arXiv:2602.17917v1 [math.CT] 20 Feb 2026

重塑交互各方接口的交互

David I. Spivak*

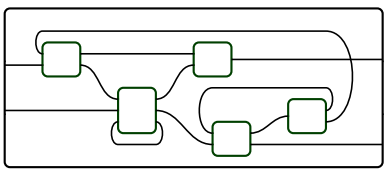
摘要

多项式函子用于建模具有接口的系统：每个多项式都指定系统能够产生的输出，以及对于每个输出，它所接受的输入。动态组织的双范畴 $\mathbb{O}\mathbf{rg}$动态组织 [Spi21]给出了一种由状态驱动、随时间演化的交互模式概念，但每个系统的接口在整个交互过程中保持固定。然而，在许多系统中，发送的输出和接收的输入都可能重塑接口本身：细胞会响应化学信号分化，从而获得或失去受体；传感器会因输入而受损，进而失去一个通道；神经网络则可能在训练过程中提高其输出分辨率。

本文引入多项式树，即终端 $(u \triangleleft u)$-余代数的元素，其中 u 是宇宙多项式，用以建模此类系统：多项式树是一种余归纳树，其节点携带多项式，并且每一轮交互——选择一个输出并接收一个输入——都会决定一个子树，进而决定下一个接口。我们构造了一个多项式树的幺半闭范畴 **PolyTr**，其中包含以余归纳方式定义的态射、张量积和内部 hom。随后，我们构造一个双范畴 $\mathbb{O}\mathbf{rgTr}$，将 $\mathbb{O}\mathbf{rg}$推广；其 Hom-范畴通过带有余归纳作用与更新数据的状态集对态射进行参数化。我们利用常值树构造局部完全忠实函子 $\mathbb{O}\mathbf{rg} \to \mathbb{O}\mathbf{rgTr}$；这类树的接口不会随时间改变。最后，我们通过提出渐进式生成对抗网络的概念来说明这一推广：在这种网络中，梯度反馈决定何时将图像生成接口扩展到更高分辨率。

1 引言

接线图是表示信息流的常见方式，例如控制论中的框图 [BE15；BSZ14]，流行病学中的房室模型 [Lib+22]，以及许多其他用于组合系统的图形形式体系 [Sel10；Spi13；VSL15]。



*本材料基于美国空军科学研究办公室资助的研究，资助编号为 FA9550-23-1-0376。

每个盒子都有一个接口——由输入和输出端口组成——而布线则规定了一个盒子的输出如何流向另一个盒子的输入，以及整个结构如何在更高的抽象层次上实现一个盒子。

但与许多现实场景不同，这些接线图是静态的，而在许多现实场景中，交互模式是动态的。也就是说，在实体之间流动的事物会影响其流经的通道：水流会为自己冲出更深的河道；一个“断开”信号会使承载信号的导线断开。在训练人工神经网络时，模型中层层传递的信息转换的准确率会引发变化，改变构成该转换本身的连接强度（权重）。

这种现象——交互模式会根据流经其中的事物而重新配置——已被形式化为 动态范畴结构：在双范畴 $\mathbb{O}\mathbf{rg}$ 的动态组织 [SS22]中丰富化的范畴、运算子和幺半范畴 $p \in \mathbf{Poly}$ （一个函子$\sum_{i:\, I} y^{A_i}$： $\mathbf{Set} \to \mathbf{Set}$，其中 I 是位置 的集合，而 A_i 是每个位置处的方向 的集合i；见第 2 节），并且盒子之间的布线由一个状态随时间演化的余代数控制。其应用包括预测市场（交易者的成功会重新配置财富分布）以及梯度下降（预测误差会重新配置模型的权重参数）。

然而，在许多值得关注的情形中，发生变化的不只是布线，接口本身也会改变。神经元可能死亡；传感器可能受损，失去一个输入通道；生物体可能获得假肢，从而具备新的能力。在所有这些情形中，流经系统的信号会改变它们所流经的盒子的形状本身。这些现象都无法用 [SS22], 的框架来描述，因为在该框架中，每个盒子始终保持其固定的接口。

本文构建了一个更一般的范畴框架，以涵盖这一现象。我们用多项式树取代每个固定的多项式：这是一种余归纳树，其中每个节点都带有一个多项式，并且该多项式的每个位置和方向都对应一个分支。这样的树编码了一个不断演变的接口：每经过一轮交互，接口都可能转变为另一个多项式，而这一过程以余归纳方式持续进行。具体而言，接口可能消亡（转变为零多项式）、分裂，或发生任何其他结构变化；这些变化完全取决于交互过程中流入和流出该接口的内容。

技术核心是一个宇宙多项式$\mathrm{u} \coloneqq \sum_{I \in \mathbb{U}} \mathrm{y}^{|I|}$，其中 $\mathbb{U}$ 是一个固定的基数宇宙（例如有限基数）。它的自代入 $\mathrm{u} \triangleleft \mathrm{u}$ 是一个多项式，其位置编码了元数来自 $\mathbb{U}$ 的多项式：在 $\mathrm{p} : \mathrm{u} \triangleleft \mathrm{u}(1)$ 中，一个位置由一个 $\mathbb{U}$-位置集以及与每个位置对应的一个 $\mathbb{U}$-方向集组成。多项式树则是终端 $(\mathrm{u} \triangleleft \mathrm{u})$-余代数的元素，也就是说，是每个节点都携带此类编码多项式的余归纳树。

我们的主要构造是一个范畴 $\mathbf{PolyTr}_{\mathbb{U}}$，其对象是多项式树，其态射则是余归纳定义的多项式映射系统；沿着每条可能的交互路径，在每个节点处都有一个这样的映射。我们证明 $\mathbf{PolyTr}_{\mathbb{U}}$ 具有：

- 一个对称幺半积 $\otimes$，将接口的并行复合扩展到演化设定中；

- 一个内部 hom $[-,-]$，使 **PolyTr**$_{\mathbb{U}}$ 成为幺半闭结构；
- 一个余积 + 由 $\otimes$分配。

多项式的全子范畴 $\mathbf{Poly}_{\mathbb{U}} \subseteq \mathbf{Poly}$，其元数取自 $\mathbb{U}$，可通过常值树忠实嵌入到**PolyTr**$_{\mathbb{U}}$中；常值树会在每个节点重复同一个多项式。我们还发展了**PolyTr**$_{\mathbb{U}}$的一个**Cat**-充实版本，其中态射 $p \to q$ 由 $\mathbb{U}$ 中的一个状态集，以及一个由多项式映射值作用和状态更新构成的余归纳系统指定。由此得到的双范畴 $\mathbb{O}$**rgTr** 推广了 $\mathbb{O}$**rg**：对于常值多项式树，**rg** $\mathbb{O}$ 的 hom 范畴可全忠实地嵌入**rgTr** $\mathbb{O}$的 Hom-范畴中。在**rgTr** $\mathbb{O}$中的充实产生了动态范畴、运算子和幺半范畴，其中布线和 接口都协同演化。

论文结构。 第 2 节回顾多项式函子及其幺半结构。第 3 节将多项式树定义为余自由 $(u \triangleleft u)$-余代数的元素，并构造范畴 **PolyTr**$_{\mathbb{U}}$，其态射、张量积和幺半闭结构均以余归纳方式定义。第4 节研究 **Poly** 与 **PolyTr**$_{\mathbb{U}}$之间的关系，构造推广 $\mathbb{O}$**rg**的双范畴 $\mathbb{O}$**rgTr**，并通过一个渐进式深度学习的例子加以说明。

2 预备知识

我们简要回顾范畴 **Poly**中的一元多项式函子（定义在 **Set**上）以及一些相关结构；参见 [NS24] 的全面论述，以及 [Spi22] 的简明参考。

2.1 多项式函子及其映射

正如引言所述，一个 多项式函子 是自函子 $p\colon \mathbf{Set} \to \mathbf{Set}$，形如$\sum_{i:I} \mathrm{y}^{A_i}$，其中 I 是位置集， A_i 是位置 i处的方向集。由于$p(1) = \sum_{i:I} 1 = I$，我们可以避免另设变量，而采用紧凑记号 $p(1)$ 表示位置集，并用 $p[i] \coloneqq A_i$ 表示 i 处的方向集： $p(1)$，从而

$$p = \sum_{i:p(1)} y^{p[i]} = \sum_{i:p(1)} \prod_{d:p[i]} y. \tag{1}$$

正如中学代数那样，我们用并置表示数乘： $pq \coloneqq p \times q$，并记 $1 = \mathrm{y}^0$。一个 单项式 是形如 $M\mathrm{y}^N = \sum_{m:M} \mathrm{y}^N$ 的多项式；我们将 y^N 称为可表对象，并将 $M = M\mathrm{y}^0$ 称为 常数。

态射 $\varphi\colon p \to q$ 在 **Poly** 中（即对应函子的自然变换）等价于一个在位置上正向、在方向上反向的映射：

$$\mathbf{Poly}(p, q) = \prod_{i:p(1)} \sum_{j:q(1)} \prod_{e:q[j]} \sum_{d:p[i]} 1. \tag{2}$$

根据类型论选择公理，[1] 外层的 $\prod_i \sum_j$ 给出一个函数 $\varphi_1 \colon\ p(1) \to q(1)$，而内层的 $\prod_e \sum_d$ 则针对每个 $i \colon\ p(1)$，给出一个函数 $\varphi_i^\# \colon\ q[\varphi_1(i)] \to p[i]$。下标 φ_1 记录自然变换 φ 在 $1 : \mathsf{Set}$ 处的分量。若每个 $\varphi_i^\#$ 都是同构，我们就称 φ是笛卡尔的。[2]

2.2 Poly上的幺半结构

在本节中，我们回顾**Poly** 上的幺半结构，这些结构将被提升到$\mathbf{PolyTr}_{\mathbb{U}}$。

范畴**Poly** 具有所有余积与积：对任意一族多项式$(p_\alpha)_{\alpha:A}$，

$$\Big(\sum_{\alpha:A} p_\alpha\Big)(1) = \sum_{\alpha:A} p_\alpha(1), \qquad \Big(\prod_{\alpha:A} p_\alpha\Big)(1) = \prod_{\alpha:A} p_\alpha(1) \tag{3}$$

其方向为$(\sum_{\alpha:A} p_\alpha)[(\alpha, i)] = p_\alpha[i]$ 以及$(\prod_{\alpha:A} p_\alpha)[(i_\alpha)_{\alpha:A}] = \sum_{\alpha:A} p_\alpha[i_\alpha]$。始多项式0没有位置，终多项式 $1 = \mathsf{y}^0$ 有一个位置且没有方向。在二元情形下，我们记作 $p + q$ 与 $p \times q$。

除和与积之外，**Poly** 还具有另外两个与本论文相关的幺半结构。

Diric **积与闭性**。对于多项式 $p,\ q$，它们的 狄利克雷张量积 是

$$(p \otimes q)(1) \coloneqq p(1) \times q(1), \qquad (p \otimes q)[(i,j)] \coloneqq p[i] \times q[j]. \tag{4}$$

单位是 $\mathsf{y} = \mathsf{y}^1$。$\times$ 和 $\otimes$ 都对 $+$分配。狄利克雷积是对称幺半且闭的：其内部 hom 为

$$[p,q] = \sum_{\varphi:\mathbf{Poly}(p,q)} y^{\sum_{i:p(1)} q[\varphi_1(i)]} = \prod_{i:p(1)} \sum_{j:q(1)} \prod_{e:q[j]} \sum_{d:p[i]} y \tag{5}$$

