arXiv:2602.12689v1 [cs.LO] 13 Feb 2026

索引形式的迭代参数性所具有的高度依赖递归结构

HugoHerbelin Universit´eParis Cit´e、Inria、法国国家科学研究中心、IRIF、巴黎
RamkumarRamachandra
无所属机构

摘要

Reynolds 参数关系性最初为类型配备其上的证明无关的二元命题关系。但此类关系也可以取为证明相关的或一元的，并可以以带索引的或纤维化的方式加以描述。参数关系性可以迭代；当类型为集合时，这会将集合解释为：一元情形下的增广单纯集合，或二元情形下的立方集合。

在先前的工作中，给出了用以描述集合在索引形式下的n元迭代参数性翻译的等式。该构造在 Rocq 中通过对一个嵌入等式推理的大型结构进行归纳而形式化。

当前工作分析了早期工作的依赖结构，并由此给出了用定义性推理取代等式推理的构造呈现。新构造具有很强的依赖性，其基础是一种归纳过程，要求交错进行归纳假设的规约与归纳步骤的构造。同时，该构造还原为其计算本质，并可依据经过机器检查的新形式化，进行完整而细致的描述。

2012 年 ACM 学科分类 10003752.10003790.10011740

关键词与短语 参数关系性、半单纯集合、半立方集合、形式化、Rocq、带索引与纤维化对应

数字对象标识符 10.4230/LIPIcs...

1 引言

参数关系性及其影响

Reynolds 在编程语言语义的语境中提出了 [17], 关系参数关系性，它与逻辑和数学中的若干相似概念有关。在最初的表述中，它将二元命题关系与类型关联起来。 [7]中的变体将证明相关的二元关系与类型关联起来，而 [4] 中的变体则将任意元数的关系与类型关联起来。这包括一元关系的情形：在证明无关情形下，它将谓词赋予类型；在证明相关情形下，则将族赋予类型。一元关系情形表达的是属于某个类型意味着什么，而二元关系情形表达的是在某个类型中观察等价意味着什么。与 A $\to$ B 关联的规范谓词，是定义在函数 f（其中 f 属于 A $\to$ B）上的谓词；对于所有属于与 A 关联的规范谓词的 a，f (a) 都属于与 B 关联的规范谓词。类似地，与 A $\to$ B 关联的规范关系，是定义在函数 f 和 f′（其中它们属于 A $\to$ B）上的关系；对于所有通过与 A 关联的规范关系相关联的 a 和 a′，f (a) 与 f′(a′) 都通过与 B 关联的规范关系相关联。

语言上的一元参数性是一种极端形式的可实现性，它将语言本身用作实现子的语言。因此，它与可约化性有关，也称为塔特可计算性 [18], ，后者是可实现性的一种语义形式。同理，语言上的二元参数化可以看作是在自身之上定义逻辑关系 [16] ，因为逻辑关系是二元形式的可约化性。

© Hugo Herbelin 和 Ramkumar Ramachandra；采用知识共享许可协议 CC-BY 4.0 莱布尼茨国际信息学论文集许可Dagstuhl 城堡 – 莱布尼茨信息学中心¨，Dagstuhl 出版社，德国

参数关系性与可实现性、可约化性和逻辑关系一样，通常以带索引的形式表达。设我们使用一种类型论语言，其中包含一种称为HProp 的子单例类型，以及一种集合类型HSet。那么，谓词的类型为$A \to$ HProp，族的类型为$A \to$ HSet，关系的类型为$A \times A \to$ HProp，而相关关系有时称为对应，其类型为$A \times A \to$ HSet。索引形式中的所有构造也都可以通过带索引与纤维化对应，以纤维化的形式表达：存在一个

.....之间的等价关系$A \to$ HProp和 $\Sigma B : \text{HSet}$。$(B \xrightarrow{inj} A)$ 在证明无关情形下，以及.....之间$A \to$ HSet 和 $\Sigma B : \text{HSet}$。$(B \to A)$ 在证明相关情形下，对于$A : \text{HSet}$。也曾探讨过一种用于参数关系性的纤维方法，并建议对参数关系性进行迭代。

