定义为嵌套元组态射

$$(f, g) : (S, U) \to T$$

with

$$\mathsf{flat}((f, g)) = f^\flat \star g^\flat.$$

更一般地，若 $f_i : S_i \to T$是针对 $1 \leq i \leq k$的嵌套元组态射，且 f_1、…、 f_k的像两两不相交，则我们将串接

$$(f_1, \ldots, f_k) : (S_1, \ldots, S_k) \to T.$$

定义为具有以下形式的嵌套元组态射

$$(f_1, \ldots, f_k)^\flat = f_1^\flat \star \cdots \star f_k^\flat.$$

例 3.2.6.5。例 f 与 g 中态射的串接 3.2.6.2是嵌套元组态射

$$(f, g) : ((3, (512, 512)), (2, 2)) \to (2, 512, 2, 512)$$

位于 $\alpha \star \beta = (*, 2, 4, 1, 3)$ 上。

例 3.2.6.6. 态射 f 与 g （例 3.2.6.3）的串接是嵌套元组态射

$$(f,g):((2,(32,64)),((2,2)))\to(32,(2,2,2),64)$$

位于 $\alpha\star\beta=(3,\ 1,\ 5,\ 2,\ 4)$ 上。

例 3.2.6.7. 如果

$$f:(2,2)\to(2,3,5,2,3,5)$$

位于 $\alpha=(1,\ 4)$ 上

$$g:(3,3)\to(2,3,5,2,3,5)$$

位于 $\beta=(2,\ 5)$ 上，并且

$$h:(5,5)\to(2,3,5,2,3,5)$$

位于 $\gamma=(3,\ 6)$ 上，那么 f、g 和 h 的像两两不相交，并且串接

$$(f,g,h):((2,2),(3,3),(5,5))\to(2,3,5,2,3,5)$$

位于 $\alpha\star\beta\star\gamma=(1,\ 4,\ 2,\ 5,\ 3,\ 6)$ 上。

例 3.2.6.8. 设 $f:S\to T$ 是一个嵌套元组态射，并且设

$$S^\flat=(s_1,\dots,s_m).$$

回顾例 3.2.3.4，对于任意 $1\leq i\leq m$，都存在一个嵌套元组态射

$$f_i:s_i\to T.$$

称为 f 的第 i 个条目。这些态射的像两两不相交，而其串接

$$(f_1,\dots,f_m):S^\flat\to T$$

是例

$$(f_1,\dots,f_m)=f\circ\mathsf{id}^S_{S^\flat}$$

中的复合 3.2.3.2

例 3.2.6.9. 设 $f:S\to T$ 是一个嵌套元组态射，并且设

$$S=(S_1,\dots,S_r).$$

回顾例 3.2.3.6 可知，对于任意 $1\leq i\leq r$，都存在一个嵌套元组态射

$$f_i:S_i\to T.$$

称为 f 的第 i 个模式。这些态射的像两两不相交，且其串接

$$(f_1,\dots,f_r):S\to T$$

等于 f。换句话说，每个嵌套元组态射 f 都可以写成其模式的串接：

$$f=(f_1,\dots,f_r).$$

命题 3.2.6.10。 如果 $f_1,\dots,f_k$ 是具有相同陪域且像两两不相交的嵌套元组态射，那么

$$L_{(f_1,\dots,f_k)}=(L_{f_1},\dots,L_{f_k}).$$

证明。 根据构造，我们有

$$\begin{aligned}\mathsf{shape}((L_{f_1},\dots,L_{f_k})) &= (\mathsf{shape}(L_{f_1}),\dots,\mathsf{shape}(L_{f_k}))\\ &= \mathsf{shape}(L_{(f_1,\dots,f_k)}).\end{aligned}$$

并根据命题 3.1.5.38，我们有

$$\begin{aligned}(L_{f_1},\dots,L_{f_k})^\flat &= L_{f_1}^\flat \star\cdots\star L_{f_k}^\flat\\ &= L_{f_1^\flat}\star\cdots\star L_{f_k^\flat}\\ &= L_{f_1^\flat\star\cdots\star f_k^\flat}\\ &= L_{(f_1,\dots,f_k)^\flat}\\ &= (L_{(f_1,\dots,f_k)})^\flat.\end{aligned}$$

□

3.2.6.2 合并

如果 f 是嵌套元组态射，那么我们可以将 $\mathsf{coal}(f)$ 定义为 $\mathsf{coal}^\flat(f^\flat)$。从理论上说，这是一个合理的定义。但是，为了使我们的定义与 `cute` 实现兼容，我们对 $\mathsf{coal}(f)$ 的定义稍作修改。

定义 3.2.6.11。 假设 $f\colon S\to T$ 是一个嵌套元组态射，并记作

$$\mathsf{coal}^\flat(f^\flat):(s_1,\dots,s_m)\to(t_1,\dots,t_n).$$

- （情形 1）：如果 $m>1$，我们定义
$$\mathsf{coal}(f)=\mathsf{coal}^\flat(f^\flat).$$
- （情形 2）：如果 $m=1$，我们将 $\mathsf{coal}(f)$ 定义为复合
$$s_1\xrightarrow[(1)]{}(s_1)\xrightarrow{\mathsf{coal}^\flat(f^\flat)}(t_1,\dots,t_n).$$
- (情形 3)：如果 $m=0$，我们将 $\mathsf{coal}(f)$ 定义为该复合
$$1\xrightarrow[(*)]{}()\xrightarrow{\mathsf{coal}^\flat(f^\flat)}(t_1,\dots,t_n).$$

例 3.2.6.12。如果

$$f:((2,2),(3,3),(5,5))\to(5,5,3,3,2,2)$$

位于 $\alpha=(5,\ 6,\ 3,\ 4,\ 1,\ 2)$ 之上，则

$$\mathsf{coal}(f):(4,9,25)\to(25,9,4)$$

位于 $\alpha'=(3,\ 2,\ 1)$ 之上。

命题 3.2.6.13。 如果 $f\colon\ S\to T$ 是一个嵌套元组态射，那么

$$\mathsf{coal}(L_f)=L_{\mathsf{coal}(f)}.$$

证明。 我们再次记作

$$(s_1,\dots,s_m) \xrightarrow[\alpha]{\mathsf{coal}^\flat(f^\flat)} (t_1,\dots,t_n).$$

需要考虑三种情形。

- （情形 1）：假设 $m>1$。则

$$\begin{aligned} L_{\mathsf{coal}(f)} &= L_{\mathsf{coal}^\flat(f^\flat)} \\ &= \mathsf{coal}^\flat(L_{f^\flat}) \\ &= \mathsf{coal}((L_f)^\flat) \\ &= \mathsf{coal}(L_f). \end{aligned}$$

- （情形 2）：假设 $m=1$。则

$$\begin{aligned} L_{\mathsf{coal}(f)} &= s_1 : t_1 \dots, t_{\alpha(1)-1} \\ &= \mathsf{coal}((s_1) : (t_1 \cdots t_{\alpha(1)-1})) \\ &= \mathsf{coal}(L_{\mathsf{coal}^\flat(f^\flat)}) \\ &= \mathsf{coal}(\mathsf{coal}^\flat(L_{f^\flat})) \\ &= \mathsf{coal}((L_f)^\flat) \\ &= \mathsf{coal}(L_f). \end{aligned}$$

- （情形 3）：假设 $m=0$。则

$$\begin{aligned} L_{\mathsf{coal}(f)} &= 1 : 0 \\ &= \mathsf{coal}(() : ()) \\ &= \mathsf{coal}(L_{\mathsf{coal}^\flat(f^\flat)}) \\ &= \mathsf{coal}(\mathsf{coal}^\flat(L_{f^\flat})) \\ &= \mathsf{coal}((L_f)^\flat) \\ &= \mathsf{coal}(L_f). \end{aligned}$$

□

3.2.6.3 补

在本节中，我们定义互补嵌套元组态射的概念。

定义 3.2.6.14. 假设 $f\colon S \to T$ 和 $g\colon U \to T$ 是像不相交的嵌套元组态射。若 g 是 f 的补，则

$$(f,g) : (S,U) \to T$$

是一个同构。

注记 3.2.6.15. 如果 $f\colon S \to T$ 和 $g\colon U \to T$ 是嵌套元组态射，那么 g 是f的补，当且仅当 $g^\flat$ 是 $f^\flat$ 的补，因为 $(f,g)^\flat = f^\flat \star g^\flat$。

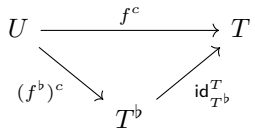
命题 3.2.6.16。 如果 $f\colon S \to T$ 是一个嵌套元组态射，且 $g\colon U \to T$ 是 f 的补，那么 L_g 是 L_f 的 大小（T）-补。

证明。 观察3.2.2.5 表明

$$(L_f)^\flat = L_{f^\flat}, \text{ and}$$
$$(L_g)^\flat = L_{g^\flat}$$

以及引理 2.3.6.2 使我们能够归约到平坦情形（命题 3.1.5.42）。 □

构造 3.2.6.17。 设 $f : S \to T$ 是一个嵌套的嵌套元组态射。我们将的补 定义为复合



其中 $(f^\flat)^c$，是 如构造 3.1.5.46中所定义，且 $\mathsf{id}^T_{T^\flat} \colon T^\flat \cong T$是反扁平化同构 .