满足 $\mathbf{Poly}(r \otimes p,\ q) \cong \mathbf{Poly}(r,\ [p,q])$，且关于 r 自然。

注意，$[p,q](1) = \mathrm{Poly}(p,q)$，这从比较 (5) 和 (2) 可立即得出。$\varphi$ 处的一个方向：$[p,q](1)$ 由一对 (i,e) 构成，其中 $i \colon\ p(1)$ 是一个 p-位置，而$e \colon\ q[\varphi_1(i)]$ 是一个 q-方向。但请注意，这些数据也隐含地指定了两个多项式的位置和方向：我们有一个 q-位置 $j \coloneqq \varphi_1(i)$，并且还有一个p-方向 $d \coloneqq \varphi_i^\sharp(e)$。

注记 2.1. 布线图可用于表示多项式映射，即从若干“内部”盒子的张量积到单个“外部”盒子。形如y^X 的每个单项式是

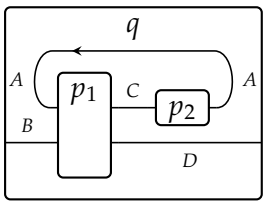
[1]类型论选择公理指出，积对和具有分配性，

$$\prod_{a:A} \sum_{b:B(a)} C(a,b) \cong \sum_{b:\prod_{a:A} B(a)} \prod_{a:A} C(a, b(a)).$$

也就是说，依赖和的依赖积等价于原依赖积在选择函数上的和。与集合论选择公理不同，这在类型论中是可证的；参见[**Voevodsky:2013a**]。

[2]术语笛卡尔的 出奇地稳健。若映射 φ 的自然性方块都是拉回方块，则称该映射为笛卡尔的；当且仅当 φ 对应于连接相关丛的一个拉回方块时，亦当且仅当 φ 在格罗滕迪克纤维化$\mathbf{Poly} \to \mathbf{Set}$ （发送 $p \mapsto p(1)$）的意义下是笛卡尔的。

画成一个盒子x w，它在 W 中输出值，并在 X中输入值，其中W, $X : \mathbf{Set}$。那么，像下面这样的布线图

$\varphi :=$ 

表示一个映射 $\varphi\colon\ p_1 \otimes p_2 \to q$，其中 $p_1 = CD\mathrm{y}^{AB}$、$p_2 = A\mathrm{y}^{C}$，且 $q = D\mathrm{y}^{B}$。事实上，要定义一个映射

$$CDAy^{ABC} \cong p_1 \otimes p_2 \xrightarrow{\varphi} q = Dy^{B}$$

需要一个函数 $\varphi_1\colon\ CDA \to D$ 以及一个函数 $\varphi^\sharp\colon\ CDAB \to ABC$；我们对每个函数都使用投影。[3] 在第 3节中，我们将把这个思路推广到多项式树，其中盒子 p_1、p_2、q 的接口可能在每一轮交互后发生变化——这取决于流经导线的内容（即 $CDAB$的元素）——交互模式 φ也必须相应地调整。◊

替换积。替换积（或复合）积定义为 $(p \triangleleft q)(S) := p(q(S))$，其单位为 y。利用式（1）的展开形式：

$$p \triangleleft q = \sum_{i:p(1)} \prod_{d:p[i]} \sum_{j:q(1)} \prod_{e:q[j]} y. \tag{6}$$

因此，$p \triangleleft q$ 的位置是一个有序对 $(i,\ j)$，其中 $i :\ p(1)$，并且根据类型论选择公理，$j\colon\ p[i] \to q(1)$，将一个 q- 位置 $j_d :\ q(1)$ 指派给 i 处的每个方向 $d :\ p[i]$。方向集为 $(p \triangleleft q)[(i,\ j)] = \sum_{d:p[i]} q[j_d]$，因此一个方向是方向的有序对 $(d,\ e)$，$d :\ p[i]$且 $e :\ q[j_d]$。

2.3 余自由余单子

本节回顾余自由余单子的构造，它将为多项式树提供余归纳框架。

A $\triangleleft$-余幺半群 是**Poly** 中的一个余幺半群，其对应于替换积：一个多项式 c配备余单位 $\epsilon\colon\ c \to \mathrm{y}$ 和余乘法 $\delta\colon\ c \to c \triangleleft c$，并满足余结合律和余单位性。由于替换积是**Set**上的自函子的复合，$\triangleleft$-余幺半群等价于**Set**上的（多项式）余单子。令**Cmd(Poly)**表示由 $\triangleleft$-余幺半群及其态射构成的范畴。

命题 2.2 ([LS24])。 遗忘函子 $U\colon$ **Cmd**(**Poly**) $\to$ **Poly** 存在一个右伴随c: **Poly** $\to$ **Cmd**(**Poly**)，

$$\mathbf{Cmd}(\mathbf{Poly}) \underset{c}{\overset{U}{\rightleftarrows}} \mathbf{Poly}$$

[3]接线图表示单项式之间仅包含投影和对角映射的多项式映射W^X。

称为余自由余单子。对于任意多项式 p，其构造如下。定义多项式 $p^{(i)}$，其中 $i : \mathbb{N}$，使得

$$p^{(0)} \coloneqq y \qquad \text{and} \qquad p^{(1+i)} \coloneqq y \times (p \triangleleft p^{(i)}) \tag{7}$$

带有投影映射 $\pi^{(i)} : \quad p^{(i+1)} \to p^{(i)}$，其中 $\pi^{(0)}$ 是终端映射，且 $\pi^{(i+1)} \coloneqq y \times (p \triangleleft \pi^{(i)})$ 。于是 $\mathfrak{c}_p \coloneqq \lim_i p^{(i)}$。

因此，位置集 $\mathfrak{c}_p(1)$ 由 p-行为树组成：余归纳地，一个 p-行为树 t 具有一个根位置 $t_{\text{。root}} : p(1)$，以及一个子树函数 $t_{\text{。rest}} : p[t_{\text{。root}}] \to \mathfrak{c}_p(1)$，它将子树 $t_{\text{。rest}_d}$ 分配给每个方向 $d : p[t_{\text{。root}}]$。方向集 $\mathfrak{c}_p[t] = 1 + \sum_{d : p[t_{\text{。root}}]} \mathfrak{c}_p[t_{\text{。rest}_d}]$ 是 t 的节点集：要么是根节点，要么是一个根方向 $d : p[t_{\text{。root}}]$，后面接上 $t_{\text{。rest}_d}$ 中的一个节点。

回顾一下，对于自函子 $p : \mathrm{Set} \to \mathrm{Set}$，其余代数是一个集合 S，并配备一个结构映射 $\beta : S \to p(S)$。$\mathfrak{c}_p(1)$—— p-行为树的集合——是终端 p-余代数：对于任意 p-余代数 (S, β)，都存在唯一映射 $S \to \mathfrak{c}_p(1)$，将每个状态映射到其余归纳行为树。该泛性质将在第 3 节及后续章节中使用。

2.4 宇宙多项式

固定一个宇宙 $\mathbb{U}$，其中的基数在单元素集、依赖和与依赖积下封闭；也就是说，这是 Martin-Löf 类型论意义下的 ΠΣ-宇宙 [Mar75]。任意 Grothendieck 宇宙都给出这样一个 $\mathbb{U}$，有限基数构成的宇宙也是如此。对于每个基数 $I \in \mathbb{U}$，记 $|I|$ 为该基数的一个典范集合。定义宇宙多项式

$$\mathrm{u}_{\mathbb{U}} \coloneqq \sum_{I \in \mathbb{U}} y^{|I|} \tag{8}$$

使得 $\mathrm{u}(1) = \mathbb{U}$ 且 $\mathrm{u}[I] = |I|$。例如，当 $\mathbb{U}$ 是有限基数构成的宇宙时，$\mathrm{u}_{\mathbb{U}}$ 是列表多项式 $\mathrm{列表} \coloneqq \sum_{N : \mathbb{N}} \mathrm{y}^N$。通常我们会省略下标，直接写作 u。

由于 $\mathbb{U}$ 在单元素集和依赖和下封闭，u 在 $(\mathbf{Poly}, \mathrm{y}, \triangleleft)$ 中具有笛卡尔单子的结构：单位 $\eta : \mathrm{y} \to \mathrm{u}$ 选出单元素集，而乘法 $\mu : \mathrm{u} \triangleleft \mathrm{u} \to \mathrm{u}$ 将一个由 $\mathbb{U}$-个 $\mathbb{U}$-集合组成的集合映射为它们的依赖和；笛卡尔性保证了这一点（见第 2.1 节末）。在有限情形下，这两者分别是单元素列表和列表拼接。例如，μ 将自然数列表 $(N_1, \ldots, N_K)$ 映射为它们的和 $N_1 + \cdots + N_K$，并在方向上给出典范双射。从第 3 节开始将用到依赖积下的封闭性。

复合 $\mathrm{u} \triangleleft \mathrm{u}$ 将是本文的核心：它的位置编码了元数来自 $\mathbb{U}$ 的多项式。根据 (6)，位置 $\mathrm{w} = (I, D) : (\mathrm{u} \triangleleft \mathrm{u})(1)$ 由 $I : \mathrm{u}(1)$ 和 $D : \mathrm{u}[I] \to \mathrm{u}(1)$ 组成，也就是说，一个基数 I 以及一个由 $\mathbb{U}$ 索引的基数组。这些数据编码了多项式

$$[\![\mathrm{w}]\!] \coloneqq \sum_{i : |I|} y^{|D_i|} : \mathbf{Poly}. \tag{9}$$

方向集 $(\mathrm{u} \triangleleft \mathrm{u})[\mathrm{w}] = \sum_{i : \mathrm{u}[I]} \mathrm{u}[D_i] = \sum_{i : |I|} |D_i|$（同样根据 (6)）由位置-方向对组成 (i, d) 属于 $[\![\mathrm{w}]\!]$，其中 $i : |I|$ 且 $d : |D_i|$。

我们称多项式 $p = \sum_{i:p(1)} \mathrm{y}^{p[i]} : \mathrm{Poly}$ 的元数来自 $\mathbb{U}$，如果 $p(1) = |I|$ 且 $p[i] = |D_i|$，其中 $I, D_i : \mathrm{u}(1)$ 是（必然唯一的）元素，并记 $\mathrm{Poly}_{\mathbb{U}} \subseteq \mathrm{Poly}$ 为此类多项式构成的全子范畴。对于任意 $p : \mathrm{Poly}_{\mathbb{U}}$，定义其编码 $\ulcorner p \urcorner := (I, D) : (\mathrm{u} \triangleleft \mathrm{u})(1)$，其中 I 和 D 是满足 $|I| = p(1)$ 和 $|D_i| = p[i]$ 的唯一元素。于是 $[\![\cdot]\!]$ 和 $\ulcorner \cdot \urcorner$ 是 $(\mathrm{u} \triangleleft \mathrm{u})(1)$ 与 $\mathrm{Ob}(\mathrm{Poly}_{\mathbb{U}})$ 之间的互逆双射：$[\![\ulcorner p \urcorner]\!] = p$ 以及 $\ulcorner [\![(I, D)]\!] \urcorner = (I, D)$。每个位置集与方向集的基数由 $\mathbb{U}$ 表示的多项式，都（非典范地）同构于 $\mathrm{Poly}_{\mathbb{U}}$ 中的一个多项式。

3 范畴 $\mathbf{PolyTr}_{\mathbb{U}}$：多项式树

现在，我们定义多项式树——由演化中的多项式接口构成的余归纳树——并将其作为范畴 $\mathbf{PolyTr}_{\mathbb{U}}$ 的对象。

3.1 多项式树

本节将多项式树定义为终端 $(\mathrm{u} \triangleleft \mathrm{u})$-余代数的元素 $\mathrm{p} : \mathfrak{c}_{\mathrm{u} \triangleleft \mathrm{u}}(1)$，并描述其余归纳结构。