迭代参数性

一元索引形式的参数关系性内化在 [5] 中得到研究，并由此发展出一种迭代参数性的类型论。这种类型论可以在立方集合的“ 一元”形式 [3]中得到解释。按照标准定义，立方集合是立方范畴上的预层，可以视为处于迭代参数性的纤维一侧。[1] 这一点在 [13] 中特别受到强调；该文将立方集合解释为纤维形式的高维参数化。迭代二元参数化会产生半立方集合；此外也有人指出，迭代一元参数性会产生增广半单纯集合。[2]

元数	风格	迭代	相关性	逻辑结构
一元	带索引的	非迭代的	证明无关	谓词
		非迭代的	相关的	族
		迭代的	相关的	本工作
	纤维化的	非迭代的	证明无关	子对象
		非迭代的	相关的	纤维化
		迭代的	相关的	增广的半单纯集合
二元的	带索引的	非迭代的	证明无关	关系
		非迭代的	相关的	对应
		迭代的	相关的	本工作
	纤维化的	非迭代的	证明无关	纤维关系
		非迭代的	相关的	span
		迭代的	相关的	半立方集合

索引形式的迭代参数性

在 [14] 中给出了带索引形式的半单纯集合的非正式描述。后来提出了几种类型论中的形式化定义 [19, 9, 15, 1]。在先前工作中，提出了 ν-集合的概念，它推广了增广半单纯集合和半立方集合，并反映了

[1] 有趣的是，这与格罗滕迪克构造的情况形成对比：预层位于带索引的一侧，而纤维化位于纤维化的一侧[2] 作者从 Hugo Moeneclaey 和 Thorsten Altenkirch 那里了解到这一点

给出了 ν 元迭代参数性 [10],，并构造了带索引形式的 ν-集合。作为示例，对于 $\nu = 2$，使用的是相关关系族，而不是纤维形式的半立方集合。

$$X_0 : \mathsf{HSet} \begin{array}{c} \leftarrow \partial^R - \\ \leftarrow \partial^L - \end{array} X_1 : \mathsf{HSet} \begin{array}{c} \leftarrow \partial^{\star R} - \\ \leftarrow \partial^{\star L} - \\ \leftarrow \partial^{R\star} - \\ \leftarrow \partial^{L\star} - \end{array} X_2 : \mathsf{HSet} \qquad \ldots \qquad \text{(fibred)}$$

$$\begin{array}{ll} E_0 & : \mathsf{HSet} \\ E_1 & : E_0 \times E_0 \to \mathsf{HSet} \\ E_2 & : \Pi abcd.\, E_1(a,b) \times E_1(c,d) \times E_1(a,c) \times E_1(b,d) \to \mathsf{HSet} \\ \ldots & \end{array} \qquad \text{（相关关系）}$$

文中给出了以索引形式刻画 ν-集合的方程，并通过对涉及等式推理的大型结构进行归纳，证明了该构造的良基性。当前工作提出了一种改进的构造，展现了该构造的计算本质。

2 索引形式的迭代参数性的高层次描述

我们回顾先前工作，说明如何形式化定义一个类型，用以刻画序列 $E_0, E_1, \ldots$. 我们考虑的类型论包含 Σ 和 Π 类型、Martin-L¨ 的恒等类型 $=$、宇宙 U_m、单位类型 unit、自然数、自然数上的定义性证明无关不等式、流以及函数外延性。Π 和 Σ 的非依赖变体分别记作 $\to$ 和 $\times$。若 e 是 x $=$ y 的一个证明，p 是一个证明，则我们将 $\overrightarrow{\ }$e p 记作用 y 替换 p 所证明命题中的 x。unit 的典范元素记作 $\star$，满足 UIP：$\Pi xy : T.\Pi pq : x = y.p = q$ 的类型 $T : U_m$ 所构成的类型子宇宙记作 HSet_m。不等式的定义性证明无关性原则上并非必不可少，加入少量额外的传输后，可以用命题证明无关性取代。我们使用函数外延性，是因为将 n 元组表示为函数；若改用列表表示 n 元组，则可以避免使用函数外延性。