例 3.2.6.18。 该嵌套元组态射的补

$$((2,2),(5,5)) \xrightarrow[(1,4,2,5)]{f} ((2,5,7),(2,5,7))$$

is

$$(7,7) \xrightarrow[(3,6)]{f^c} ((2,5,7),(2,5,7)).$$

命题 3.2.6.19。 设 $f : S \to T$ 和 $g : U \to T$ 是嵌套元组态射。若 f 是单射，且 g是 f的补，那么 L_g是 L_f的大小(T)-补。

证明。 由命题 3.1.5.42 和引理 2.3.6.2 可得

$$(L_f)^\flat = L_{f^\flat}$$
$$(L_g)^\flat = L_{g^\flat}.$$

□

命题 3.2.6.20。 若 $f : S \to T$ 是单射嵌套元组态射，则

$$\mathsf{coal}(L_{f^c}) = \mathsf{comp}(L_f, \mathsf{size}(T)).$$

证明。 由于 f^c 是由 $(f^\flat)^c$ 后复合一个重新加括号同构得到的，因此

$$L_{f^c} = L_{(f^\flat)^c}$$

因此根据命题 3.2.6.20，可得

$$\mathsf{coal}^\flat(L_{f^c}) = \mathsf{comp}^\flat(L_f, \mathsf{size}(T)).$$

对两边应用 coal(–)，即可得到结果。 □

3.2.6.4 复合

我们可以利用第 3.2.4 节的实现函子，证明嵌套元组态射的复合与相关布局的复合相兼容。

Theo注 3.2.6.21. 如果 f 和 g 是非退化的可复合嵌套元组态射，则 n

$$L_{g\circ f} = L_g \circ L_f.$$

证明。 设 $f\colon S \to T$ 和 $g\colon T \to U$ 是非退化嵌套元组态射。我们需要验证

1. $\mathsf{shape}(L_{g\circ f})$ 细化 $\mathsf{shape}(L_f)$：这是成立的，因为

$$\mathsf{shape}(L_f) = S = \mathsf{shape}(L_{g\circ f}).$$

2. $L_{g\circ f}$ 在 *shape()* 上合并： 这是成立的，因为嵌套元组态射 $g \circ f$ 是非退化的，因此布局 $L_{g\circ f}$ 也是非退化的。
3. $\Phi_{L_{g\circ f}} = \Phi_{L_g} \circ \Phi^{\mathsf{size}(L_g)}{}_{L_f}$： 根据引理 3.2.4.2，我们有

$$\begin{aligned}\Phi^{\mathsf{size}(U)}_{L_{g\circ f}} &= |g \circ f| \\ &= |g| \circ |f| \\ &= \Phi^{\mathsf{size}(U)}_{L_g} \circ \Phi^{\mathsf{size}(T)}_{L_f}\end{aligned}$$

并通过与包含映射 $[0, \mathsf{size}(U)) \subset \mathbb{Z}$ 后复合，并注意到 $\mathsf{size}(T) = \mathsf{size}(L_g)$，即可得出结果。

□

3.2.6.5 逻辑除法

接下来，我们介绍嵌套元组态射的逻辑除法。该构造通过引入嵌套轮廓从扁平除法得到，不带任何相容性约束。

定义 3.2.6.22. 设 f 和 g 是嵌套元组态射。若 g 和 f 可复合，我们称 g 整除 f。换言之，

$$\mathsf{codomain}(g) = \mathsf{domain}(f).$$

定义 3.2.6.23. 设 $g\colon S \to T$ 和 $f\colon T \to U$ 是嵌套元组态射。我们将 f 除以 g 的逻辑除法定义为嵌套元组态射

$$f \oslash g = f \circ (g, g^c).$$

例 3.2.6.24。 逻辑除法

$$((2,2),2) \xrightarrow[(2,4,*)]{f} ((4,2),(4,2))$$

by

$$(2,2) \xrightarrow[(1,3)]{g} ((2,2),2)$$

is

$$((2,2),2) \xrightarrow[(2,*,4)]{f\oslash g} ((4,2),(4,2)).$$

例 3.2.6.25。 逻辑除法

$$(8,8,512,512,512) \xrightarrow[(*,*,1,2,3)]{f} (512,512,512)$$

by

$$(8,512) \xrightarrow[(1,5)]{g} (8,8,512,512,512)$$

is

$$((8,512),(8,512,512)) \xrightarrow[(*,1,*,2,3)]{f\oslash g} ((4,2),(4,2)).$$

命题 3.2.6.26. 如果 $g\colon S \to T$ 和 $f\colon T \to U$ 是非退化嵌套元组态射，那么

$$\mathsf{coal}(L_{f\oslash g}) = \mathsf{coal}(L_f \oslash L_g).$$

证明。 由命题 3.2.6.20，我们有

$$\mathsf{coal}(\mathsf{comp}(L_g, \mathsf{size}(L_f))) = \mathsf{coal}(L_{g^c})$$

并且我们计算得

$$\begin{aligned}
\mathsf{coal}(L_f \oslash L_g) &= \mathsf{coal}(L_f \circ (L_g, \mathsf{comp}(L_g, \mathsf{size}(L_f)))) \\
&= \mathsf{coal}(L_f \circ (L_g, L_{g^c})) \\
&= \mathsf{coal}(L_f \circ L_{(g,g^c)}) \\
&= \mathsf{coal}(L_f \circ L_{(g,g^c)}) \\
&= \mathsf{coal}(L_{f\circ(g,g^c)}) \\
&= \mathsf{coal}(L_{f\oslash g}).
\end{aligned}$$

□

命题 3.2.6.27. 如果 f 和 g 是嵌套元组，且 g 整除 f，那么

$$(f \oslash g)^\flat = f^\flat \oslash^\flat g^\flat.$$

证明。 我们计算得

$$\begin{aligned}
(f \oslash g)^\flat &= (f \circ (g, g^c))^\flat \\
&= f^\flat \circ (g, g^c)^\flat \\
&= f^\flat \circ (g^\flat \star (g^c)^\flat) \\
&= f^\flat \circ (g^\flat \star (g^\flat)^c) \\
&= f^\flat \oslash^\flat g^\flat.
\end{aligned}$$

□

3.2.6.6 逻辑乘积

在本节中，我们定义嵌套元组态射的逻辑乘积。

定义 3.2.6.28。 假设 f 和 g 是嵌套元组态射。如果陪域(g) = 定义域(f^c)，则称 f 和 g 是乘积可容许 的。若 f 和 g 乘积可容许，则将 f 和 g 的逻辑乘积定义为以下嵌套元组态射

$$f \otimes g = (f, f^c \circ g).$$

例 3.2.6.29。 嵌套元组态射

$$(8,8) \xrightarrow[(1,2)]{f} (8,8,16,16)$$

and

$$(16,16) \xrightarrow[(1,2)]{g} (16,16)$$

是乘积可容许的，并且它们的逻辑乘积为

$$((8,8),(16,16)) \xrightarrow[(1,2,3,4)]{f\otimes g} (8,8,16,16).$$

例 3.2.6.30. 嵌套元组态射

$$(128,128) \xrightarrow[(3,4)]{f} (32,32,128,128)$$

and

$$(32) \xrightarrow[(2)]{g} (32,32)$$

是乘积可容许的，并且它们的逻辑乘积为

$$((128,128),(32)) \xrightarrow[(3,4,2)]{f\otimes g} (32,32,128,128).$$

命题 3.2.6.31. 设 f 和 g 是非退化嵌套元组态射，且 f 和 g是乘积可容许的。那么

$$L_{f\otimes g} = L_f \otimes L_g.$$

证明。 设 $f\colon S \to T$和 $g\colon U \to V$是乘积可容许的，并令

$$L_f^* = \mathsf{comp}(L_f, \mathsf{size}(L_f)\cdot \mathsf{cosize}(L_g))$$

由于 f是单射，且 $\mathsf{codomain}(g) = \mathsf{domain}(f^c)$，因此可得

$$\mathsf{size}(L_f)\cdot \mathsf{cosize}(L_g) \leq \mathsf{size}(S)\cdot \mathsf{size}(V) = \mathsf{size}(T).$$

利用这一事实，以及

$$\Phi_{\mathsf{comp}(L_f,\mathsf{size}(T))} = \Phi_{L_{f^c}},$$

我们有

$$\begin{aligned} L_f^* \circ L_g &= \mathsf{comp}(L_f, \mathsf{size}(T)) \circ L_g \\ &= L_{f^c} \circ L_g. \end{aligned}$$

利用这一事实，我们计算

$$\begin{aligned} L_f \otimes L_g &= (L_f, L_f^* \circ L_g) \\ &= (L_f, L_{f^c} \circ L_g) \\ &= (L_f, L_{f^c \circ g}) \\ &= L_{(f, f^c\circ g)} \\ &= L_{f\otimes g} \end{aligned}$$

□

第4章

计算

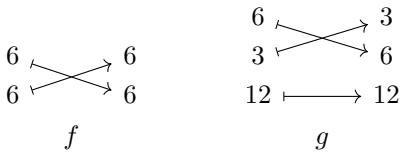
范畴**Tuple** 和**Nest** 为使用可处理布局进行计算提供了强大的框架。然而，在实际应用中，人们经常会遇到这样的可处理布局 A 和 B ：它们在 `cute` 的语境下是可复合的，但其标准表示既无法在 **Tuple** 中复合，**也无法在 Nest 中复合**。本章旨在阐明，借助**Tuple** 和**Nest** 这两个范畴，如何仍然能够利用复合、逻辑除法和逻辑乘积的概念来计算可处理布局的**相互细化**。我们将在第4.1.1节介绍这一概念，在算法4.1.1中给出计算相互细化的算法，并通过许多具体示例进行详细说明。