根据（6），位置 $\mathrm{w} : (\mathrm{u} \triangleleft \mathrm{u})(1)$ 具有两个分量：$\mathrm{w.pos} : \mathrm{u}(1)$ 和 $\mathrm{w.dir} : \mathrm{u}[\mathrm{w.pos}] \to \mathrm{u}(1)$，一个位置集 $|\mathrm{w.pos}|$：**Set** 以及一个方向赋值。它们共同编码了多项式 $[\![\mathrm{w}]\!] = \sum_{i : |\mathrm{w.pos}|} \mathrm{y}^{|\mathrm{w.dir}_i|} : \mathbf{Poly}$，参见（9）；其方向集为 $(\mathrm{u} \triangleleft \mathrm{u})[(\mathrm{w.pos}, \mathrm{w.dir})] = \sum_{i : |\mathrm{w.pos}|} |\mathrm{w.dir}_i|$，即 $[\![\mathrm{w}]\!]$ 的所有方向。

定义 3.1。一个多项式树是以下元素 $\mathrm{p} : \mathfrak{c}_{\mathrm{u} \triangleleft \mathrm{u}}(1)$，终端 $(\mathrm{u} \triangleleft \mathrm{u})$-余代数（第2.3节）。余归纳地，p由以下部分组成：

- 一个根编码 $\mathrm{p}_{。根} = (\mathrm{p}_{。位置}, \mathrm{p}_{。方向}) : (\mathrm{u} \triangleleft \mathrm{u})(1)$，其中 $\mathrm{p}_{。位置} : \mathrm{u}(1)$，且 $\mathrm{p}_{。方向} : \mathrm{u}[\mathrm{p}_{。位置}] \to \mathrm{u}(1)$，用于编码根多项式 $[\![\mathrm{p}_{。根}]\!] = \sum_{i : |\mathrm{p}_{。位置}|} \mathrm{y}^{|\mathrm{p}_{。方向}{}^{i}|} : \mathbf{Poly}$；以及
- $\mathrm{p}_{。余项} : (\mathrm{u} \triangleleft \mathrm{u})[\mathrm{p}_{。根}] \to \mathfrak{c}_{\mathrm{u} \triangleleft \mathrm{u}}(1)$，为每个方向 $(i, d) : (\mathrm{u} \triangleleft \mathrm{u})[\mathrm{p}_{。根}]$ 分配一棵子树 $\mathrm{p}_{。余项 i, d} : 。\lozenge$

多项式树编码了一个在交互过程中不断演变的接口。根多项式 $[\![\mathrm{p}_{。根}]\!] : \mathbf{Poly}$ 是当前接口；当选取位置 $i : \mathrm{p}_{。位置}$ 并收到方向 $d : |\mathrm{p}_{。方向 i}|$ 时，接口转变为 $[\![\mathrm{p}_{。余项 i, d。根}]\!]$，这一过程继续进行。在 $\mathrm{p}_{。根} = N\mathrm{y}^0$ 为常量的任何分支上，这棵树都会终止。

例 3.2（通信协议）。多项式树可以指定一种通信协议，其可用操作会随时间变化。在每个节点上，位置表示客户端可以发送的消息类型，方向表示服务器可能的响应。

例如，考虑一台服务器，初始时提供登录（响应为成功或失败）和退出（无响应）：根多项式为 $\mathrm{y}^2 + 1$。登录/成功后，接口转换为字符串 $\mathrm{y}^{\mathrm{String}}$ +字符串 $\mathrm{y}^{\mathrm{Int}} + 1$：已认证会话提供查询（发送查询字符串并返回结果字符串）、Set（发送字符串 并返回整数 状态编码的写操作）以及注销（终止会话）。之后

登录/失败，它会返回到y^2+1以便重试。多项式树编码了完整协议，余归纳结构则捕捉了无界交互。这一视角与会话类型 [Hon93]。◊

例 3.3（细胞分化）。细胞分化时，其接口会发生变化。设 L 为浓度级别的有限集合，且足够大，能够涵盖任何信号分子的所有相关浓度。对一个干细胞建模：它输出浓度为 $l : L$ 的生长因子，并同时在三个受体上接收信号，每个受体都能检测 L中的一种浓度。根多项式为 $Ly^{L^3} : \mathrm{Poly}$，其中每个 $x : L$都有 $|\mathrm{p.pos}| = L$ 和 $|\mathrm{p.dir}_x| = L^3$。方向 $d = (l_1,l_2,l_3) : L^3$ 是一个浓度三元组，每个受体对应一个浓度。子树取决于此输入：例如，高 l_1 可能触发 $[\![\mathrm{p.rest}_{x,d}.\mathrm{root}]\!] = Ly^L$（分化为输出神经递质的神经元，具有一个受体）；高 l_2 可能产生 $[\![\mathrm{p.rest}_{x,d}.\mathrm{root}]\!] = Ly^{L^3}$（生长信号，接口不变）；高 l_3 可能产生 $[\![\mathrm{p.rest}_{x,d}.\mathrm{root}]\!] = 0$（细胞凋亡）。神经元子树可能是 Ly^L上的常量；生长子树则可余归纳地等于 p 本身。◊

例3.4（国际象棋）。国际象棋自然可以用多项式树来建模。在每个棋盘状态 b下，当前玩家都有一个有限的合法走法集合 M_b；走出着法$m : M_b$后，对手则有一组合法应着 $R_{b,m}$。因此，状态 b 对应的多项式为$\sum_{m:M_b} y^{R_{b,m}}$，而 (m, r) 对应的子树则是该着法所产生棋盘状态的多项式树。每走一步，接口都可能发生相当大的变化：位于 $e2$的兵可以前进到 $e3$ 或 $e4$，但只有在对手棋子占据 $d3$ 或 $f3$ 时才能斜向吃子；只有当王和车都未移动过，且中间的格子均未受攻击时，才能王车易位；兵到达第八横线时必须升变，替换为一个新棋子，并改变后续所有局面的走法集合。国际象棋的多项式树编码了完整的博弈树，每个节点上的多项式都会随合法走法和应着集合的变化而变化。◊

最简单的多项式树是常值树：对于任意 $p : \mathrm{Poly}_{\mathbb{U}}$，常值树 $\bar{\mathrm{p}}$ 由余归纳地定义为 $\bar{\mathrm{p}}.\mathrm{root} :\equiv \ulcorner p \urcorner$ and $\bar{\mathrm{p}}.\mathrm{rest}_{i,d} :\equiv \bar{\mathrm{p}}$，对所有 i and d 都成立，因此$[\![$每个节点处都有 $\bar{\mathrm{p}}.\mathrm{root}]\!] = p$。常值树刻画了永不改变的接口；在第 4 节中，我们将证明 $p \mapsto \bar{\mathrm{p}}$ 扩展为一个忠实函子 $\mathrm{Poly}_{\mathbb{U}} \to \mathrm{PolyTr}_{\mathbb{U}}$。

3.2 多项式树范畴

本节将多项式树的态射集 $\mathbf{PolyTr}_{\mathbb{U}}(\mathrm{p},\mathrm{q})$ 定义为有限逼近的极限，从而使 $\mathbf{PolyTr}_{\mathbb{U}}$ 成为一个范畴，其复合由余归纳地定义。

定义 3.5. 对于多项式树 $\mathrm{p},\mathrm{q} : \mathfrak{c}_{u \lhd u}(1)$，对 $n : \mathbb{N}$ 定义集合 $\mathbf{PolyTr}^{(n)}_{\mathbb{U}}(\mathrm{p},\mathrm{q})$：$\mathbf{PolyTr}^{(0)}_{\mathbb{U}}(\mathrm{p},\mathrm{q}) :\equiv 1$ and

$$\mathbf{PolyTr}^{(n+1)}_{\mathbb{U}}(\mathrm{p},\mathrm{q}) :\equiv \prod_{i:|\mathrm{p.pos}|} \sum_{j:|\mathrm{q.pos}|} \prod_{e:|\mathrm{q.dir}_j|} \sum_{d:|\mathrm{p.dir}_i|} \mathbf{PolyTr}^{(n)}_{\mathbb{U}}(\mathrm{p.rest}_{i,d}, \mathrm{q.rest}_{j,e}). \quad (10)$$

这些构成一个集合的余滤图：存在唯一的函数 π_0: $\mathbf{PolyTr}^{(1)}_{\mathbb{U}}(\mathrm{p},\ \mathrm{q}) \to 1$并递归地定义投影为 π_{n+1}: $\mathbf{PolyTr}^{(n+2)}_{\mathbb{U}}(\mathrm{p},\ \mathrm{q}) \to \mathbf{PolyTr}^{(n+1)}_{\mathbb{U}}(\mathrm{p},\ \mathrm{q})$

由 $\prod_i \sum_j \prod_e \sum_d \pi_n$ 给出。态射集取为极限

$$\mathbf{PolyTr}_{\mathbb{U}}(\mathrm{p},\mathrm{q}) \coloneqq \lim_{n:\mathbb{N}} \mathbf{PolyTr}^{(n)}_{\mathbb{U}}(\mathrm{p},\mathrm{q}). \qquad \lozenge$$

记 $\mathrm{h} \coloneqq [\![\mathrm{p.root}]\!, [\![\mathrm{q.root}]\!]$：Poly 为根多项式的内部 hom；则 $\mathrm{h}(1) = \mathrm{Poly}([\![\mathrm{p.root}]\!], [\![\mathrm{q.root}]\!])$ 是它们之间的多项式映射的集合，而 $\mathrm{h}[\varphi] = \sum_i : |\mathrm{p.pos}|\, |\mathrm{q.dir}_{\varphi_1}(i)|$ 是映射 φ：$\mathrm{h}(1)$ 处的方向集。由于 $\mathrm{PolyTr}^{(0)}_{\mathbb{U}} = 1$，比较（10）与（2）可得 $\mathrm{PolyTr}^{(1)}_{\mathbb{U}}(\mathrm{p},\mathrm{q}) = \mathrm{h}(1) = \mathrm{Poly}([\![\mathrm{p.root}]\!], [\![\mathrm{q.root}]\!])$。

一般而言，元素 φ：$\mathbf{PolyTr}^{(n+1)}_{\mathbb{U}}(\mathrm{p}, \mathrm{q})$ 指定：

- 一个根映射 φ。root：$\mathbf{Poly}([\![\mathrm{p。root}]\!], [\![\mathrm{q。root}]\!])$；连同
- 对每个方向 $(i,\ e)$：$h[\varphi\text{。根}]$，设 $j \coloneqq \varphi_1(i)$ 和 $d \coloneqq \varphi\sharp_i(e)$，一个子树元素 φ。余部$_{i,\ e}$：$\mathbf{PolyTr}_{(n)\mathbb{U}}(\mathrm{p}\text{。余部}_{i,\ d},\ \mathrm{q}\text{。余部}_{j,\ e})$。

因此，态射 φ：$\mathbf{PolyTr}_{\mathbb{U}}(\mathrm{p},\ \mathrm{q})$ 由一个根映射 φ。根 : $h(1)$ 以及以下内容组成：对于 h 在 φ。根处的每个方向，都有一个连接相应子树的态射；沿着每条路径依此类推。

回顾一下，多项式映射 φ：$\mathrm{Poly}(p, q)$ 可以理解为一种交互模式，是对布线图（备注 2.1）的推广：φ 指定了输出和输入如何在 p 与 q 之间路由。态射 φ：$\mathrm{PolyTr}_{\mathbb{U}}(\mathrm{p}, \mathrm{q})$ 是由这类交互模式构成的一棵树：在根部，$\varphi.\mathrm{root}$ 将当前接口 $[\![\mathrm{p.root}]\!]$ 连接到 $[\![\mathrm{q.root}]\!]$；随后经过一轮交互，接口转变为新的多项式接口，而 φ 则提供适用于新接口的新交互模式。余归纳定义确保这些交互模式沿着每条交互路径都有定义。一般而言，映射 $\mathrm{p}_1 \otimes \cdots \otimes \mathrm{p}_k \to \mathrm{q}$ 可理解为使用接口 $\mathrm{p}_1, \ldots, \mathrm{p}_k$ 来实现接口 q 的一种方式，其中布线会随着接口的演变而调整。我们将在第 3.3 节定义多项式树上的 $\otimes$。