下面简要说明如何为元数 ν 定义索引形式的迭代参数性。E_{n+1} 对所有 E^p （$p \leq n$）的依赖，使得族 E 成为一个高度依赖的函数 [11]。表达这种结构的标准方式，是将其表示为一个流，其中第 (n + 1) 个元素 E_{n+1} 依赖于该流所有先前元素构成的束。我们将所有 E_0,、……、E_{n-1} 构成的束记作 $\nu\mathsf{Set}^{<}\mathrm{nm}$，并将剩余流 E_n、E_{n+1}、……记作 $\nu\mathsf{Set}^{\geq}\mathrm{nm}$ (D)；它依赖于由先前的 E_0,、……、E_{n-1} 构成的束 E：$\nu\mathsf{Set}^{<}\mathrm{nm}$。$E_n$ 的类型也依赖于束 D：$\nu\mathsf{Set}^{<}\mathrm{nm}$，记作 $\nu\mathsf{Set}^{=}\mathrm{nm}$ (D)，如表 1 和表 2 所示。当 n $= 0$ 时，$\nu\mathsf{Set}^{<0}\mathrm{m}$ 退化为单位类型。因此，$\nu\mathsf{Set}^{\geq 0}\mathrm{m}$ ($\star$) 表示整个流的类型。

类型 ν集合$^{=}$nm ($E_0, \ldots, E_{n-1}$) 表示一个族。它可以规范地表示为形式 fullframe($E_0, \ldots, E_{n-1}$) $\to$ HSet，其中 fullframe 汇集了 E_n 的所有依赖关系。其递归定义见表 3，使用了 frame 、 layer 和 painting。完整框架描述了标准形式（单纯形、立方体）的边界，我们将其分解为 n 层，每个维度对应一层，每层由 ν 绘画构成，其中一幅绘画对应一个填充的框架。请注意，类型 layer 依赖于框架限制算子 $\mathrm{restr}^{\mathrm{frame}}$；表4中为介于 1 和 ν之间的任意 ϵ 定义了该算子，而此限制算子又通过 $\mathrm{restr}^{\mathrm{layer}}$ 和 $\mathrm{restr}^{\mathrm{painting}}$定义。

XX:4 索引形式的迭代参数性的极度依赖递归结构

$\nu\mathrm{Set}_m$		:	HSet_{m+1}
$\nu\mathrm{Set}_m$		$\triangleq$	ν集合$_m^{\geq 0}(*)$
ν集合$_m^{\geq n}$	$(D\colon \nu$集合$_m^{<n})$	:	HSet_{m+1}
ν集合$_m^{\geq n}$	D	$\triangleq$	$\Sigma E:\ \nu$集合$_m^{=n}(D).\nu$集合$_m^{\geq n+1}(D,E)$

表 1 主要定义

ν集合$_m^{<n}$		:	HSet_{m+1}
ν集合$_m^{<0}$		$\triangleq$	unit
ν集合$^{<n'+1}{}_m$		$\triangleq$	$\Sigma D:\ \nu$集合$^{<n'}\,m\circ\ \nu$集合$^{=n'}\,m\,(D)$
ν集合$_m^{=n}$	$(D\colon \nu$集合$_m^{<n})$	:	HSet_{m+1}
ν集合$_m^{=n}$	D	$\triangleq$	完整框架$_m^n(D)\to\ \mathrm{HSet}_m$

表 2 截断的 ν-集合，核

完整框架$_m^n$	$(D\colon \nu$集合$_m^{<n})$	:	HSet_m
完整框架$_m^n$	D	$\triangleq$	框架$^{n,\ p}(D)$
框架$^{n,\ p,\ p\leq n}$	$(D\colon \nu$集合$_m^{<n})$	:	HSet_m
框架^{n,0}	D	$\triangleq$	unit
框架$^{n,p+1}$	D	$\triangleq$	$\Sigma d:$ 框架$^{n,p}(D)\circ$ 层$_{n-1}^{\ p}(d)$
层$^{n,p,p\leq n}$	$\{D:\ \nu$集合$_m^{<n+1}\}$ $(d\colon$ 框架$^{n+1,\ p}(D))$	:	HSet_m
层$^{n,\ p}$	$D\ d$	$\triangleq$	$\Pi\epsilon.$绘画$^{n,\ p}(D.2)($restr$^{n,\ p}_{\text{框架},\ p,\ \epsilon}(d))$
绘画$^{n,p,p\leq n}$	$(D\colon \nu$集合$_m^{<n})$ $(E\colon \nu$集合$_m^{=n}(D))$ $(d:$ 框架$^{n,\ p}(D))$	:	HSet_m
绘画$^{n,p,p=n}$	$D\ E\ d$	$\triangleq$	$E(d)$
绘画$^{n,p,p<n}$	$D\ E\ d$	$\triangleq$	$\Sigma l:$ 层$_{n-1}^{\ p}(d).$绘画$^{n,p+1}(E)(d,l)$