4.1 可处理布局的复合

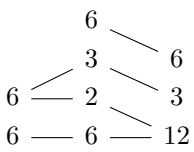
假设我们要计算可处理布局的复合 $B \circ A$

$$A = (6,6) : (6,1),$$
$$B = (12,3,6) : (1,72,12).$$

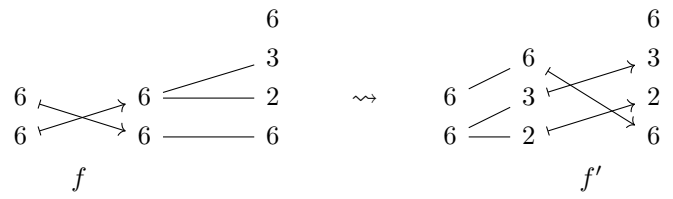
我们可以尝试通过计算 A 和 B 的标准表示 f 和 g 的复合，来计算 $B \circ A$ ：

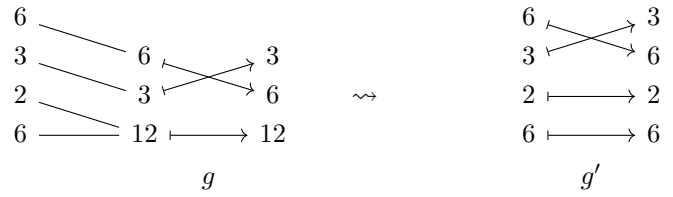


然而，这些态射不可复合，因为 f 的陪域 (6,6) 不等于 g 的定义域 (12,3, 6)。这意味着我们无法直接使用态射 f 和 g 来计算复合 $B \circ A$。不过，我们可以通过求出 (6,6) 和 (12, 3,6) 的相互细化，继续进行计算，如下图所示

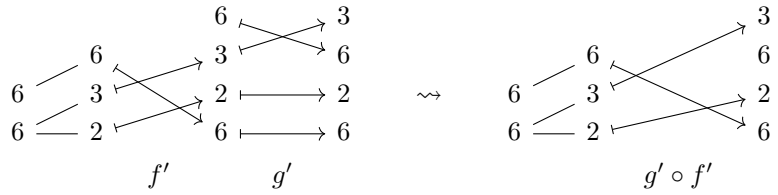


这是一个将 f 和 g 转换为可复合态射 f' 和 g' 的设备：





态射 f' 和 g' 可复合，因此我们可以构造复合态射



而计算编码布局可得

$$B \circ A = L_{g' \circ f'} = ((2,3),6) : ((6,72),1).$$

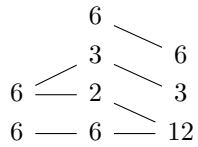
本节的目标是将这一计算过程形式化为一种用于计算可处理布局 A and B 的复合的算法 。正如例子所示，计算中的关键步骤是

1. 找到某些（嵌套）元组的相互细化，以及
2. 使用相互细化，将 f and g 转换为可复合态射 f' and g'。

接下来的两节将详细阐述这些步骤。

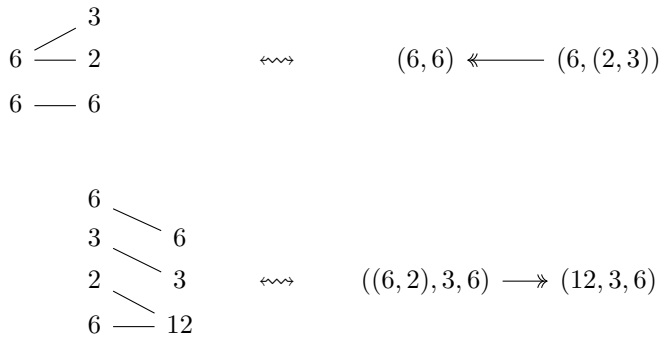
4.1.1 相互细化

在使用第 3 章的范畴框架给出相互细化的精确定义之前，我们先作一个非正式概述。考虑我们启发性示例中的元组 (6，6) 和 (12，3，6)。我们断言，图表



是 (6, 6) 和 (12, 3,6) 的相互细化。我们可以如下更精确地描述这种相互细化。图表的左半部分表示细化 (6,6) ↞ (6,(2,3))，而

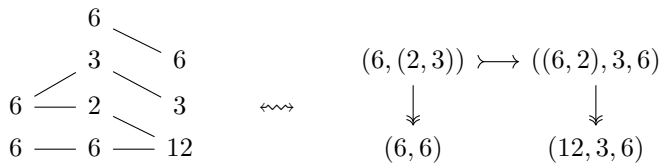
图表的右半部分表示细化 ((6, 2), 3,6) ↠ (12,3, 6):



图表的两半可以拼接在一起，这一事实对应于嵌套元组 (6、(2、3)) 整除((6、2)、3、6)，我们将其记作

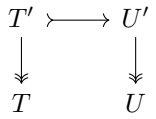
$$(6,(2,3)) \rightarrowtail ((6,2),3,6).$$

将这些观察结合起来，我们可以将相互细化精确地表示为 s



其中，我们选择将细化 (6、6) ↞ (6、(2、3)) 和 ((6、2)、3、6) ↠ (12、3、6) 垂直绘制。现在我们可以给出相互细化的精确定义。

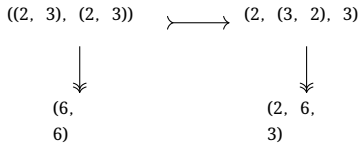
定义 4.1.1.1。 设 T和 U是嵌套元组。(T、 U) 的相互细化是形如以下形式的图表



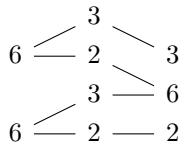
明确地说，这是一对嵌套元组 (T′,U′)，满足

1. T′细化 T，2.
   U′细化 U，以及
3. T′整除 U′。

例 4.1.1.2。 T = (6， 6) 与 U = (2， 6， 3) 的相互细化为



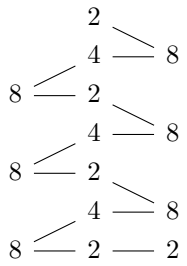
我们将这种相互细化表示如下。



例 4.1.1.3. $T = (8,8,8)$ 和 $U = (2,8,8,8)$ 的一种相互细化为

$$\begin{array}{ccc} ((2,4),(2,4),(2,4)) & \rightarrowtail & (2,(4,2),(4,2),(4,2)) \\ \downarrow & & \downarrow \\ (8,8,8) & & (2,8,8,8) \end{array}$$

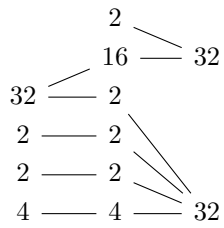
我们将这种相互细化表示如下。



例 4.1.1.4. $T = (4,2,2,32)$ 和 $U = (32,32)$ 的一种相互细化如下

$$\begin{array}{ccc} (4,2,2,(2,16)) & \rightarrowtail & ((4,2,2,2),(16,2)) \\ \downarrow & & \downarrow \\ (4,2,2,32) & & (32,32) \end{array}$$

我们将这种相互细化表示如下。



例 4.1.1.5. 如果 $T = (8,8)$ 和 $U = (3,8,8)$，那么 T 和 U 不存在相互细化。

例 4.1.1.6. 如果 T 和 U 是元组，且 $\mathsf{size}(T) = 2^k$，$\mathsf{size}(L) = 2^\ell$，其中 $k \leq \ell$，那么 T 和 U 存在相互细化。更一般地，如果 T 和 U 是元组，且 $\mathsf{size}(T) \leq \mathsf{size}(U)$ 是某个固定整数的幂，那么 T 和 U 存在相互细化。

观察 4.1.1.7. 在前面的每个示例中，我们都考虑了平坦 元组 T 和 U之间的相互细化。然而，相互细化的定义允许 T 和 U 是任意的嵌套元组。无论如何，将范围限制在平坦情形并不失一般性，因为一对嵌套元组 (T,U) 的相互细化，与它们的展平 $(T^\flat, U^\flat)$ 的相互细化之间存在一一对应关系（见引理 3.2.5.20）。特别地，存在 (T,U) 的相互细化，当且仅当存在 $(T^\flat, U^\flat)$ 的相互细化。

完成适当的定义后，我们给出一种算法，用于计算 (T,U) 的相互细化。

算法 4.1.1：相互细化算法

输入： 嵌套元组 T 和 U。

输出： (T, U) 的一个相互细化 (T', U')（如果存在），否则为 **无**。

$X \leftarrow T$; $Y \leftarrow U$

X', Y', X_{mode}, $Y_{\text{mode}} \leftarrow ()$

$i \leftarrow 1$; $j \leftarrow 1$

当 $i \leq$ 长度(X) **且** $j \leq$ 长度(Y) **时执行**

  如果 条目$_i(X)$ $=$ 条目$_j(Y)$ **那么**

    将条目$_i(X)$追加到 X_{mode}；将 X_{mode} 追加到 X'； $X_{\text{mode}} \leftarrow ()$

    将条目$_j(Y)$追加到 Y_{mode}；

    将 Y_{mode} 追加到 Y'；

    $Y_{\text{mode}} \leftarrow ()$

    $i \leftarrow i + 1$;

    $j \leftarrow j + 1$

  否则如果条目$_i(X)$ 整除 条目$_j(Y)$ **则**

    将条目$_i(X)$追加到 X_{mode}；

    将 X_{mode} 追加到 X'；

    $X_{\text{mode}} \leftarrow ()$

    将条目$_i(X)$追加到 Y_{mode}

    条目$_j(Y)$ $\leftarrow$ 条目$_j(Y)$/条目$_i(X)$;

    $i \leftarrow i + 1$

  否则如果条目$_j(Y)$ 整除 条目$_i(X)$ **则**

    将条目$_j(Y)$追加到 X_{mode};