示例 3.6（协议细化）。考虑示例 3.2中的服务器协议 q，其根节点为 $\mathrm{y}^2 + 1$ (登录或退出)。一个只读 客户端协议 p 具有相同的初始根节点 $\mathrm{y}^2 + 1$，但在登录/成功后，它转移到 字符串$\mathrm{y}^{\mathrm{String}} + 1$ (仅包含查询和注销)，而 q 则转移到 字符串$\mathrm{y}^{\mathrm{String}}$ + 字符串$\mathrm{y}^{\mathrm{Int}} + 1$ (包含查询、设置和注销)。一个态射φ : $\mathbf{PolyTr}_{\mathbb{U}}(\mathrm{p},\mathrm{q})$将只读协议嵌入完整协议中。在初始根节点处，$\varphi$。根 = id。在已认证根节点处，$\varphi_1$ 将每个查询字符串 s : 字符串 映射到 q 中相同的查询字符串，并将注销 $\mapsto$ 映射到注销，且在每个节点上，方向上的映射都是 $\varphi^\sharp = \mathrm{id}$；q 中的设置直和项不在 φ_1的像中。登录/失败后，两棵树都返回到 $\mathrm{y}^2 + 1$ 并且 φ 重复；查询并收到响应 s : 字符串后，两棵树都返回各自的已认证根节点，且 φ 重复。只读协议在每个深度都包含于完整协议之中，只使用 q 的部分能力。$\lozenge$

例 3.7（**从 y 映射及映射到 y**）由于 y 的每个节点都只有一个位置和一个方向，从 y 映射及映射到 y 的映射具有一种特别简单的形式。

一个映射 $\mathrm{y} \to \mathrm{p}$ 余归纳地由一个位置 j：$|\mathrm{p.pos}|$ 组成；此外，对于每个方向 e：$|\mathrm{p.dir}_j|$，还要有一个映射 $\mathrm{y} \to \mathrm{p.rest}_{j,\ e}$：先选择一个输出，再处理每一种可能的输入。这样的映射存在，当且仅当 $[\![\mathrm{p.root}]\!]$ 具有非空的位置集；在所选位置沿方向可到达的所有子节点处也同样如此。

映射 $\mathrm{p} \to \mathrm{y}$ 余归纳地由以下部分组成：对于每个位置 i：$|\mathrm{p.pos}|$，一个方向d：$|\mathrm{p.dir}_i|$，以及一个映射 $\mathrm{p.rest}_{i,\ d} \to \mathrm{y}$：对于每个输出，选择一个输入来响应。

当且仅当 $[\![\mathrm{p.root}]\!]$ 的每个位置都有非空方向集，并且沿着所选方向可到达的所有子节点处也满足同样条件时，这样的映射才存在。

对于示例 3.4中的国际象棋树，映射 $w\colon \mathrm{y} \to \widehat{\mathrm{chess}}$ 是白方的不败策略：在每个棋盘状态下，选择一个合法着法并应对每一种响应。如果所选着法将死黑方（$R_{b,\,m} = \emptyset$），该分支便终止（空积为 1）。当且仅当黑方无法迫使局面进入白方无合法走法的状态时，这样的映射 w 才存在。对偶地，映射 $b\colon \mathrm{chess} \to \mathrm{y}$ 是黑方的不败反策略：对每个合法着法，选择一个响应。当且仅当没有可达的着法能够将死黑方时，这样的映射 b 才存在。[4] ◊

命题 3.8。 多项式树及其态射构成一个范畴 $\mathbf{PolyTr}_{\mathbb{U}}$。

证明。 我们在塔上定义复合和恒等，并取其极限。

恒等。 对于 $\mathrm{p}:\ \mathsf{y}^{\mathrm{u}\lhd\mathrm{u}}(1)$，通过 $(\mathrm{id}_{\mathrm{p}})_{\mathrm{root}} \coloneqq \mathrm{id}_{[\![\mathrm{p.root}]\!]}$ and $(\mathrm{id}_{\mathrm{p}})_{\mathrm{rest}_{i,\,d}} \coloneqq \mathrm{id}_{\mathrm{p.rest}_{i,\,d}}$ 余归纳地定义 $\mathrm{id}_{\mathrm{p}} \in \mathbf{PolyTr}_{\mathbb{U}}(\mathrm{p},\,\mathrm{p})$。

复合。 给定 $\varphi : \mathbf{PolyTr}_{\mathbb{U}}(\mathrm{p},\mathrm{q})$ 和 $\psi : \mathbf{PolyTr}_{\mathbb{U}}(\mathrm{q},\mathrm{r})$，我们通过归纳定义 $\psi \circ \varphi : \mathbf{PolyTr}_{\mathbb{U}}(\mathrm{p},\mathrm{r})$。在 $\mathbf{PolyTr}_{\mathbb{U}}^{(0)} = 1$ 处，复合是平凡的。在第 $n+1$ 层，我们必须构造一个元素，属于

$$\prod_{i:|\mathrm{p.pos}|}\ \sum_{k:|\mathrm{r.pos}|}\ \prod_{f:|\mathrm{r.dir}_k|}\ \sum_{d:|\mathrm{p.dir}_i|} \mathbf{PolyTr}_{\mathbb{U}}^{(n)}(\mathrm{p.rest}_{i,d},\,\mathrm{r.rest}_{k,f}).$$

给定 i，令 $j \coloneqq \varphi_1(i)$ 和 $k \coloneqq \psi_1(j)$。给定 $f\colon |\mathrm{r.dir}_k|$，令 $e \coloneqq \psi_j^{\#}(f) : |\mathrm{q.dir}_j|$，以及 $d \coloneqq \varphi_i^{\#}(e) : |\mathrm{p.dir}_i|$。递归分量为

$$\psi.\mathrm{rest}_{j,f} \circ \varphi.\mathrm{rest}_{i,e} : \mathbf{PolyTr}_{\mathbb{U}}^{(n)}(\mathrm{p.rest}_{i,d},\,\mathrm{r.rest}_{k,f})$$

由归纳假设定义。

结合律和单位律通过归纳在每一层 $\mathbf{PolyTr}_{\mathbb{U}}^{(n)}$ 都成立，基例$\mathbf{PolyTr}_{\mathbb{U}}^{(0)} = 1$ 则显然成立。因此，它们在极限中也成立。 □

命题 3.9. 对于 $p, q:\ \mathbf{Poly}_{\mathbb{U}}$，$\mathbf{PolyTr}_{\mathbb{U}}(\mathrm{p},\mathrm{q})$ 是终端 $[p,q]$-余代数：

$$\mathbf{PolyTr}_{\mathbb{U}}(\overline{\mathrm{p}},\overline{\mathrm{q}}) \cong \mathsf{y}^{[p,q]}(1).$$

证明。因为 $\mathrm{p}_\circ$ 剩余$_{i,\,d} = \overline{\mathrm{p}}$ 和 $\mathrm{q}_\circ$ 剩余$_{j,\,e} = \overline{\mathrm{q}}$ 对所有 $i,\ d,\ j,\ e$，递推关系（10）变为

$$\mathbf{PolyTr}_{\mathbb{U}}^{(n+1)}(\overline{\mathrm{p}},\overline{\mathrm{q}}) = \prod_{i:p(1)}\ \sum_{j:q(1)}\ \prod_{e:q[j]}\ \sum_{d:p[i]} \mathbf{PolyTr}_{\mathbb{U}}^{(n)}(\overline{\mathrm{p}},\overline{\mathrm{q}}) = [p,q]\big(\mathbf{PolyTr}_{\mathbb{U}}^{(n)}(\overline{\mathrm{p}},\overline{\mathrm{q}})\big)$$

与$\mathbf{PolyTr}_{\mathbb{U}}^{(0)}(\mathrm{p},\overline{\mathrm{q}}) = 1$。将 (7) 在 1 处求值： $\mathbf{Set}$ 以 $[p,q]$ 代替 p，并使用$y(1) = 1$，得到相同的递推关系，因此极限为 $\mathsf{y}^{[p,q]}(1)$。 □

[4]为了对可能以失败告终的对局建模，在国际象棋树的每个节点都加入一个终端位置 WhiteLose（空方向集），这样 $\mathrm{y} \to$ 国际象棋就是任意白方策略。令 $\mathrm{o} \coloneqq \mathrm{y} + \{\mathrm{WhiteLose},\ \mathrm{BlackLose}\}$ 为结果树。黑方反策略是一个映射 $\mathrm{chess} \to \mathrm{o}$，它与将 {WhiteLose} 嵌入每一侧的映射相交换（因此黑方无法错误归咎白方失败）；黑方被将死时映射到 BlackLose。复合 $\mathrm{y} \to \mathrm{o}$ 即为对局结果：每一轮要么继续（y），要么以胜利结束。

范畴 **PolyTr**$_{\mathbb{U}}$ 有有限余积。对于多项式树 p，$\mathrm{q}:\mathfrak{c}_{\mathrm{u}\triangleleft\mathrm{u}}(1)$，定义 $\mathrm{p}+\mathrm{q}:\mathfrak{c}_{\mathrm{u}\triangleleft\mathrm{u}}(1)$ 为

$$[\![(\mathrm{p}+\mathrm{q}).\mathtt{root}]\!] \coloneqq [\![\mathrm{p}.\mathtt{root}]\!] + [\![\mathrm{q}.\mathtt{root}]\!]$$

对于 $(\mathrm{p}+\mathrm{q}).\mathrm{rest}_{i,\,d} \coloneqq \mathrm{p}.\mathrm{rest}_{i,\,d}$取 $i:|\mathrm{p}.\mathrm{pos}|$，以及 $(\mathrm{p}+\mathrm{q}).\mathrm{rest}_{j,\,e} \coloneqq \mathrm{q}.\mathrm{rest}_{j,\,e}$取 $j:|\mathrm{q}.\mathrm{pos}|$。无需余归纳：子节点直接继承自各直和项。

命题 3.10. **PolyTr**$_{\mathbb{U}}$ 有有限余积，其中 $\mathbb{0}$是始对象。

证明。包含映射 $\iota_1:\mathrm{p}\to\mathrm{p}+\mathrm{q}$ 和 $\iota_2:\mathrm{q}\to\mathrm{p}+\mathrm{q}$ 由根处的余积包含以及子树上的恒等映射定义。给定 $\varphi:\mathrm{PolyTr}_{\mathbb{U}}(\mathrm{p},\mathrm{r})$ 和 $\psi:\mathrm{PolyTr}_{\mathbb{U}}(\mathrm{q},\mathrm{r})$，余配对 $(\varphi,\psi):\mathrm{PolyTr}_{\mathbb{U}}(\mathrm{p}+\mathrm{q},\mathrm{r})$ 在根处定义为 $(\varphi.\mathrm{root},\psi.\mathrm{root})$，其子树数据在第一个直和项上限制为 φ 的数据，在第二个直和项上限制为 ψ 的数据。唯一性在每一层都成立。对于始对象，对所有 n，都有 $\mathrm{PolyTr}_{\mathbb{U}}^{(n+1)}(0,\mathrm{q}) = \prod_{i:\mathbb{0}}\cdots = 1$，因此 $\mathrm{PolyTr}_{\mathbb{U}}(0,\mathrm{q}) = 1$。□