表3 框架、层和绘画

restr$^{n,p,q\leq n-p}_{\text{框架},\ q,\ \epsilon}$	$\{D:\ \nu$集合$_m^{<n+1}\}$ $(d\colon$ 框架$^{n+1,\ p}(D))$	:	框架$^{n,\ p}(D.1)$
restr$^{n,0}_{\text{框架},\ q,\ \epsilon}$	$D\ *$	$\triangleq$	$*$
restr$^{n,p+1}_{\text{框架},\ q,\ \epsilon}$	$D\ (d,l)$	$\triangleq$	$($restr$^{n,p}_{\text{框架},\ q+1,\ \epsilon}\ (d)\ ,$ restr$^{n-1,\ p}_{\text{层},\ q,\ \epsilon}\ (l))$
restr$^{n,p,q\leq n-p}_{\text{层},\ q,\ \epsilon}$	$\{D:\ \nu$集合$_m^{<n+2}\}$ $\{d:$ 框架$^{n+2,\ p}(D)\}$ $(l:$ 层$^{n+1,\ p}(d))$	:	层$^{n,\ p}\ ($restr$^{n+1,\ p}_{\text{框架},\ q+1,\ \epsilon}(d))$
restr$^{n,\ p}_{\text{层},\ q,\ \epsilon}$	$D\ d\ l$	$\triangleq$	$\lambda\omega_\circ\ ($restr$^{n,\ p}_{\text{绘画},q,\epsilon}(D.2)(l\omega))$
restr$^{n,p,q\leq n-p}_{\text{绘画},q,\epsilon}$	$\{D:\ \nu$集合$_m^{<n+1}\}$ $(E:\ \nu$集合$_m^{=n+1}(D))$ $\{d:$ 框架$^{n+1,\ p}(D)\}$ $(c:$ 绘画$^{n+1,\ p}(E)(d))$	:	绘画$^{n,\ p}(D.2)($restr$^{n,\ p}_{\text{框架},\ q,\ \epsilon}(d))$
restr$^{n,\ p}_{\text{绘画},0,\epsilon}$	$D\ E\ d(l,\ \_)$	$\triangleq$	l_ϵ
restr$^{n,p,p\leq n}_{\text{绘画},\ q+1,\ \epsilon}$	$D\ E\ d\ (l,\ c)$	$\triangleq$	$($restr$^{n-1,p}_{\text{层},\ q,\ \epsilon}\ (l)$, restr$^{n,p+1}_{\text{绘画},q,\epsilon}\ (E)(c))$