    将条目$_j(Y)$追加到 Y_{mode};

    将 Y_{mode} 追加到 Y';

    $Y_{\text{mode}} \leftarrow ()$

    条目$_i(X)$ $\leftarrow$ 条目$_i(X)$/条目$_j(Y)$;

    $j \leftarrow j + 1$

  else

    返回 None

  结束 if

结束 while

如果 $Y_{\text{mode}} \neq ()$ **则**

  将条目$_j(Y)$追加到 Y_{mode};

  将 Y_{mode} 追加到 Y';

  $j \leftarrow j + 1$

结束 if

当 $j <$ 长度(Y) **时执行**

  将条目$_j(Y)$追加到 Y';

  $j \leftarrow j + 1$

结束 while

$T' \leftarrow (X')^{\text{prof}(T)}$;

$U' \leftarrow (Y')^{\text{prof}(U)}$

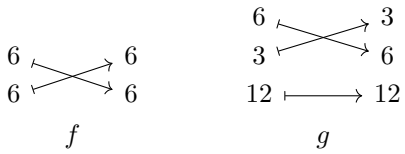
返回 (T', U')

4.1.2 从相互细化到可复合态射

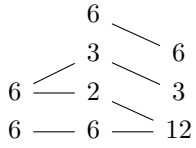
回想一下，为了计算 $B \circ A$ 的复合

$$A = (6,6) : (6,1) \text{ and}$$
$$B = (12,3,6) : (1,72,12),$$

我们构造了元组态射

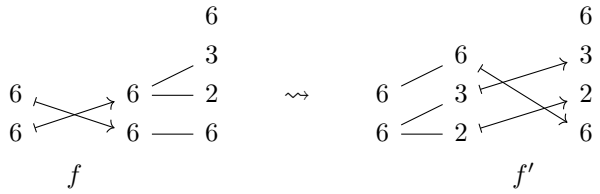


以及一个相互细化。

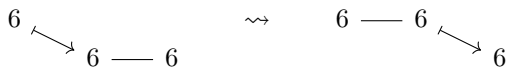


计算的下一步是利用相互细化，将 f 和 g 转换为可复合态射 f' 和 g'。在给出这一过程的正式范畴论定义之前，我们先通过一个例子来说明这个过程。

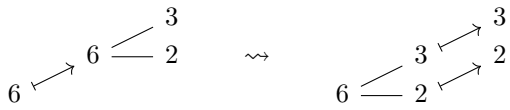
我们利用 f 和相互细化的左半部分构造 f'：



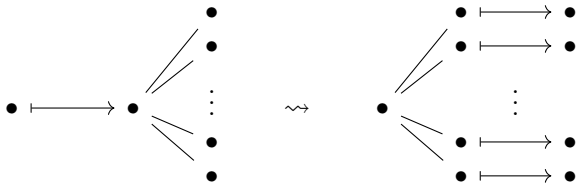
该构造是通过进行替换



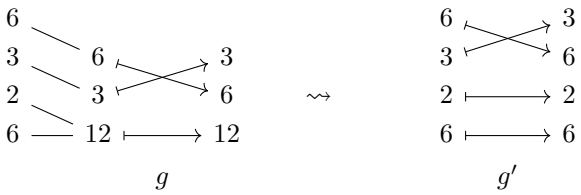
并进行替换



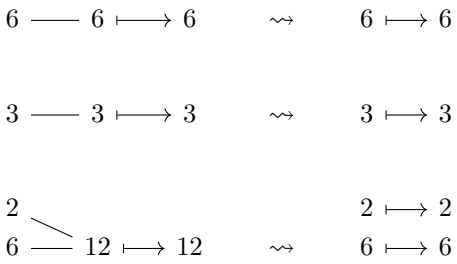
更一般地，我们进行如下替换



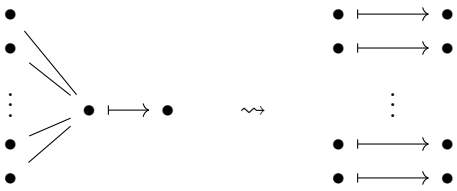
从 g 构造 g' 的过程，以及我们相互细化的右半部分，都是类似的。



此构造通过进行以下替换得到

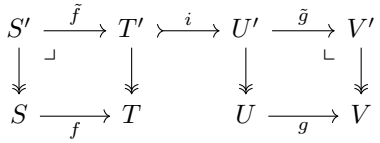


更一般地，我们进行以下替换



给出了过程的非正式描述后，我们将其精确定义如下。

构造 4.1.2.1. 设 $f\colon S \to T$和 $g\colon U \to V$是嵌套元组态射，并且 (T, U') 是 (T, U) 的相互细化。那么我们可以使用第*3.2.5* 节的拉回和推前构造来形成图表：



˜若令 $f' = i \circ f$和 $g' = \tilde{g}$，则



是可复合嵌套元组态射。

4.1.3 复合算法

算法 4.1.2：可处理布局复合算法[1] **输入：** 可处理布局 A和 B。

算法 4.1.2（续）：可处理布局合成算法

输出：若存在，则为 C 与 A 和 B的弱复合，否则为**无**。

取标准表示

$$S \xrightarrow{\quad f \quad} T \qquad\qquad U \xrightarrow{\quad g \quad} V$$

分别对应于 A 和 $\mathsf{coal}(B)$。

使用算法 4.1.1 生成相互细化

$$\begin{array}{ccc} T' & \rightarrowtail & U' \\ \downarrow & & \downarrow \\ T & & U \end{array}$$

（T，U）。如果不存在（T，U）的相互细化，则返回 **None**。使用构造 4.1.2.1 获得可复合嵌套元组态射

$$S' \xrightarrow{\quad f' \quad} U' \xrightarrow{\quad g' \quad} V'$$

复合 f' 和 g'，并计算编码布局

$$C = L_{g' \circ f'}$$

返回 C

定理 4.1.3.1。若 A 和 B 是可处理布局，则前述算法的输出 C 是 A 和 B的弱复合。因此，

$$B \circ A = \mathsf{coal}(C, \mathsf{shape}(A)).$$

证明。命题 3.2.5.15 告诉我们

$$\Phi_{L_{g'}} = \Phi_{L_g} = \Phi_{\mathsf{coal}(B)} = \Phi_B,$$

以及命题3.2.5.11 和例 3.1.3.6 告诉我们

$$\Phi_{L_{f'}} = \Phi_{L_f} = \Phi_A.$$

定理 3.2.6.21则推出

$$\begin{aligned} \Phi_C = \Phi_{L_{g' \circ f'}} &= \Phi_{g'} \circ \Phi_{f'}^{\mathsf{size}(U')} \\ &= \Phi_B \circ \Phi_A^{\mathsf{size}(B)}. \end{aligned}$$

根据构造，$L_{f'}$ 的形状 S' 细化了 A的形状 S，因此我们得出结论：C 是 A 和 B的弱复合。

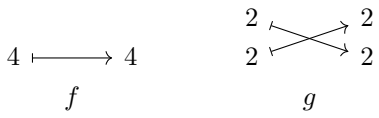
□

4.1.4 示例

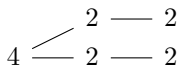
本节将说明如何使用算法 4.1.3 计算可处理布局 $B \circ A$ 的复合 A 与 B。

例 4.1.4.1。 假设 $A = (4) : (1)$，且 $B = (2,\ 2) : (2,\ 1)$。

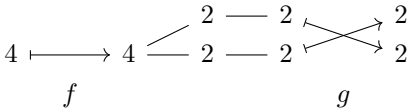
1. 取 A 和 coal(B) $= B$ 的标准表示。



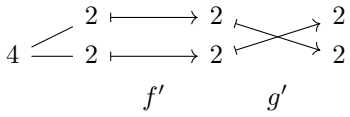
2. 应用算法 4.1.1 以获得相互细化



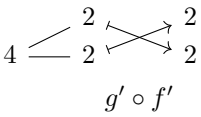
3. 构建图表



4. 求解图表



5. 组合 f' 和 g' 以获得



6. 计算相关布局

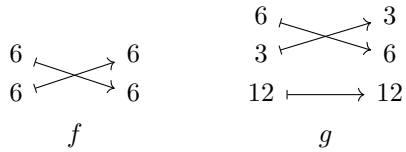
$$L_{g'\circ f'} = ((2,2)) : ((2,1)).$$

7. $L_{g'\circ f'}$ 在 (4) 上合并，因此

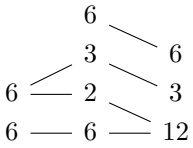
$$B \circ A = ((2,2)) : ((2,1)).$$

例 4.1.4.2. 假设 $A = (6,6) : (6,1)$，且 $B = (12,3,6) : (1,72,12)$。

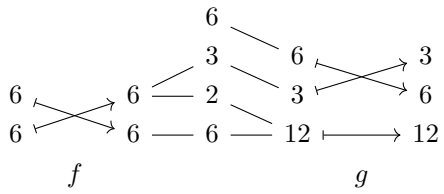
1. 取 A 和 coal(B) $= B$ 的标准表示。



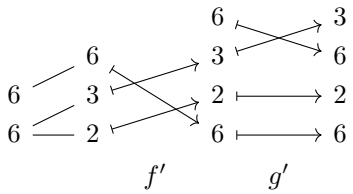
2. 应用算法 4.1.1 以获得相互细化



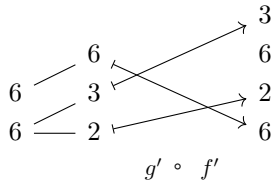
3. 构建图表



4. 求解图表以获得



5. 组合 f' 和 g' 以获得



6. 计算相关布局

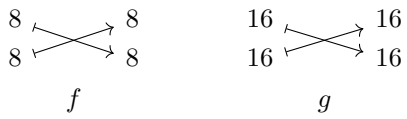
$$L_{g'\circ f'} = ((2,3),6) : ((6,72),1).$$