第 3.1 节末尾的常值树构造可推广到态射：给定 $\varphi:\ p\to q$ in **Poly**$_{\mathbb{U}}$，定义 $\overline{\varphi}:\mathrm{p}\to\mathrm{q}$，方法是 $\overline{\varphi}.\mathrm{根} \coloneqq \varphi$和 $\overline{\varphi}.\mathrm{余部}_{i,\,e} \coloneqq \overline{\varphi}$余归纳地，在每个深度应用 $\overline{(-)}$。这定义了一个忠实函子

$$\overline{(-)}:\ \mathbf{Poly}_{\mathbb{U}}\to\mathbf{PolyTr}_{\mathbb{U}}. \tag{11}$$

事实上，忠实性可由回缩 $\varphi.\mathrm{根} = \varphi$ 立即得出。

注 3.11。函子 $\overline{(-)}$ 不保余积。对于 $p,\ q:\mathbf{Poly}_{\mathbb{U}}$，树 $\mathrm{p}+\mathrm{q}$ 和 $\mathrm{p}+\mathrm{q}$ 具有相同的根多项式 $\overline{p}+\overline{q}$，但它们的子节点不同：$(\mathrm{p}+\mathrm{q})$。余部$_{i,\,d}$ = 在 p 直和项上为 p，在 q 直和项上为 q；而 $\mathrm{p}+\overline{\mathrm{q}}$。余部$_{i,\,d}$ =在两个直和项上均为 $\mathrm{p}+\mathrm{q}$。◊

3.3 幺半结构

本节将狄利克雷张量积从**Poly** 提升到 **PolyTr**$_{\mathbb{U}}$，并证明它对余积分配。对于多项式树 p，$\mathrm{q}:\mathfrak{c}_{\mathrm{u}\triangleleft\mathrm{u}}(1)$，通过余归纳定义张量积$\mathrm{p}\otimes\mathrm{q}:\mathfrak{c}_{\mathrm{u}\triangleleft\mathrm{u}}(1)$：

$$[\![(\mathrm{p}\otimes\mathrm{q}).\mathtt{root}]\!] \coloneqq [\![\mathrm{p}.\mathtt{root}]\!]\otimes[\![\mathrm{q}.\mathtt{root}]\!]$$
$$(\mathrm{p}\otimes\mathrm{q}).\mathtt{rest}_{(i,j),(d,e)} \coloneqq \mathrm{p}.\mathtt{rest}_{i,d}\otimes\mathrm{q}.\mathtt{rest}_{j,e}$$

对于 $i:|\mathrm{p}.\text{位置}|$，$j:|\mathrm{q}.\text{位置}|$，$d:|\mathrm{p}.\text{方向}_i|$，$e:|\mathrm{q}.\text{方向}_j|$。这是余归纳地良定义的：由于 $\mathbb{U}$ 在依赖积下封闭，根编码属于 $(\mathrm{u}\triangleleft\mathrm{u})(1)$，而子节点仍然是树对，因此这一构造会在每个节点处重复。

例 3.12 (干细胞培育的器官)。考虑三个来自例p 的干细胞副本3.3，每个的根都是 Ly^{L^3}。它们的张量$\mathrm{p}^{\otimes 3}$ 的根是 $L^3y^{L^9}$：位置$(x_1,\ x_2,\ x_3):\ L^3$是细胞的输出浓度，而方向 $(d_1,\ d_2,\ d_3):\ L^3\times L^3\times L^3$ 是九个受体信号。

将器官建模为常值树 $\mathrm{q} \coloneqq \mathrm{Ly}^{\mathrm{L}}$：一个聚合输出浓度、一个环境信号，且随时间保持不变。态射 $\varphi:\mathbf{PolyTr}_{\mathbb{U}}(\mathrm{p}^{\otimes 3},\mathrm{q})$

用三个演化细胞实现这一固定的外部接口。在根处，φ.root 是一个多项式映射 $L^3\mathrm{y}^{L^9} \to L\mathrm{y}^L$，由一个聚合 $\varphi_1 := \alpha\colon L^3 \to L$ 给出；对于每个 $(x_1, x_2, x_3)\colon L^3$，还有一个分布 $\varphi^\#(x_1,x_2,x_3)\colon L \to L^9$，将环境信号路由至九个受体。该分布可以依赖于细胞的当前输出：例如，它可以优先向产生低浓度的细胞发送信号。经过一轮后，在输出 (x_1, x_2, x_3)、信号为 $\mathrm{l}\colon L$的情况下，每个细胞 k 接收一个由$\varphi^\#(x_1,x_2,x_3)(\mathrm{l})$决定的三元组 $d_k\colon L^3$，并转变为 $\mathrm{p.rest}^{x_k,d_k}$，其根多项式可能不同于 $L\mathrm{y}^{L^3}$(例 3.3)。因此，子张量积 $\mathrm{p.rest}_{x_1,d_1} \otimes \mathrm{p.rest}_{x_2,d_2} \otimes \mathrm{p.rest}_{x_3,d_3}$ 具有新的根多项式，但 q 仍为 $L\mathrm{y}^L$；态射 φ 必须在此深度提供一个新的多项式映射，以适应变化后的细胞接口，之后也依此余归纳地继续。◊

命题 3.13。 $(\mathbf{PolyTr}_{\mathbb{U}}, 0, +, y, \bar{\otimes})$ 是一个对称分配幺半范畴。

证明。 我们沿着塔归纳构造所需的数据。

的函子性。 给定 $\varphi\colon \mathbf{PolyTr}_{\mathbb{U}}(\mathrm{p},\mathrm{q})$ 和 $\varphi'\colon \mathbf{PolyTr}_{\mathbb{U}}(\mathrm{p}',\mathrm{q}')$, 通过归纳定义 $\varphi \otimes \varphi'\colon \mathbf{PolyTr}_{\mathbb{U}}(\mathrm{p} \otimes \mathrm{p}', \mathrm{q} \otimes \mathrm{q}')$。在层级 $n+1$，我们必须构造一个

$$\prod_{(i,i')} \sum_{(j,j')} \prod_{(e,e')} \sum_{(d,d')} \mathbf{PolyTr}_{\mathbb{U}}^{(n)}(\mathrm{p.rest}_{i,d} \otimes \mathrm{p}'.\mathrm{rest}_{i',d'},\ \mathrm{q.rest}_{j,e} \otimes \mathrm{q}'.\mathrm{rest}_{j',e'}).$$

给定 (i, i')，令 $(j, j') := (\varphi_1(i),\ \varphi_1'(i'))$。给定 (e, e')，令 $(d, d') := (\varphi_i^\#(e),\ \varphi_{i'}'^\#(e'))$.[5] 递归分量为 φ。$\mathrm{rest}_{i,e} \otimes \varphi'$。$\mathrm{rest}_{i',e'}$，由归纳假设定义。

幺半结构。 结合子、单位子 $\mathrm{y} \otimes \mathrm{p} \xrightarrow{\cong} \mathrm{p} \xleftarrow{\cong} \mathrm{p} \otimes \mathrm{y}$，以及编织 $\mathrm{p} \otimes \mathrm{q} \xrightarrow{\cong} \mathrm{q} \otimes \mathrm{p}$，均通过余归纳定义：在每个根处，应用（**Poly**，y，$\otimes$）中的相应同构，并注意子节点仍是张量积（对于编织，顺序交换），因此该构造会递归进行。每一层 $\mathbf{PolyTr}_{\mathbb{U}}^{(n)}$ 上的相干性都由 **Poly**中的相干性推出。

分配性。 在根处，**Poly** 中的分配性给出 $[\![\mathrm{p.根}]\!] \otimes ([\![\mathrm{q.根}]\!] + [\![\mathrm{r.根}]\!]) \cong ([\![\mathrm{p.根}]\!] \otimes [\![\mathrm{q.根}]\!]) + ([\![\mathrm{p.根}]\!] \otimes [\![\mathrm{r.根}]\!])$。由于 $+$只作用于根处——子节点继承自直和项——两边的子节点已经完全相同：各自都是 $\mathrm{p.余}_{i,\ d} \otimes \mathrm{q.余}_{j,\ e}$或者 $\mathrm{p.余}_{i,\ d} \otimes \mathrm{r.余}_{k,\ f}$。因此，该同构完全存在于根层面，无需余归纳。□

推论 3.14. 常值树函子 (11) 是严格幺半的：$\mathrm{p} \otimes \mathrm{q} = \mathrm{p} \otimes \mathrm{q}$ 对所有 $\bar{p}, q : \overline{\mathbf{Poly}}_{\mathbb{U}}$，并且$y$是两个范畴中的幺半单位。

证明. $\mathrm{p} \otimes \mathrm{q}$ 和 $\mathrm{p}\bar{\otimes}\mathrm{q}$ 的根都是 $p \otimes q$，且每个子节点都等于整棵树：$(\mathrm{p} \otimes \mathrm{q}).\mathrm{rest}_{(i,j),(\bar{d},e)} \equiv \mathrm{p} \otimes \mathrm{q}$ 和 $\mathrm{p} \otimes \mathrm{q}.\bar{\mathrm{rest}}_{(i,j),(d,e)} = \overline{\mathrm{p} \otimes \mathrm{q}}$。由终余代数的唯一性可知，它们相等。□

注 3.15. 范畴 $\mathbf{PolyTr}_{\mathbb{U}}$ 没有有限积：给定态射$\alpha : \mathbf{PolyTr}_{\mathbb{U}}(\mathrm{r},\mathrm{p})$ 和 $\beta : \mathbf{PolyTr}_{\mathbb{U}}(\mathrm{r},\mathrm{q})$，积需要将它们组合成一个指向 $\mathrm{p} \times \mathrm{q}$ 的态射，但从 p 出发的方向会迫使 r 转移到一个子树，而关于该子树

[5]这种逐分量构造是 双幺半 交换的一个实例，它发生于 $\otimes$ 和 $\triangleleft$之间：幺半结构 $(\mathbf{Poly}, y, \otimes)$ 和 $(\mathbf{Poly}, y, \triangleleft)$ 构成一个正规双幺半范畴 [AM10; BS11], 因此存在一个自然映射 $(p \triangleleft p') \otimes (q \triangleleft q') \to (p \otimes q) \triangleleft (p' \otimes q')$，它在元素层面恰好给出了此处使用的独立配对。

β 没有携带任何信息。然而，$\mathrm{PolyTr}_{\mathbb{U}}$ 是一个 $(\mathrm{Poly}, 1, \times)$-作用范畴：对于 $p : \mathrm{Poly}_{\mathbb{U}}$ 和 $\mathrm{q} : \mathrm{PolyTr}_{\mathbb{U}}$，余归纳地定义 $p \times \mathrm{q} : \mathrm{PolyTr}_{\mathbb{U}}$，使得 $[\![(p \times \mathrm{q}).\mathrm{root}]\!] \coloneqq p \times [\![\mathrm{q}.\mathrm{root}]\!]$，其子树为 $(p \times \mathrm{q}).\mathrm{rest}_{(i,j),d} \coloneqq p \times \mathrm{q}$，当 $d : p[i]$ 时；并且 $(p \times \mathrm{q}).\mathrm{rest}_{(i,j),e} \coloneqq p \times \mathrm{q}.\mathrm{rest}_{j,e}$，当 $e : |\mathrm{q}.\mathrm{dir}_j|$ 时。由于 p 是静态的，p-方向不会引发转移，而作用范畴的相干性可由 $(\mathrm{Poly}, 1, \times)$ 的相干性余归纳地推出。$\lozenge$