表4 q 阶 restr 的投影，或面

$\mathrm{coh}^{n}_{\text{框架},p,r\le q\le n-p,q,r,\epsilon,\omega}$	$\{D\colon \nu\text{集合}^{<n+2}_{m}\}$ $(d:\text{框架}^{n+2,p}(D))$	:	$\mathrm{restr}^{n,p}_{\text{框架},q,\epsilon}(\mathrm{restr}^{n+1,p}_{\text{框架},r,\omega}(d))$ $=\mathrm{restr}^{n,p}_{\text{框架},r,\omega}(\mathrm{restr}^{n+1,p}_{\text{框架},q+1,\epsilon}(d))$
$\mathrm{coh}^{n}_{\text{框架},q,r,\epsilon,\omega}0$	$D*$	$\triangleq$	自反性$(*)$
$\mathrm{coh}^{n,p}_{\text{框架},q,r,\epsilon,\omega}$	$D\,(d,l)$	$\triangleq$	$(\mathrm{coh}^{n,p}_{\text{框架},q+1,r+1,\epsilon,\omega}(d)\ ,\ \mathrm{coh}^{n-1,p}_{\text{层},q,r,\epsilon,\omega}(l))$
$\mathrm{coh}^{n}_{\text{层},\epsilon,\omega,q,r},p,r\le q\le n-p$	$\{D\colon \nu\text{集合}^{<n+3}_{m}\}$ $\{d\colon \text{框架}^{n+3,p}(D)\}$	:	$\mathrm{restr}^{n,p}_{\text{层},q,\epsilon}(\mathrm{restr}^{n+1,p}_{\text{层},r,\omega}(l))$
$\mathrm{coh}^{n,p}_{\text{层},\epsilon,\omega,q,r}$	$(l:\text{层}^{n+2,p}(d))$ $D\,d\,l$	$\triangleq$	$=\mathrm{restr}^{n,p}_{\text{层},r,\omega}(\mathrm{restr}^{n+1,p}_{\text{层},q+1,\epsilon}(l))$ $\lambda\theta_{\circ}\ \mathrm{coh}^{n,p}_{\text{绘画},\epsilon,\omega,q,r}(D.2)(l_{\theta})$
$\mathrm{coh}^{n}_{\text{绘画},p,r\le q\le n-p,q,r,\epsilon,\omega}$	$\{D\colon \nu\text{集合}^{<n+2}_{m}\}$ $(E\colon \nu\text{集合}^{=n+2}_{m}(D))$ $\{d\colon \text{框架}^{n+2,p}(D)\}$ $(c\colon \text{绘画}^{n+2,p}(E)(d))$	:	$\mathrm{restr}^{n,p}_{\text{绘画},q,\epsilon}(D.2)(\mathrm{restr}^{n+1,p}_{\text{painting},r,\omega}(E)(c))$ $=\mathrm{restr}^{n,p}_{\text{绘画},r,\omega}(D.2)(\mathrm{restr}^{n+1,p}_{\text{绘画},q+1,\epsilon}(E)(c))$
$\mathrm{coh}^{n,p}_{\text{绘画},q,p\le n^{0},\epsilon,\omega}$	$D\,E\,d(l,\_)$	$\triangleq$	自反性$(\mathrm{restr}^{n,p}_{\text{绘画},q,\epsilon}(D.2)(l_{\epsilon}))$
$\mathrm{coh}^{n,p}_{\text{绘画},q+1,r+1,\epsilon,\omega}$	$D\,E\,d\,(l,c)$	$\triangleq$	$(\mathrm{coh}^{n-1,p}_{\text{层},q,r,\epsilon,\omega}(l),\mathrm{coh}^{n,p}_{\text{绘画},q,r,\epsilon,\omega}(E)(c))$

表5 q 次投影与 r 次投影的交换，或相干条件

定义 $\mathrm{restr}^{\text{layer}}$ 依赖于一个表达两个 $\mathrm{restr}^{\text{frame}}$的复合运算交换性的等式。该交换性的证明在 [3] 表5中明确给出。它通过对 框架、层和 绘画的结构进行递归来完成。这正是 $\mathrm{coh}^{\text{frame}}$ 利用证明 $\mathrm{coh}^{\text{layer}}$ 和 $\mathrm{coh}^{\text{painting}}$所做的事情。注意，对于固定的 n，相干条件会计算为自反性证明，因此该构造所建立的是一个不再提及相干性的类型有效序列。

3 索引形式的迭代参数性的递归结构分析

表1至表5的构造的良基性，在先前工作中通过对一个涉及等式推理的大型结构进行归纳得到了论证。相反，我们将在本文中证明，可以避免等式推理，并且只需对n 进行归纳来构造以下结构：