7. $L_{g'\circ f'}$ 在 (6,6) 上合并，因此

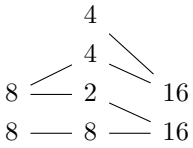
$$B \circ A = ((2,3),6) : ((6,72),1).$$

例 4.1.4.3. 假设 $A = (8,8) : (8,1)$，且 $B = (16,16) : (16,1)$。

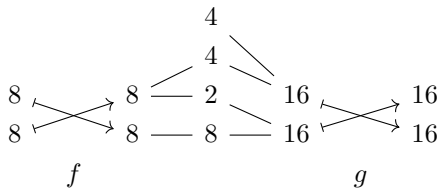
1. 取 A 和 coal(B) $= B$ 的标准表示。



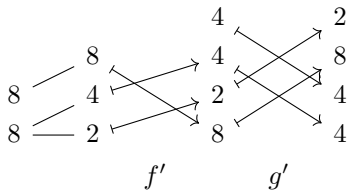
2. 应用算法 4.1.1 以获得相互细化



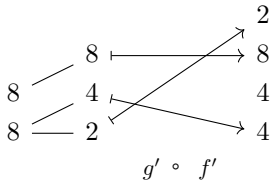
3. 构造图表



4. 求解图表，得到



5. 组合 f' 和 g'，得到



6. 计算相关布局

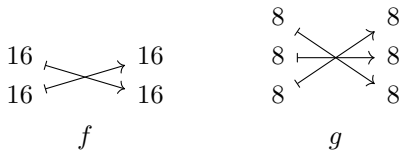
$$L_{g'\circ f'} = ((2,4),8) : ((128,1),16)$$

7. $L_{g'\circ f'}$ 在 (8,8) 上合并，因此

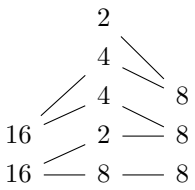
$$B \circ A = ((2,4),8) : ((128,1),16)$$

例 4.1.4.4. 假设 $A = (16,16) : (16,1)$，且 $B = (8,8,8) : (64,8,1)$。

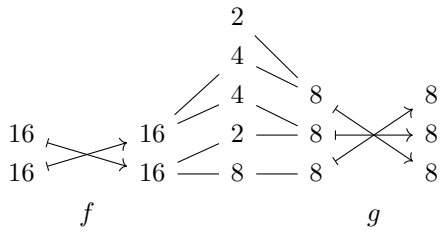
1. 采用 A 和 coal(B) $= B$ 的标准表示。



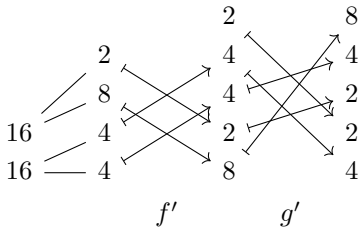
2. 应用算法4.1.1 以获得相互细化



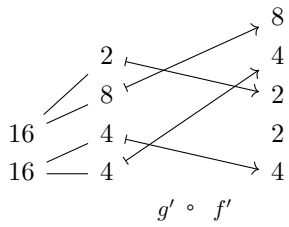
3. 构建图表



4. 解析图表，得到



5. 组合 f' 和 g'，得到



6. 计算相关布局

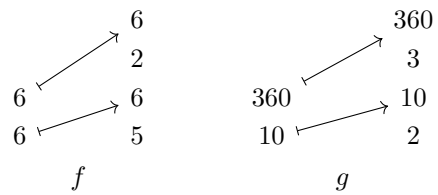
$$L_{g'\circ f'} = ((4,4),(8,2)) : ((16,1),(64,8)).$$

7. $L_{g'\circ f'}$ 在 (16,16) 上合并，因此

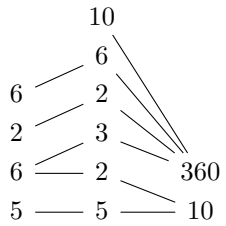
$$B \circ A = ((4,4),(8,2)) : ((16,1),(64,8)).$$

例 4.1.4.5. 假设 $A = (6,6) : (5,60)$，且 $B = (10,360) : (2,60)$。

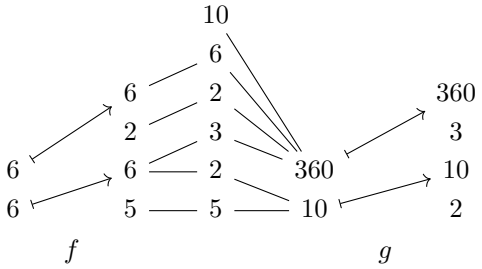
1. 取 A 和 coal(B) $= B$ 的标准表示。



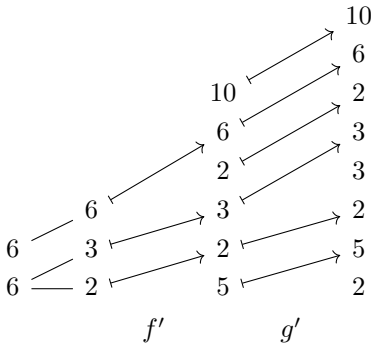
2. 应用算法4.1.1 以获得相互细化



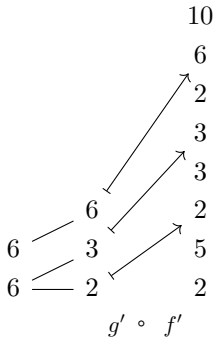
3. 构建图表



4. 求解图表以得到



5. 组合 f' 和 g' 以得到



6. 计算相关布局

$$L_{g'\circ f'} = ((2,3),6) : ((10,60),360).$$

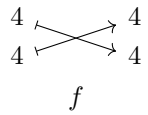
7. 布局 $L_{g'\circ f'}$ 在 (6,6) 上是合并的，因此

$$B \circ A = ((2,3),6) : ((10,60),360).$$

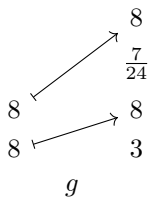
4.1.5 更一般的复合

我们所建立的图形演算可以自然地扩展，用于计算可处理布局 A 与任意 CuTe 布局 B的复合 $B \circ A$。简单来说，我们通过允许元组的条目取自 $\mathbb{Q}_{>0} \supset \mathbb{Z}_{>0}$来实现这一点。下面用一个示例计算来说明这一扩展。

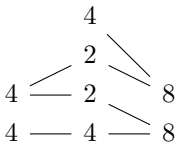
考虑布局 $A = (4, 4) : (4, 1)$ 和 $B = (8, 8) : (3, 7)$。布局 A是可处理的，其标准表示是下方所示的元组态射 f。



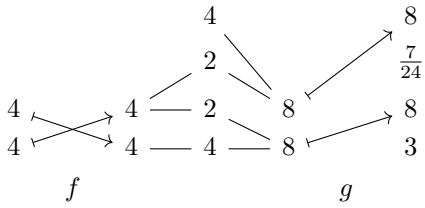
布局 B 不可处理，但我们仍可用图表表示 B



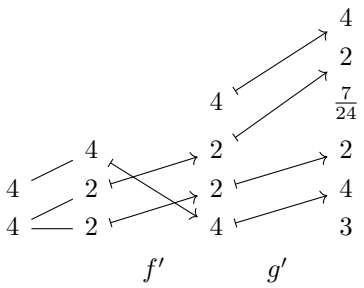
这个图表并不对应一个真正的元组态射，因为“余域元组”$(3, 8, 7\,24, 8)$含有非整数条目。不过，它仍通过通常的前缀乘积公式编码了布局 B，并且仍可作为输入用于我们的复合算法：我们可以应用算法 4.1.1来得到相互细化



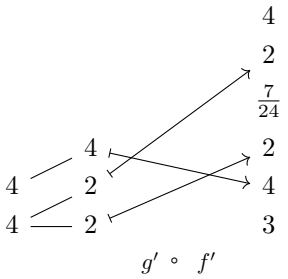
构成图表



解析此图表以获得



并组合 f' 和 g' 以获得该图表



编码布局为 ((2，2)，4) : ((12，7)，3)，它在 (4，4) 上是合并的，因此我们得出

$$B \circ A = ((2,2),4) : ((12,7),3).$$

4.1.6 复合可容许性

在 [16], 中，作者引入了复合可容许性这一概念，它是布局 $B \circ A$ 与 A 复合 B 存在的充分条件。我们来回顾一下复合可容许性的定义。与 [16], 中一样，我们将注意力限制在形状条目不等于 1 的平坦布局上，并假设复合中的第一个布局没有等于 0 的步长。

定义 4.1.6.1。设

$$A = (s_1, \ldots, s_m) : (d_1, \ldots, d_m)$$

是一个没有 $s_i = 1$ 且没有 $d_i = 0$ 的平坦布局。设 B 是一个具有以下特征的平坦布局

$$\mathsf{shape}(B) = (u_1, \ldots, u_p).$$

我们称 A 和 B 可组合，如果满足以下条件。

1. 对每个 $1 \le i \le m$，存在 $1 \le k \le \ell \le p$ 使得

   (a) $u_1 \cdots u_{k-1}$ 整除 d_i，(b) d_i 整除 $u_1 \cdots u_k$（若 $k < p$，则为真整除），(c) $u_1 \cdots u_{\ell-1}$ 整除 $s_i d_i$，(d) $s_i d_i$ 整除 $u_1 \cdots u_\ell$（若 $\ell < p$，则为真整除）。

2. 这些区间

   $$[d_i, d_i(s_i - 1)] \cap [1, s_1 \cdots s_{m-1})$$

   两两不相交。

注记 4.1.6.2。 上面定义中的指标 k， ℓ 在 [16]中称为“除法指标”。

注记 4.1.6.3. 在 [16], 中，作者使用布局的“扩展布局函数”，它可以视为将最终形状项 s_m 替换为∞ 后得到的布局所对应的布局函数。我们在此给出的定义是适用于普通布局函数的对应版本。

引理 4.1.6.4. 设 $T = (t_1, \ldots, t_n)$ 和 $U = (u_1, \ldots, u_p)$ 是正整数元组，并且假设 (T, U) 存在相互细化。那么，对于 T和 U的任意前缀乘积 $t_1 \cdots t_j$和 $u_1 \cdots u_k$，分别有

1. $t_1 \cdots t_j$ *is greater than* $u_1 \cdots u_k$, *or*

2. $t_1 \cdots t_j$ *divides* $u_1 \cdots u_k$.