3.4 幺半闭性

本节我们证明，狄利克雷张量积在 $\mathbf{PolyTr}_{\mathbb{U}}$ 上是闭的，其内部 hom 由与态射集相同的塔定义（定义 3.5），但基础情形用 y 代替 1。

对于多项式树 $\mathrm{p}, \mathrm{q} : \mathfrak{c}_{\mathrm{u} \triangleleft \mathrm{u}}(1)$，定义内部 *hom* 多项式 $[\mathrm{p}, \mathrm{q}]^{(n)}$，其中 $n : \mathbb{N}$，由 $[\mathrm{p}, \mathrm{q}]^{(0)} \coloneqq \mathrm{y}$ 以及

$$[\mathrm{p}, \mathrm{q}]^{(n+1)} \coloneqq \prod_{i : |\mathrm{p}.\mathrm{pos}|} \sum_{j : |\mathrm{q}.\mathrm{pos}|} \prod_{e : |\mathrm{q}.\mathrm{dir}_j|} \sum_{d : |\mathrm{p}.\mathrm{dir}_i|} [\mathrm{p}.\mathtt{rest}_{i,d}, \mathrm{q}.\mathtt{rest}_{j,e}]^{(n)}. \tag{12}$$

每个 $[\mathrm{p}, \mathrm{q}]^{(n)}$ 都是元数属于 $\mathbb{U}$ 的多项式，这是由于在依赖和与积下封闭。定义 $[\mathrm{p}, \mathrm{q}] \coloneqq \mathrm{lim}_n [\mathrm{p}, \mathrm{q}]^{(n)}$。

比较（12）与（10），这两个塔具有相同的 $\prod \sum \prod \sum$ 递推关系；它们仅在基础情形上不同：y 与 1。由于 $y(1) = 1$ 且 $\sum$，多项式的 $\prod$ 与在任意集合处的求值可交换，我们得到

$$\mathbf{PolyTr}_{\mathbb{U}}(\mathrm{p}, \mathrm{q}) \;=\; [\mathrm{p}, \mathrm{q}](1) \tag{13}$$

多项式树版本的 $\mathbf{Poly}(p, q) \;=\; [p, q](1)$。对于常值树，塔（12）退化为固定多项式 $[p, q]$ 的余自由余单子塔（7），从而恢复命题 3.9。

定理 3.16。$(\mathbf{PolyTr}_{\mathbb{U}}, y, \otimes, [-,-])$ 是一个幺半闭范畴：对于所有多项式树 p、q、r，

$$\mathbf{PolyTr}_{\mathbb{U}}(\mathrm{r} \otimes \mathrm{p}, \mathrm{q}) \cong \mathbf{PolyTr}_{\mathbb{U}}(\mathrm{r}, [\mathrm{p}, \mathrm{q}]).$$

证明。我们证明，对所有 n，都有 $\mathbf{PolyTr}^{(n)}_{\mathbb{U}}(\mathrm{r} \otimes \mathrm{p}, \mathrm{q}) \cong \mathbf{PolyTr}^{(n)}_{\mathbb{U}}(\mathrm{r}, [\mathrm{p}, \mathrm{q}])$；通过归纳，并与投影相容，因此该同构可传递到极限。

在 $n = 0$ 处，两边均为 1。对于归纳步骤，记 $I \coloneqq |\mathrm{p}.\mathrm{pos}|$，$J \coloneqq |\mathrm{q}.\mathrm{pos}|$，$K \coloneqq |\mathrm{r}.\mathrm{pos}|$，$D_i \coloneqq |\mathrm{p}.\mathrm{dir}_i|$，$E_j \coloneqq |\mathrm{q}.\mathrm{dir}_j|$，$F_k \coloneqq |\mathrm{r}.\mathrm{dir}_k|$。分别在根多项式 $[\![\mathrm{r}.\mathrm{root}]\!] \otimes [\![\mathrm{p}.\mathrm{root}]\!]$ 和 $[\mathrm{p}, \mathrm{q}]^{(1)}$ 处展开同态塔（10），并应用类型论选择公理

$$\prod_{a : A} \sum_{b : B_a} P(a, b) \;\cong\; \sum_{b : \prod_{a:A} B_a} \prod_{a : A} P(a, b(a)) \tag{14}$$

对后者作用两次以恢复 $\prod \sum \prod \sum$ 结构 $[\mathrm{p}, \mathrm{q}]^{(1)}$，得到

$$\mathbf{PolyTr}^{(n+1)}_{\mathbb{U}}(\mathrm{r} \otimes \mathrm{p}, \mathrm{q}) \;=\; \prod_{k : K} \prod_{i : I} \sum_{j : J} \prod_{e : E_j} \sum_{f : F_k} \sum_{d : D_i} T_{k,i,j,e,f,d} \tag{15}$$

$$\mathbf{PolyTr}_{\mathbb{U}}^{(n+1)}(\mathrm{r},\,[\mathrm{p},\mathrm{q}]) \;\cong\; \prod_{k:K}\prod_{i:I}\sum_{j:J}\prod_{e:E_j}\sum_{d:D_i}\sum_{f:F_k} T_{k,i,j,e,f,d} \tag{16}$$

其中 $T_{k,i,j,e,f,d} \coloneqq \mathrm{PolyTr}_{\mathbb{U}}^{(n)}(\mathrm{r.rest}_{k,f} \otimes \mathrm{p.rest}_{i,d},\, \mathrm{q.rest}_{j,e})$。根据归纳假设，$T \cong \mathrm{PolyTr}_{\mathbb{U}}^{(n)}(\mathrm{r.rest}_{k,f},\, [\mathrm{p.rest}_{i,d},\, \mathrm{q.rest}_{j,e}])$。这两个表达式的区别仅在于交换 $\sum_{d:D_i}$ 和 $\sum_{f:F_k}$。与投影的相容性显而易见。□

注 3.17。集合 $[\mathrm{p},\mathrm{q}]^{(1)}(1) = \prod_{i:|I|}\sum_{j:|J|}\prod_{e:|E_j|}|D_i|$ 是定义良好的，其基数由依赖积和和下的封闭性可知属于 $\mathbb{U}$。然而，$[\mathrm{p},\mathrm{q}]^{(1)}(1)$ 并非严格地具有 $|K|$ 的形式：它是典范代表的依赖积，本身并不是典范代表。因此，$[\mathrm{p},\mathrm{q}]^{(1)}$ 不属于 $\mathrm{Poly}_{\mathbb{U}}$，而只是与 $\mathrm{Poly}_{\mathbb{U}}$ 中的一个多项式同构（第 2.4 节）。这与阻止 $\mathrm{u} \triangleleft \mathrm{u}$ 成为单子的原因相同：乘法要求类型论选择公理 (14) 是集合间的等式，而不能仅仅是双射，并且不存在这样的分配律 [ZM19]。不过，同态塔 (10) 只使用底层集合 $[\mathrm{p},\mathrm{q}]^{(1)}(1)$ 和 $[\mathrm{p},\mathrm{q}]^{(1)}[\varphi]$，并不使用它们的任何 $\mathbb{U}$ 表示，而 (14) 是集合层面的双射。因此，定理 3.16 的伴随不受影响。◊

4 与 Poly 和 $\mathbb{O}$rg 的关系；双范畴 $\mathbb{O}$rgTr

元素 $\varphi : \mathrm{PolyTr}_{\mathbb{U}}(\mathrm{p},\mathrm{q})$ 为每条交互路径上的每个节点指定一个多项式映射。实际上，用状态集和转移规则来描述态射更为自然。这是一种熟悉的模式：对于任意自函子 p: Set $\to$ Set，p-余代数 $(S, \beta\colon S \to p(S))$ 可以简洁地编码终端 p-余代数 $\zeta_p(1)$ 中的一族元素；每个状态 $s : S$ 都通过余归纳展开确定一棵行为树。在用于 [SS22]，的双范畴 $\mathbb{O}$rg 中，这一思想应用于内部同态多项式 $[p,q]$：$\mathbb{O}$rg 中的态射 $p \to q$ 是一个 $[p,q]$-余代数 $(S, \beta\colon S \to [p,q](S))$，它为每个 $s : S$ 指定一个作用 $\mathrm{act}(s) : \mathrm{Poly}(p,q)$，以及一个更新 $\mathrm{upd}_s\colon [p,q][\mathrm{act}(s)] \to S$，用于为每个方向指定一个后继状态。例如，通过权重更新规则（权重空间 S 上的余代数）来建模梯度下降，要比展示所有可能训练轨迹构成的完整树实用得多；后者是 $\zeta_{[p,q]}(1) = \mathrm{PolyTr}_{\mathbb{U}}(\mathrm{p},\mathrm{q})$ 的一个元素（命题 3.9）。

– –

在本节中，我们对一般多项式树采取同样的做法：用状态集 $S \in \mathbb{U}$ 对同态塔进行参数化，从而对每一对 p、q 得到一个范畴 $\mathbb{O}\mathrm{rgTr}(\mathrm{p},\mathrm{q}) : \mathrm{Cat}$，其对象是基于状态的描述。集合 $\mathrm{PolyTr}_{\mathbb{U}}(\mathrm{p},\mathrm{q})$ 可通过基数为 $S = 1$ 的状态集加以恢复（命题 4.5）。对于常值树，$[p,q]$-余代数作为时间不变对象完全忠实地嵌入 $\mathbb{O}\mathrm{rgTr}(\mathrm{p},\mathrm{q})$（命题 4.6），并且常值树函子 (11) 延拓为一个 Cat-富集函子。

4.1 参数化同态

在本节中，我们通过引入状态集 S，推广同态塔（10），从而可以用作用与更新规则来描述态射，而不必显式给出多项式映射树。

定义 4.1。对于多项式树 $\mathrm{p}、\mathrm{q} : \mathfrak{c}_{\mathrm{u}\triangleleft\mathrm{u}}(1)$ 以及集合 $S \in \mathbb{U}$，定义集合 $H_S^{(n)}(\mathrm{p},\mathrm{q})$，其中 $n : \mathbb{N}$，定义如下：$H_S^{(0)}(\mathrm{p},\mathrm{q}) \coloneqq 1$ 并且

$$H_S^{(n+1)}(\mathrm{p},\mathrm{q}) \coloneqq \prod_{s:S} \prod_{i:|\mathrm{p.pos}|} \sum_{j:|\mathrm{q.pos}|} \prod_{e:|\mathrm{q.dir}_j|} \sum_{d:|\mathrm{p.dir}_i|} \sum_{s':S} H_S^{(n)}(\mathrm{p.rest}_{i,d}, \mathrm{q.rest}_{j,e}). \tag{17}$$

该参数化同态集合是 $H_S(\mathrm{p}、\mathrm{q}) \coloneqq \lim_{n:\mathbb{N}} H_S^{(n)}(\mathrm{p}、\mathrm{q})$。 ◊

与（10）相比，参数化塔将原对象封装在$\prod_{s:\,S}$ 中，并在递归项之前插入$\sum_{s':\,S}$。记 $h \coloneqq [[\mathrm{p}。根]、[\mathrm{q}。根]]$：如第**3.2**节中的Poly。（$(\mathrm{p}、\mathrm{q})$ 的一个元素为每个状态指定 $s : S$：

- 一个作用$\mathrm{act}(s)$：$h(1) = \mathbf{Poly}([\mathrm{p}。\mathrm{root}]、[\mathrm{q}。\mathrm{root}])$；以及
- 对于每个方向 $(i、e)$：$h[\mathrm{act}(s)]$，令 $j \coloneqq \mathrm{act}(s)_1(i)$和 $d \coloneqq \mathrm{act}(s)_{\sharp\, i}(e)$，一个后继状态$\mathrm{upd}(s、i、e)$：$S$以及 $H_{(n)\,S}(\mathrm{p}。\mathrm{rest}_{i、d}、\mathrm{q}。\mathrm{rest}_{j、e})$的一个元素。