证明。选择一个相互细化 $(T\prime U')$，使其细化 (T, U)，并将 $U\prime$的扁平化记为 $(u'_1, \ldots, u'_{p'})$。T或 U的任意前缀乘积也是 U'的前缀乘积；由于固定正整数元组的前缀乘积满足 $x \leq y \Rightarrow x \mid y$，结论得证。□

定理 4.1.6.5. 假设 A 是一个平坦可处理布局，其中没有形状条目等于1，也没有步幅条目等于0。假设 B 是一个平坦可处理布局。令 $f\colon S \to T$ 和 $g\colon U \to V$ 分别表示 A 和$\mathsf{coal}(B)$的标准表示。如果 T 和 U 存在相互细化，则 A和 B 可组合。

证明。记 $S = (s_1, \ldots, s_m)$, $T = (t_1, \ldots, t_n)$, $U = (u_1, \ldots, u_p)$，并记 $\alpha : \langle m \rangle_* \to \langle n \rangle_*$为 f 所在映射。我们需要检查定义 4.1.6.1 中的条件是否成立。

1. 假设 $1 \leq i \leq m$。那么 $d_i = t_1 \cdots t_{j-1}$，其中某个 j满足条件，即 $j = \alpha(i)$。假设我们有 (T, U) 的相互细化 $(T\prime U\prime)$，并记 $(T\prime)^\flat = (t'_1, \ldots, t'_{n'})$ 和 $(U')^\flat = (u'_1, \ldots, u'_{p'})$。

   - (a) 和 (b)：由于 $T\prime$细化 T，因此存在某个 $1 \leq a \leq n\prime$使得

   $$d_i = t_1 \cdots t_{j-1} = t'_1 \cdots t'_a = u'_1 \cdots u'_a.$$

   取最大的 $k \in \langle p \rangle$，使得 $u_1 \cdots u_{k-1} \leq u'_1 \cdots u'_a$。

   – 假设 $k < p$。我们观察到

   $$u_1 \cdots u_{k-1} \leq d_i < u_1 \cdots u_k.$$

   其中，第二个不等式由 $k \in \langle p \rangle$ 的极大性得出。引理 4.1.6.4 推出 $u_1 \cdots u_{k-1}$整除 d_i，且 d_i整除 $u_1 \cdots u_k$，且为真整除。– 假设 $k = p$。我们观察到

   $$u_1 \cdots u_{k-1} \leq d_i = t_1 \cdots t_{j-1} < t_1 \cdots t_n \leq u_1 \cdots u_p = u_1 \cdots u_k.$$

   引理 4.1.6.4 推出 $u_1 \cdots u_{k-1}$整除 d_i，且 d_i整除 $u_1 \cdots u_k$（实际上是严格整除，但我们不需要这一点）。

   - (c) 和 (d)：同样，由于 $T\prime$细化 T，存在某个 $1 \leq b \leq n\prime$使得

   $$s_i d_i = t_1 \cdots t_j = t'_1 \cdots t'_b = u'_1 \cdots u'_b.$$

   取最大的 $\ell \in \langle p \rangle$，使得 $u_1 \cdots u_{\ell-1} \leq u'_1 \cdots u'_b$。

   – 假设 $\ell < p$。我们观察到

   $$u_1 \cdots u_{\ell-1} \leq s_i d_i < u_1 \cdots u_\ell.$$

   其中，第二个不等式由 $\ell \in \langle p \rangle$ 的极大性得出。引理 4.1.6.4 蕴含$u_1 \cdots u\ell - 1$整除 $s_i d_i$且 $s_i d_i$整除 $u_1 \cdots u_\ell$成立。

   – 假设 $\ell = p$。我们观察到

   $$u_1 \cdots u_{\ell-1} \leq s_i d_i = t_1 \cdots t_j \leq t_1 \cdots t_n \leq u_1 \cdots u_p = u_1 \cdots u_k.$$

   引理 4.1.6.4 意味着 $u_1 \cdots u_{k-1}$整除 d_i且 d_i整除 $u_1 \cdots u_k$。

2. 对任意 $i \neq i'$，属于 $\langle m \rangle$，有 $d_i = t_1 \cdots t_{j-1}$, $s_i = t_j$, $d_{i'} = t_1 \cdots t_{j'-1}$，以及 $s_{i'} = t_{j'}$，其中 $j = \alpha(i)$ 和 $j' = \alpha(i')$。于是我们有

$$[d_i, d_i(s_i - 1)] = [t_1 \cdots t_{j-1}, t_1 \cdots t_{j-1}(t_j - 1)]$$

and

$$[d_{i'}, d_{i'}(s_{i'} - 1)] = [t_1 \cdots t_{j'-1}, t_1 \cdots t_{j'-1}(t_{j'} - 1)]$$

如果 $j' > j$，那么

$$t_1 \cdots t_{j-1}(t_j - 1) < t_1 \cdots t_{j'-1}$$

因此这些区间不会重叠；类似地，如果 $j < j'$。

□

4.2 逻辑除法与逻辑乘积

本节将说明如何使用4.1.3中的复合算法来计算逻辑除法和逻辑乘积。

4.2.1 逻辑除法示例

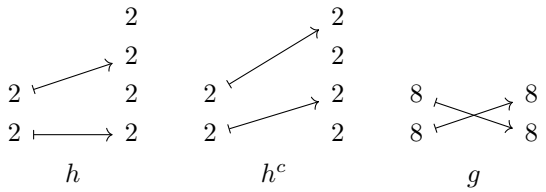
回顾一下，如果 A 和 B 是布局，则逻辑除法 $A \oslash B$ 定义为

$$A \oslash B = A \circ (B, B^c)$$

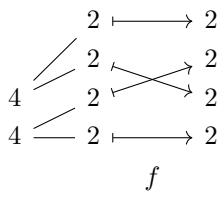
其中

$$B^c = \mathsf{comp}(B, \mathsf{size}(A)).$$

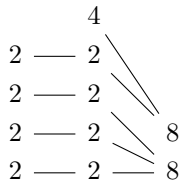
例 4.2.1.1. 假设我们想要计算逻辑除法 $A \oslash B$，其中 $A = (8,8) : (8,1)$ 和 $B = (2,2) : (1,4)$。那么我们可以写成 $A = L_g$、$B = L_h$ 和 $B^c = L_{h^c}$，其中 f 和 f^c 是下方所示的元组态射。



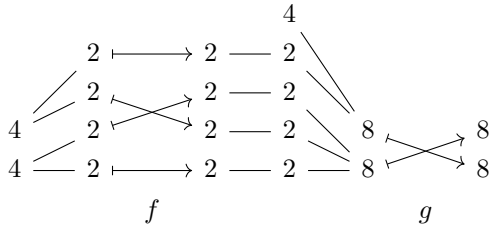
It 由此可知，(B, B^c) 由下方所示的嵌套元组态射 $f = (h, h^c)$ 编码 .



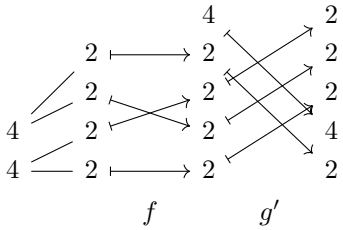
接下来，我们像之前一样执行复合算法。我们使用算法 4.1.1 来求出相互细化



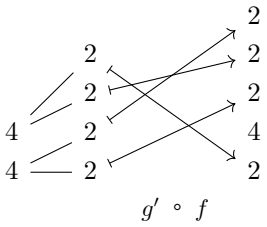
构成该图表



解析该图表以获得



并组合 f 和 g' 以获得



此嵌套元组态射编码的布局为

$$L_{g'\circ f} = ((2,2),(2,2)) : ((8,32),(16,1))$$

它在 (2，2)，(2，2) 上进行了合并，因此我们得出

$$A \oslash B = ((2,2),(2,2)) : ((8,32),(16,1)).$$

4.2.2 逻辑积示例

回顾一下，如果 A 和 B 是布局，则逻辑乘积 $A \otimes B$ 定义为

$$A \otimes B = (A, A^c \circ B)$$

其中

$$A^c = \mathsf{comp}(A, \mathsf{size}(A) \cdot \mathsf{cosize}(B)).$$

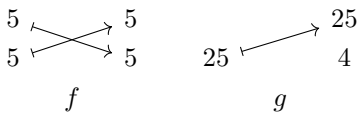
具体来说，如果我们想手动计算 $A \otimes B$，只需计算 $A^c \circ B$，然后将结果与 A 拼接。