后继状态记录下一交互步骤中要应用的多项式映射；递归元素确保该结构贯穿所有子树持续存在。

该结构是路径依赖的：子树对 $(\mathrm{p.rest}_{i,\,d},\ \mathrm{q.rest}_{j,\,e})$ 会随每次交互而变化，因此每个节点处的递归数据所涉及的多项式同态可能不同于其父节点处的多项式同态。这正是它与 $[p,q]$-余代数的关键区别，在后者中，p 和 q 是固定的，而映射 $S \to [p,q](S)$ 始终相同。

定义 4.2。对于多项式树 $\mathrm{p}，\mathrm{q} : \mathfrak{c}_{\mathrm{u}\triangleleft\mathrm{u}}(1)$，同态范畴 $\mathbb{O}\mathbf{rgTr}(\mathrm{p}，\mathrm{q})$ 定义如下。

- 对象：形如 $(S,\ \alpha)$ 的对，其中 $S \in \mathbb{U}$ 和 $\alpha : H_S(\mathrm{p}，\mathrm{q})$。
- 态射 $(S,\ \alpha) \to (S\prime,\ \alpha\prime)$：函数 $f : S \to S\prime$，使得对于所有 $s : S$，(i) $\mathrm{act}\prime(f(s)) = \mathrm{act}(s)$；(ii) $f(\mathrm{upd}(s,\ i,\ e)) = \mathrm{upd}\prime(f(s),\ i,\ e)$ 对所有方向 $(i,\ e)$ 均成立；并且在所有子树对处余归纳地满足相同条件，其中 f 在每个节点处关联后继状态。◊

$\mathbb{O}\mathbf{rgTr}(\mathrm{p}，\mathrm{q})$ 中的复合和恒等继承自集合之间的函数。

命题 4.3. $\mathbb{O}\mathbf{rgTr}$是一个双范畴，其复合 $(S,\ \alpha) \fatsemi (T,\ \beta) \coloneqq (S \times T,\ \ \alpha \fatsemi \beta)$通过按命题 *3.8* 中的方式复合作用并配对后继状态来定义。p 上的恒等态射为p 是$(1, \mathrm{id}_{\mathrm{p}})$。

证明。给定 $(S,\alpha) : \mathbb{O}\mathrm{rgTr}(\mathrm{p},\mathrm{q})$ 和 $(T,\beta) : \mathbb{O}\mathrm{rgTr}(\mathrm{q},\mathrm{r})$，归纳定义 $\alpha \fatsemi \beta : H_{S\times T}(\mathrm{p},\mathrm{r})$。对于 $(s,t) : S \times T$ 和 $i : |\mathrm{p.pos}|$，令 $j \coloneqq \mathrm{act}_\alpha(s)_1(i)$ 和 $k \coloneqq \mathrm{act}_\beta(t)_1(j)$；对于 $f : |\mathrm{r.dir}_k|$，令 $e \coloneqq \mathrm{act}_\beta(t)^\sharp_j(f)$ 和 $d \coloneqq \mathrm{act}_\alpha(s)^\sharp_i(e)$。那么

$$\mathrm{act}_{\alpha;\beta}(s,t) \coloneqq \mathrm{act}_\alpha(s) \fatsemi \mathrm{act}_\beta(t), \qquad \mathrm{upd}_{\alpha;\beta}((s,t),i,f) \coloneqq \big(\mathrm{upd}_\alpha(s,i,e),\ \mathrm{upd}_\beta(t,j,f)\big).$$

递归因子 $(\mathrm{p.rest}_{i,d}, \mathrm{r.rest}_{k,f})$ 由 α 和 β 在中间子树处的递归数据，根据命题 3.8 余归纳地定义。恒等映射$(1, \mathrm{id}_{\mathrm{p}})$ 在每个节点处余归纳地应用恒等多项式映射。

对于态射，给定各自 Hom-范畴中的 $\sigma : (S,\alpha) \to (S',\alpha')$ 和 $\tau : (T,\beta) \to (T',\beta')$，$\sigma \times \tau : S \times T \to S' \times T'$ 交织复合作用：定义 (i) 的条件

成立，因为 4.2 中的定义 $\mathrm{act}_{\alpha'}(\sigma(s)) = \mathrm{act}_{\alpha}(s)$ 并且$\mathrm{act}_{\beta'}(\tau(t)) = \mathrm{act}_{\beta}(t)$，因此复合作用一致；条件 (ii) 成立，因为 σ 和 τ 分别交织后继状态。余归纳条件可通过对子树应用相同论证得到。塔的每一层都满足结合律和单位律，由归纳法可知，极限中也成立。 □

我们通过一个深度学习中的例子来说明双范畴结构。令$t \coloneqq \sum_{x:\mathbb{R}} \mathrm{y}^{T_x^*\mathbb{R}}$ 表示单个实值通道的多项式编码：位置是实数，方向是余切向量（梯度反馈）。在 [SS22], 中，将 m 输入映射到 n 输出的梯度下降器是一个状态为（M， f， p）的 $[t^{\otimes m}, t^{\otimes n}]$-余代数：参数维数 $M:\ \mathbb{N}$，可微函数 $f:\ \mathbb{R}^{M+m} \to \mathbb{R}^n$，以及当前参数 $p:\ \mathbb{R}^M$。其作用是应用 f $(p,\ \text{–})$进行前向传播，并执行反向传播（Df）$^\top$；更新则对 p执行梯度下降。这些是 $\mathbb{O}$rg-态射：接口 t 始终保持不变。

例 4.4（渐进式生成架构）。 在生成对抗网络的渐进式训练中 [Kar+18]，图像分辨率会随着训练逐步提高：生成器最初生成尺寸较小、像素较粗糙的图像，然后逐渐过渡到尺寸更大、分辨率更高的图像，在学习精细细节之前先学习全局结构。输出多项式会在每个分辨率步骤发生变化，因此生成器的接口在训练过程中以受控方式演变，这正是多项式树旨在形式化的内容。令 $N_0 < N_1 < \cdots < N_L$ 为分辨率逐渐提高时的像素数。多项式$t^{\otimes N_l} = \sum_{x:\ \mathbb{R}^{N_l}} \mathrm{y}^{T_x^*\mathbb{R}^{N_l}}$ 是分辨率为 l时的接口：位置是图像（N_l 实值像素强度），方向是携带梯度反馈的余切向量。

渐进式图像树。 定义一个多项式树 $\mathrm{g}_0:\ \mathfrak{c}_{\mathrm{u}\triangleleft\mathrm{u}}(1)$，其根为 g_0。根 $\coloneqq \ulcorner t^{\otimes N_0} \urcorner$。每幅图像 $x:\ \mathbb{R}^{N_0}$ 处的方向集 $T_x^*\mathbb{R}^{N_0}$ 划分为

$$T_x^*\mathbb{R}^{N_0} = D_x^{\mathrm{stay}} \sqcup D_x^{\mathrm{grow}}$$

（例如， $D_x^{\mathrm{stay}} \coloneqq \{v:\ \|v\| > \epsilon\}$ 对于阈值 $\epsilon > 0$：较大的梯度表示持续学习，较小的梯度表示在当前分辨率下收敛）。子树据此分支：

$$\mathrm{g}_0.\mathtt{rest}_{x,v} \coloneqq \begin{cases} \mathrm{g}_0 & \text{if } v \in D_x^{\mathrm{stay}}, \\ \mathrm{g}_1 & \text{if } v \in D_x^{\mathrm{grow}}, \end{cases}$$

其中，g_1 的定义类似，根为 $\ulcorner t^{\otimes N_1} \urcorner$，依此类推，直到 $\mathrm{g}_L \coloneqq \mathrm{t}^{\otimes N_L}$（在$\overline{\text{最终}}$分辨率下为常数）。多项式树编码了所有可能的分辨率轨迹，而梯度反馈则从中进行选择。

生成器作为一个 $\mathbb{O}$**rgTr**-态射。 设 $k:\ \mathbb{N}$ 为生成器输入空间的维度（即“潜空间”：一个固定维度的随机向量空间，图像由此生成）。生成器是一个对象

$$(W_G,\ \alpha_G) : \mathbb{O}\mathbf{rgTr}(\overline{\mathrm{t}^{\otimes k}},\ \mathrm{g}_0)$$

从常值树 $\mathrm{t}^{\otimes k}$（固定输入接口）到渐进树 g_0 $\overline{}$（演化输出接口）的同态范畴中。状态集 $W_G:\ \mathbb{U}$ 由生成器的

可学习参数组成。在分辨率层级 l 和状态 w : W_G下，作用 $\mathrm{act}^{\alpha_G}(w)$:**Poly**$(t^{\otimes k}, t^{\otimes N_l})$ 将参数化函数 $f_G(w,-)$: $\mathbb{R}^k \to \mathbb{R}^{N_l}$作为前向传播应用，而反向传播 $(D\, f_G)^\top$ 则将梯度反馈从输出传回输入。后继状态 upd $\overset{\alpha_G}{\mathbb{R}^k}$ w z v W_G v $\mathbb{R}^{N_l}$ $w \mapsto w + \epsilon\, \pi_{W_G}\, D_{(w,z)} f_G{}^\top\ v$ z (,,) :

执行梯度上升：( ) ·，其中：是输入，且：$T^*_{f_G(w,z)}$是来自判别器的梯度反馈。符号翻转（$+\epsilon$ 而非 $-\epsilon$）体现了对抗目标：生成器沿着判别器损失上升。 v 属于 D^{stay} 还是D^{grow} ，决定了子树，也就决定了下一个分辨率层级。

在 $\mathbb{O}$**rgTr** 中的复合。 判别器（W_D, α_D）: $\mathbb{O}$**rgTr**$(\mathrm{g}_0, \mathrm{t})$ 的定义方式类似：在分辨率 l 和状态 w : W_D下，作用应用 $f_D(w,-)$: $\mathbb{R}^{N_l} \to \mathbb{R}$，将图像分类为真或假，并通过梯度下降更新 W_D。在 $\mathbb{O}$**rgTr** 中的复合给出

$$(W_G \times W_D,\ \alpha_G \fatsemi \alpha_D) : \mathbb{O}\mathbf{rgTr}(\overline{\mathrm{t}^{\otimes k}},\ \overline{\mathrm{t}}).$$

这是一个联合参数空间为 $W_G \times W_D$ 的端到端系统：复合作用将一个潜向量依次输入生成器和判别器，产生单个真实输出，并通过反向传播同时更新两组参数。最外层接口 t 处的梯度反馈由损失函数（例如，针对真实数据的二元交叉熵）提供，它充当环境。

为什么 $\mathbb{O}$**rgTr**，而不是 $\mathbb{O}$**rg**。 在分辨率 l下，生成器作用为多项式映射 $t^{\otimes k} \to t^{\otimes N_l}$。由于 $N_l \neq N_{l'}$对于 $l \neq l'$，这些是在树的不同节点处不同多项式之间的映射；一个 $\mathbb{O}$**rg**-态射要求所有时刻都使用同一个固定 hom $[t^{\otimes k}, t^{\otimes N}]$。达到层级 L后，接口趋于稳定，动力学简化为一个 $\mathbb{O}$**rg**-态射，这与下文的命题 4.6 一致。

自适应与固定调度。 在 [Kar+18]中，分辨率增长遵循固定调度：训练 N 步，然后增长。这相当于使用这样一种多项式树：其中所有 方向都指向同一个子树（分支是平凡的，状态中的计数器控制转换）。上述依赖方向的表述更加自适应：梯度反馈本身会发出当前分辨率已经学会的信号，仅当训练动态表明条件成熟时，才触发向更高分辨率的转换。这可以减少在已经收敛的分辨率上浪费的计算，并防止在达到收敛之前过早增长。这个例子具有推测性；我们尚未实现它。 ◊