例 4.2.2.1. 假设我们想计算逻辑乘积 $A \otimes B$，其中 $A = (2,2) : (1,2)$，且 $B = (5,5) : (5,1)$。那么

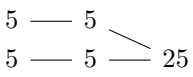
$$\begin{aligned} A^c &= \mathsf{comp}(A, \mathsf{size}(A) \cdot \mathsf{cosize}(B)) \\ &= \mathsf{comp}(A, 100) \\ &= (25) : (4). \end{aligned}$$

我们按照上一节中的方法继续。

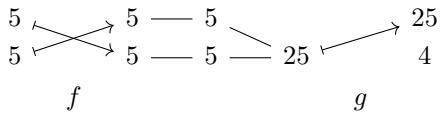
1. 取 B 和 $\mathsf{coal}(A^c) = A^c$的标准表示。



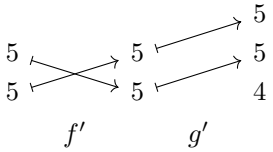
2. 应用算法 4.1.1 以求得相互细化



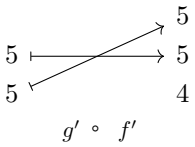
3. 构造图表



4. 求解图表以得到



5. 组合 f' 和 g' 以得到



6. 计算编码布局

$$L_{g' \circ f'} = (5,5) : (20,4).$$

7. 布局 $L_{g' \circ f'}$ 在 (5,5) 上合并，因此

$$A^c \circ B = (5,5) : (20,4).$$

我们得出结论

$$A \otimes B = ((2,2),(5,5)) : ((1,2),(20,4)).$$

附录 A

范畴引言

在本文中，我们会自由运用范畴的语言。范畴是一种数学对象，它抽象了态射及其复合的概念。本附录旨在简明、易懂地介绍范畴的基础知识。具体而言，我们希望回答以下问题：

1. 什么是范畴?
2. 什么是函子?

能够自信地回答这些问题，并且心中有相应示例的人，将能够理解本作品中最重要的概念和构造。对于那些希望学习范畴论中更高级概念（例如 自然变换、伴随和（余）极限）的人，我们推荐 [13]。

A.1 什么是范畴?

我们先来回答第一个问题。在给出定义之前，先看一个启发性的例子。假设 X 和 Y 是集合。一个函数 $f\colon X \to Y$ 会为每个元素 $x \in X$ 指定一个元素 $f(x) \in Y$。我们称 X 为 f 的定义域，称 g 为 Y 的陪域。

例 A.1.0.1. 有一个函数 $f\colon \mathbb{Z} \to \mathbb{Z}$，定义如下

$$f(x) = 2x.$$

例 A.1.0.2. 有一个函数 $g\colon \mathbb{Z} \to \mathrm{Bool}$，其中 $\mathrm{Bool} = \{\mathbf{True}, \mathbf{False}\}$，定义如下

$$g(x) = \begin{cases} \mathbf{True} & x \text{ is even,} \\ \mathbf{False} & x \text{ is odd.} \end{cases}.$$

如果 $f\colon X \to Y$ 和 $g\colon Y \to Z$ 是函数，那么我们可以将 f 和 g 进行复合：f 和 g 的复合是函数 $g \circ f\colon X \to Z$，定义如下

$$(g \circ f)(x) = g(f(x)).$$

例 A.1.0.3. 若 f 和 g 分别是示例 A.1.0.1 和 A.1.0.2 中的函数，那么复合函数 $g \circ f\colon \mathbb{Z} \to$ 布尔值 定义如下：

$$(g \circ f)(x) = \mathbf{True}.$$

函数复合满足两个基本性质。首先，复合满足结合律：若 f和 g 可复合，且 g 和 h 可复合，则

$$h \circ (g \circ f) = (h \circ g) \circ f.$$

其次，每个集合 X 都有一个恒等函数 $\mathsf{id}_X : X \to X$，定义为

$$\mathsf{id}_X(x) = x.$$

若 $f : X \to Y$ 是任意函数，那么与 id_X 前复合或与 id_Y 后复合，都会使该函数 f 保持不变：

$$f \circ \mathsf{id}_X = f = \mathsf{id}_Y \circ f.$$

在纯数学与应用数学中，有许多情况会涉及某个**对象**的**集合**，以及这些对象之间的**态射**，它们具有与集合和函数相同的形式性质：态射可以按满足结合律的方式复合，而对象则具有恒等态射。虽然集合之间的函数是典型例子，但范畴中的对象不一定是集合，范畴中的态射也不一定是函数。我们稍后会看到许多这样的例子。为了概括这种反复出现的结构，我们定义**范畴**这一概念。

定义 A.1.0.4. 一个范畴**C** 由以下部分组成

1. 一个对象集合：

$$\mathsf{ob}(\mathbf{C}) = \{X, Y, Z, \dots\}.$$

这些对象可以是集合、元组、数、向量空间、矩阵，或其他数学结构，具体取决于范畴**C**。

2. 这些对象之间的一组态射：

$$\mathsf{mor}(\mathbf{C}) = \{f, g, h, \dots\}.$$

每个 m 态射 f： $X \to Y$ 在**C** 中都有一个定义域 X 和一个陪域 Y，它们都是 **C**.

3. 一个复合规则：若 $f : X \to Y$ 和 g： $Y \to Z$ 是**C**中的态射，则存在一个态射

$$g \circ f : X \to Z$$

称为 f 和 g的复合。**C** 中态射的复合满足结合律，也就是说

$$h \circ (g \circ f) = (h \circ g) \circ f,$$

在有定义时。

4. 恒等态射：如果 X 是**C**中的一个对象，那么存在一个态射

$$\mathsf{id}_X : X \to X$$

称为 X上的恒等态射。如果 $f : X \to Y$ 是**C**中的任意态射，那么

$$f \circ \mathsf{id}_X = f = \mathsf{id}_Y \circ f.$$

我们来看看 来看看一些重要的范畴示例。我们从这个引例开始 .

例 A.1.0.5. 有一个范畴**Set**，其对象是集合，态射是函数。态射的复合由函数复合给出:

$$(g \circ f)(x) = g(f(x))$$

而集合 X 上的恒等态射就是恒等函数

$$\mathsf{id}_X(x) = x.$$

例 A.1.0.6. 存在一个范畴 **Vect**，其对象是 $\mathbb{R}^n$ 上的向量空间，其中 $n \geq 0$ 表示相应参数，而态射是矩阵。具体来说，一个态射

$$A : \mathbb{R}^n \to \mathbb{R}^m$$

在**Vect** 中是一个 $m \times n$ 矩阵 A。**Vect** 中的复合通过矩阵乘积来定义:

$$B \circ A = BA,$$

而 $\mathbb{R}^n$ 上的恒等态射是 $n \times n$ 矩阵

$$\mathsf{id}_{\mathbb{R}^n} = I_n = \begin{bmatrix} 1 & 0 & \cdots & & 0 \\ 0 & 1 & & & \\ \vdots & & \ddots & & \vdots \\ & & & 1 & 0 \\ 0 & & \cdots & 0 & 1 \end{bmatrix}.$$

例 A.1.0.7. 存在一个范畴 **Div**，其对象是整数 $a \geq 1$，并且其中存在唯一一个态射

$$\mathsf{div}_a^b : a \to b$$

当 a 整除 b 时。如果 a 整除 b 且 b 整除 c，那么 a 整除 c，这意味着我们有一个定义良好的复合规则

$$\mathsf{div}_b^c \circ \mathsf{div}_a^b = \mathsf{div}_a^c,$$

以及恒等态射

$$\mathsf{id}_a = \mathsf{div}_a^a$$

存在，因为每个正整数 a 都整除自身。

除了范畴的定义之外，还有一些重要的范畴概念需要我们理解。例如，理解同构的概念很重要，它是对集合双射 概念的推广。

定义 A.1.0.8. 设**C** 是一个范畴，并设 $f\colon X \to Y$ 是**C**中的一个态射。若存在**C** 中的一个态射 $f^{-1}\colon Y \to X$，使得 f满足以下条件，我们就称其为一个同构：

1. $f^{-1} \circ f = \mathsf{id}_X$，以
及 2. $f \circ f^{-1} = \mathsf{id}_Y$。

例 A.1.0.9. 在**Set**范畴中，同构就是一个双射：一个函数 $f\colon X \to Y$，使得对每个 $y \in Y$，都存在唯一的 $x \in X$，满足 $f(x) = y$。例如，函数 $f\colon \mathbb{Z} \to \mathbb{Z}$定义为

$$f(x) = x + 10$$

是一个双射，其逆映射 $f^{-1}\colon \mathbb{Z} \to \mathbb{Z}$ 由下式给出

$$f^{-1}(x) = x - 10.$$

例 A.1.0.10. 在范畴 **Vect** 中，同构是一个可逆矩阵。例如，矩阵

$$A = \begin{bmatrix} 3 & 2 \\ 1 & 1 \end{bmatrix}$$

可逆，其逆映射为

$$A^{-1} = \begin{bmatrix} 1 & -2 \\ -1 & 3 \end{bmatrix}$$

因为

$$A^{-1}A = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = AA^{-1}$$

Exa例 A.1.0.11. 在范畴 **Div** 中，唯一的同构是恒等态射 s

$$\mathsf{id}_a = \mathsf{div}_a^a.$$

这是因为，如果 a 整除 b，且 b 整除 a，那么 $a = b$。

A.2 什么是函子?