4.2 恢复 PolyTr$_{\mathbb{U}}$ 和 $\mathbb{O}$rg

在本节中，我们将展示**PolyTr**$_{\mathbb{U}}$ 可通过将 4.1）中的参数化同态设置为 $S = 1$而恢复，并且 $\mathbb{O}$**rg** 通过常值树全忠实地嵌入 $\mathbb{O}$**rgTr** 中。

命题 4.5。 对于所有多项式树p、q，存在一个同构

$$H_1(\mathrm{p}, \mathrm{q}) \cong \mathbf{PolyTr}_{\mathbb{U}}(\mathrm{p}, \mathrm{q}).$$

证明。当 $S = 1$时，(17) 中的因子 $\prod_s$:1 和 $\sum_{s'}$:1 是平凡的，因此通过与 (10) 比较，对于所有 n，都有 $H_1^{(n+1)}$ (p,q) =PolyTr$_{\mathbb{U}}^{(n+1)}$ (p, q)。取极限可得 H_1(p,q) ≅PolyTr$_{\mathbb{U}}$ (p, q)。□

因此，$\mathbb{O}$rgTr(p, q) 中状态集为单元素的对象恰好是 PolyTr$_{\mathbb{U}}$ 的态射。状态集更大的对象则为相同的态射提供紧凑的、基于状态的描述：任意一个 (S, α) 和初始状态 $s_0 : S$，通过余归纳地展开，都唯一确定一个 $\varphi : \mathrm{PolyTr}_{\mathbb{U}}(\mathrm{p}, \mathrm{q})$。在命题 4.5 的识别下，$\mathbb{O}$rgTr 中的复合在 $S = T = 1$ 处还原为 PolyTr$_{\mathbb{U}}$ 中的复合。

命题 4.6。对于多项式 $p, q : \mathbf{Poly}_{\mathbb{U}}$，存在一个全忠实函子 $[p, q]\text{-}\mathbf{Coalg} \to \mathbb{O}\mathbf{rgTr}(\mathrm{p}, \mathrm{q})$。

证明。记 $h := [p, q] : \mathbf{Poly}$。对于常值树，(17)变为

$$H_S^{(n+1)}(\overline{\mathrm{p}}, \overline{\mathrm{q}}) = \prod_{s:S} \prod_{i:p(1)} \sum_{j:q(1)} \prod_{e:q[j]} \sum_{d:p[i]} \sum_{s':S} H_S^{(n)}(\overline{\mathrm{p}}, \overline{\mathrm{q}}). \tag{18}$$

一个 h-余代数 $(S, \beta\colon\ S \to \mathrm{h}(S))$ 按如下方式确定一个元素 $\mathrm{h}^\beta : H_S(\overline{\mathrm{p}}, \overline{\mathrm{q}})$。由于根据 (5)，$\mathrm{h} = \prod_{i:p(1)} \sum_{j:q(1)} \prod_{e:q[j]} \sum_{d:p[i]} \mathrm{y}$，因此，元素 $\beta(s) \in \mathrm{h}(S)$ 为每个 $i : p(1)$ 指定一个位置 $j : q(1)$，并为每个方向 $e : q[j]$ 指定一个方向 $d : p[i]$，以及一个元素 $s' : S$。j 和 d 的选取给出作用 $\mathrm{act}(s) : \mathrm{h}(1)$，而元素$s'$给出后继状态 $\mathrm{upd}(s, -) : \mathrm{h}[\mathrm{act}(s)] \to S$。这正是 (18) 在每个 $s : S$ 处所需的数据。每个方向上的递归因子 $H_S^{(n)}$ 由正在定义的元素填充，因为子树都是常值树：$\mathrm{p.rest}_{i,d} = \mathrm{p}$ 和 $\mathrm{q.rest}_{j,e} = \mathrm{q}$。这个余归纳定义在每个节点处都使用相同的 β。

余代数态射 $f\colon (S, \beta) \to (S', \beta')$ 满足 $\mathrm{h}(f) \circ \beta = \beta' \circ f$，这恰好是定义 4.2 中关于 h^β 和 $\mathrm{h}^{\beta'}$的条件 (i)–(ii)。由于子树是常量，余归纳条件简化为这一层。反过来，$\mathbb{O}$rgTr(p, q) 中的任意态射$f\colon (S, \mathrm{h}^\beta) \to (S', \mathrm{h}^{\beta'})$ 都满足根节点处的 (i)–(ii)，从而有 $\mathrm{h}(f) \circ \beta = \beta' \circ f$。因此，该函子是全忠实的。□

注记 4.7。在 (18) 中，每个状态的每个方向各出现一次递归因子 $H_S^{(n)}$，因此每个状态的每个方向都独立指定 $H_S^{(n)}$ 的一个元素。$\mathbb{O}$rgTr(p, q) 的一般对象可以在每个节点使用不同的映射 $S \to \mathrm{h}(S) = [p, q](S)$：一次交互之后所选的作用和更新不必与另一次交互之后所选的相同。$[p, q]$-Coalg 的像由时间不变对象构成，即在每个节点都使用相同的 β。到根级数据的投影给出一个回缩 $\mathbb{O}\mathrm{rgTr}(\mathrm{p}, \mathrm{q}) \to [p, q]\text{-}\mathrm{Coalg}$。◊

推论 4.8. 常值树函子 (11) 扩展为一个局部全忠实双函子 $\mathbb{O}\mathbf{rg} \to \mathbb{O}\mathbf{rgTr}$：在对象上对应 $p \mapsto \mathrm{p}$，在 *Hom*-范畴上对应命题 4.6中的全忠实嵌入 $[p, q]\text{-}\mathbf{Coalg} \hookrightarrow \mathbb{O}\mathbf{rgTr}(\mathrm{p},\mathrm{q})$。

证明。我们必须检查它与复合的相容性。给定一个 $[p, q]$-余代数 (S, β) 和一个 $[q, r]$-余代数 (T, γ)，在 $\mathbb{O}$rg 中的复合通过复合作用并配对后继状态，得到 $S \times T$ 上的一个 $[p, r]$-余代数。该余代数在 $\mathbb{O}$rgTr(p, r) 中的图像，等于 (S, β) 和 (T, γ) 在命题 4.3 下的图像的复合，因为二者由相同公式定义。与恒等态射的相容性是显然的。

□

注 4.9（充实于 $\mathbb{O}\mathbf{rgTr}$）。由于 $\mathbb{O}\mathbf{rgTr}$ 是一个双范畴（命题 4.3），充实于 $\mathbb{O}\mathbf{rgTr}$ 是一个标准概念 [Kel82]：一个 $\mathbb{O}\mathbf{rgTr}$-充实范畴 $\mathcal{C}$ 由以下部分组成

一个对象集合，其中每个对象都配有一个多项式树 p_X：$\mathfrak{c}_{u \triangleleft u}(1)$，以及每对对象对应的同态对象 $(S_{X,Y}, \alpha_{X,Y})$：$\mathbb{O}\mathrm{rgTr}(\mathrm{p}_X, \mathrm{p}_Y)$，并在相应的同态范畴中配有复合与恒等态射，满足结合律和单位律。在这种结构中，对象之间的交互模式（由依赖于状态的作用 act 控制）以及对象自身的接口（由多项式树控制）都随交互共同演化。当所有对象都配有常值树时，每个同态范畴 $\mathbb{O}\mathrm{rgTr}(\mathrm{p}, \mathrm{q})$ 仍包含作为时不变对象的 $[p, q]$-Coalg（命题 4.6 和注记 4.7），但也包括在不同节点上使用不同作用的时变态射。因此，即使对于常值树，$\mathbb{O}$rgTr-充实也严格比 $\mathbb{O}$rg 对 [SS22], 的充实更一般。寻找相关的 $\mathbb{O}$rgTr-充实范畴、运算子和幺半范畴实例，留待未来研究。◊

参考文献

[AM10] Marcelo Aguiar 和 Swapneel Mahajan。《幺半函子、物种与霍普夫代数》。第 29 卷。CRM 专著丛书。美国数学学会，2010 年（引自第 12 页）。[BE15] John C. Baez 和 Jason Erbele。“控制中的范畴”。载于：《Theory and Applications of Categories》30（2015），论文第 24 号，836–881 页（引自第 1 页）。[BS11] Thomas Booker 和 Ross Street。“双幺范畴的田中对偶与卷积”。（2011）。电子预印本：1111.5659（引自第 12 页）。[BSZ14] Filippo Bonchi、Paweł Sobociński 和 Fabio Zanasi。“信号流图的范畴语义”。载于：并发理论国际会议。2014 年，第 435–450 页（引自第 1 页）。[Hon93] Kohei Honda。“二元交互的类型”。载于：CONCUR’93：并发理论。第 715 卷。计算机科学讲义。施普林格，1993 年，第 509–523 页（引自第 8 页）。[Kar+18] Tero Karras、Timo Aila、Samuli Laine 和 Jaakko Lehtinen。“渐进式生成生成对抗网络，以提升质量、稳定性和多样性”。载于：国际学习表征会议。2018 年（引自第 16、17 页）。[Kel82] Gregory Maxwell Kelly。《富化范畴论基本概念》。第 64 卷。剑桥大学出版社档案库，1982 年（引自第 18 页）。[Lib+22] Sophie Libkind、Andrew Baas、Micah Halter、Evan Patterson 和 James Fairbanks。《结构化流行病建模的代数框架》。2022 年。网址：https://arxiv.org/abs/2203.16345（引自第 1 页）。[LS24] Sophie Libkind 和 David I. Spivak。“模式作用于物质：作为余自由余单子余单子上的模的自由单子单子”。载于：第七届应用范畴论年会论文集。EPTCS。2024 年（引自第 5 页）。[Mar75] Per Martin-Löf。“直觉类型论：预定部分”。载于：《Studies in Logic and the Foundations of Mathematics》。第 80 卷。爱思唯尔，1975 年，第 73–118 页（引自第 6 页）。

[NS24] Nelson Niu 和 David I. Spivak。《多项式函子：交互的数学理论》。伦敦数学学会讲义，即将出版。剑桥大学出版社，2024 年（引自第 3 页）。[Sel10] Peter Selinger。“幺半范畴图形语言综述”。载于：《物理学的新结构》。施普林格，2010 年，第 289–355 页（引自第 1 页）。[Spi13] David I. Spivak。“布线图的拟运算：为数据库、递归和即插即用电路形式化图形语言”。（2013）。电子预印本：arXiv:1305.0297（引自第 1 页）。[Spi21] David I. Spivak。“学习者语言”。载于：第四届应用范畴论年会论文集。ACT。英国剑桥：EPTCS，2021 年（引自第 1 页）。[Spi22] David I. Spivak。《Poly 上范畴结构的参考资料》。2022 年。网址：https://arxiv.org/abs/2202.00534（引自第 3 页）。[SS22] Brandon T. Shapiro 和 David I. Spivak。“动态范畴、动态拟运算：从深度学习到预测市场”。载于：《理论计算机科学电子论文集》（2022）（引自第 2、14、16、19 页）。[VSL15] Dmitry Vagner、David I. Spivak 和 Eugene Lerman。“布线图拟运算上的开放动力系统代数”。载于：《Theory and Applications of Categories》30（2015），论文第 51 号，1793–1822 页（引自第 1 页）。[ZM19] Maaike Zwart 和 Dan Marsden。“分配律的禁行定理”。载于：第 34 届 ACM/IEEE计算机科学逻辑研讨会（LICS）论文集。arXiv:2003.12531。2019 年。doi：10.1109/LICS.2019.8785662（引自第 14 页）。