接下来，我们来关注第二个问题。

定义 A.2.0.1。 设**C** 和**D** 是范畴。一个函子 F :**C** $\to$**D** 由以下部分组成:

1. 对于**C**中的每个对象 X，指定**D**中的一个对象 FX，并且

2. 对于**C**中的每个态射 $f : X \to Y$，指定一个态射

$$Ff : FX \to FY$$

在**D**中,

满足以下性质:

1. F 与复合相容: 如果**C**中 f 和 g 是可复合态射，那么

$$F(g \circ f) = Fg \circ Ff.$$

2. F 与恒等态射相容: 如果 X 是 **C** 中的一个对象，那么

$$F\mathsf{id}_X = \mathsf{id}_{FX}.$$

例 A.2.0.2。 存在一个函子 F: **Div** $\to$**Set**，定义如下。在对象上， F 由下式给出

$$Fa = [0, a] = \{x \in \mathbb{R} \mid 0 \le x \le a\}.$$

在态射上， F 由下式给出

$$F\mathsf{div}_a^b(x) = \tfrac{b}{a} \cdot x.$$

让我们验证 F 是一个函子。

1. F 与复合相容: 如果 a 整除 b且 b整除 c，那么

$$(F\mathsf{div}_b^c \circ F\mathsf{div}_a^b)(x) = F\mathsf{div}_b^c(F\mathsf{div}_a^b(x)) = \frac{c}{b} \cdot (\frac{b}{a} \cdot x) = \frac{c}{a} \cdot x = F\mathsf{div}_a^c(x).$$

2. F 与恒等态射兼容: 如果 $a \ge 1$，则

$$F\mathsf{id}_a(x) = F\mathsf{div}_a^a(x) = \frac{a}{a} \cdot x = \mathsf{id}_{Fa}(x).$$

参考文献

[1] Somashekaracharya G. Bhaskaracharya 等。《使用整数集关系建模布局抽象》。2025年。arXiv：2511.10374。网址： https://arxiv.org/abs/2511.10374。[2] Uday Bondhugula 等。《实用的自动多面体并行化器与局部性优化器》。载于：第 29 届 ACM SIGPLAN 编程语言设计与实现会议论文集。PLDI '08。ACM，2008年。doi： 10.1145/1375581.1375595。[3] Cris Cecka、Vijay Thakkar 和 Tejash Shah。《CUTLASS：通过张量和空间微内核处理多维数据的原理化抽象》。NVIDIA 技术博客。2025年。网址： https://developer.nvidia.com/blog/cutlass-principled-abstractions-for-handling-multidimensional-data-through-tensors-and-spatial-microkernels/。[4] Zhaodong Chen 等。《EVT：使用 EpilogueVisitor Tree 加速深度学习训练》。载于：第 29 届 ACM 编程语言与操作系统体系结构支持国际会议（ASPLOS）论文集，第 3 卷。ACM，2024年，第 301–316 页。doi：10.1145/3620666.3651369。[5] NVIDIA 公司。CuTe — NVIDIA CuTe 文档。2024年。网址： https://docs.nvidia.com/cutlass/media/docs/cpp/cute/index.html。[6] NVIDIA 公司。CuTe DSL — NVIDIA CUTLASS 文档。2025年。网址： https://docs.nvidia.com/cutlass/media/docs/pythonDSL/cute_dsl.html。[7] Tri Dao。《FlashAttention-2：通过更好的注意力机制并行性与工作划分实现更快的注意力计算》。载于：国际学习表征会议（ICLR）。2024年。[8] Oliver J. Sharp、David F. Bacon、Susan L. Graham。《面向高性能计算的编译器变换》。载于：ACM 计算调查 26.4（1994年12月），第 345–420 页。doi：10.1145/197405.197406。[9] Tobias Grosser、Armin Gr¨oßlinger 和 Christian Lengauer。《Polly——在低级中间表示上执行多面体优化》。载于：并行处理快报 22.4（2012年）。[10] Wentao Guo 等。SonicMoE：通过输入输出和感知分块的优化加速混合专家。2025年。arXiv：2512.14080 [cs.LG]。网址： https://arxiv.org/abs/2512.14080。[11] Yong-Jae Ju 和 Henry Dietz。《通过编译器数据布局和循环变换降低缓存一致性开销》。载于：并行计算语言与编译器。Utpal Banerjee 等编。柏林、海德堡：施普林格，1992年。doi： 10.1007/BFb0038675。[12] Tamara G. Kolda 和 Brett W. Bader。《张量分解及其应用》。载于：SIAM 评论 51.3（2009年），第 455–500 页。doi： 10.1137/07070111X。网址： https://doi.org/10.1137/07070111X。

[13] Saunders Mac Lane。《工作数学家的范畴论》。第2版。第5卷。数学研究生教材。施普林格，1998年。isbn：978-0-387-98403-0。[14] OpenAI。《Triton 中的线性布局》。2025年。网址：https://github.com/triton-lang/triton/blob/main/include/triton/Tools/LinearLayout.h。[15] Easwaran Raman、Robert Hundt 和 Sandya Mannarswamy。《多线程程序的结构布局优化》。载于：国际代码生成与优化研讨会论文集。CGO '07。美国华盛顿特区：IEEE 计算机学会，2007年，第271–282页。doi：10.1109/CGO.2007.18。[16] Jay Shah。《CuTe 布局代数札记》。技术报告。Colfax Research，2024年1月。网址：https://research.colfax-intl.com/wp-content/uploads/2024/01/layout_algebra.pdf。[17] Jay Shah、Paul VanKoughnett 和 Ryo Asai。《CUTLASS 教程：在 NVIDIA Blackwell GPU 上使用线程块簇进行通用矩阵乘法》。Colfax Research。2025年。网址：https://research.colfax-intl.com/cutlass-tutorial-gemm-with-thread-block-clusters-on-nvidia-blackwell-gpus/。[18] Jay Shah、Paul VanKoughnett 和 Ryo Asai。《CUTLASS 教程：持久化内核与 Stream-K》。Colfax Research。2024年。网址：https://research.colfax-intl.com/cutlass-tutorial-persistent-kernels-and-stream-k/。[19] Jay Shah、Paul VanKoughnett 和 Ryo Asai。《CUTLASS 教程：在 NVIDIA Blackwell GPU 上进行子字节通用矩阵乘法》。Colfax Research。2025年。网址：https://research.colfax-intl.com/cutlass-tutorial-sub-byte-gemm-on-nvidia-blackwell-gpus/。[20] Jay Shah、Paul VanKoughnett 和 Ryo Asai。《CUTLASS 教程：使用张量内存为 NVIDIA Blackwell GPU 编写通用矩阵乘法内核》。Colfax Research。2025年。网址：https://research.colfax-intl.com/cutlass-tutorial-writing-gemm-kernels-using-tensor-memory-for-nvidia-blackwell-gpus/。[21] Jay Shah 等。《FlashAttention-3：以异步和低精度实现快速、准确的注意力》。载于：神经信息处理系统进展（神经信息处理系统大会）。2024年。[22] Kamal Sharma 等。《面向可移植性能的数据布局优化》。载于：欧洲并行处理会议。Euro-Par 2015。柏林，海德堡：施普林格，2015年。doi：10.1007/978-3-662-48096-0_20。[23] Yang Shi 等。《在中央处理器和图形处理器上使用扩展 BLAS 内核进行张量缩并》。载于：arXiv 预印本（2016年）。电子版：arXiv:1606.05696。网址：https://arxiv.org/abs/1606.05696。[24] Brandon Sun 等。使用 CuTe DSL 的 Python API 实现 CUTLASS C++ 性能。NVIDIA 技术博客。2025年11月。网址：https://developer.nvidia.com/blog/achieve-cutlass-c-performance-with-python-apis-using-cute-dsl/。[25] Vijay Thakkar 等。CUTLASS 3.x：面向通用矩阵乘法内核设计的正交、可复用且可组合的抽象。NVIDIA 技术博客。2025年。网址：https://developer.nvidia.com/blog/cutlass-3-x-orthogonal-reusable-and-composable-abstractions-for-gemm-kernel-design/。[26] Arun Thangamani、Vincent Loechner 和 Stéphane Genaud。《通用多面体编译器综述》。载于：ACM 架构与代码优化汇刊 21.4（2024年），第1–26页。doi：10.1145/3674735。[27] Nicolas Vasilache 等。《Tensor Comprehensions：框架无关的高性能机器学习抽象》。载于：arXiv 预印本 arXiv:1802.04730（2018年）。Facebook 人工智能研究院技术报告。

[28] Sven Verdoolaege。《isl：多面体模型的整数集库》。载于：数学软件 – *ICMS 2010*。Komei Fukuda 等编。第6327卷。计算机科学讲义。柏林，海德堡：施普林格，2010年。DOI：10.1007/978-3-642-15582-6_49。[29] Sven Verdoolaege。*Presburger*公式与多面体编译。技术报告／教程。版本 v0.02-13-g53eb23d，2021年8月21日。Polly Labs 和鲁汶大学，2021年。[30] Zhiying Xu 等。《ALT：打破深度学习编译中数据布局与循环优化之间的壁垒》。载于：第十八届欧洲计算机系统会议论文集。EuroSys '23。美国纽约：计算机协会，2023年。DOI：10.1145/3552326.3587440。[31] Tuowen Zhao 等。《稀疏张量收缩的多面体规范与代码生成：协同迭代》。载于：*ACM* 架构与代码优化汇刊 20.1（2022年12月）。第16号文章，第1–26页。DOI：10.1145/3566054。[32] Keren Zhou 等。《线性布局：使用 F2 高效张量计算的稳健代码生成》。载于：第*31*届 *ACM* 编程语言与操作系统架构支持国际会议论文集，第*1*卷。ASPLOS '26。美国宾夕法尼亚州匹兹堡：ACM，2026年3月，第1–18页。DOI： 10.1145/3760250.3762221。