框架^{n,p} : HSet 绘画^{n,p} : 框架^{n,p} →HSet $\mathrm{restr}^{n,p}_{\text{框架}}$: $\Pi q\ \le n\ -p_{\circ}\ \Pi\epsilon_{\circ}$ 框架$^{n+1,p}$ → 框架^{n,p} $\mathrm{restr}^{n,p}_{\text{绘画}}$: $\Pi q\ \le n\ -p_{\circ}\ \Pi\epsilon_{\circ}\ \Pi d$: 框架$^{n+1,p}{}_{\circ}$ 绘画$^{n+1,p}$(d) → 绘画^{n,p}($\mathrm{restr}^{n,p}_{\text{框架}}$(d)) $\mathrm{coh}^{n,p}_{\text{框架}}$: $\Pi qr\epsilon\omega_{\circ}$ $\mathrm{restr}^{n}_{\text{框架},q,\epsilon}{}^{\circ}$ $\mathrm{restr}^{n+1,p}_{\text{框架},\omega,r}$ $=\mathrm{restr}^{n,p}_{\text{框架},\omega,r}\circ\mathrm{restr}^{n}_{\text{框架},q+1,\epsilon}\,\mathrm{coh}^{n,p}_{\text{绘画}}$: $\Pi qr\epsilon\omega_{\circ}\ \Pi d$: 框架$^{n+2,p}{}_{\circ}$ $\mathrm{restr}^{n,p}_{\text{绘画},q,\epsilon}(\mathrm{restr}^{n+1,p}_{\text{绘画},r,\omega}(d))\circ\mathrm{restr}^{n+1,p}_{\text{绘画},r,\omega}(d)\ =\mathrm{restr}^{n,p}_{\text{绘画},r,\omega}(\mathrm{restr}^{n+1,p}_{\text{框架},q+1,\epsilon}(d))\circ\mathrm{restr}^{n+1,p}_{\text{绘画},q+1,\epsilon}(d)$

这种简洁性的代价，是将归纳假设的规约与归纳步骤的构造混在一起。首先，$\mathrm{restr}^{n,p}_{\text{框架}}$ 取决于框架$^{n+1,p}$:

[3] 不过，我们在表格中省略类型转换，就如同在外延类型论中工作一样。例如，$\mathrm{coh}^{\text{frame}}$ 在$\mathrm{restr}^{\text{layer}}$的定义中被隐式使用。关于相干性证明的使用位置，详见 [10]。

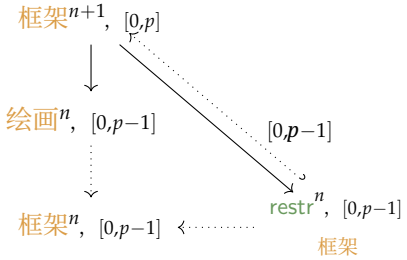
XX：6 索引形式的迭代参数关系性的依赖递归结构

它已经取决于能否构造出层级 $n+1$处结构的框架 分量。因此，指定层级 n 处所假设的结构，与据此定义层级 $n+1$处的结构，彼此相互依赖。我们必须按照特定顺序进行：先指定框架^{n}，据此可以指定绘画^{n}，再据此构造框架$^{n+1}$，最后据此指定$\mathrm{restr}^{n+1}_{\mathrm{frame}}$。现在，框架$^{n+1,p}$的定义通过对$p$进行归纳来给出，而它本身又依赖于$\mathrm{restr}^{n,\,p-1}$ 框架，其类型反过来又取决于框架$^{n+1,p-1}$。因此，必须相互定义框架$^{n+1,p}$ 以及$\mathrm{restr}^{n,\,p}$ 框架 的类型。

$\mathrm{frame}^{n+1,p}$ 的定义还通过 $\mathrm{layer}^{n,p-1}$依赖于 $\mathrm{painting}^{n,p-1}$。通过 $\mathrm{painting}^{n,p-1}$和 $\mathrm{restr}^{n,p-1}$ frame，它还因类型依赖而依赖于 $\mathrm{frame}^{n,p-1}$。因此，我们必须假设 $\mathrm{frame}^{n,p-1}$ 和 $\mathrm{painting}^{n,p-1}$。此外，$\mathrm{frame}^{n+1,p}$ 的定义依赖于 $\mathrm{frame}^{n+1,p-1}$，因此对于 $k < p$，必须已知 $\mathrm{frame}^{n,k}$、$\mathrm{painting}^{n,k}$和 $\mathrm{restr}^{n,k}$ frame。

我们将所有 $\mathrm{frame}^{n,k}$ （其中 $k < p$）的集合简写为 $\mathrm{frame}^{n,[0,p-1]}$，对$\mathrm{painting}^{n,[0,p-1]}$ 和 $\mathrm{restr}^{n,[0,p-1]}$ frame 也作类似处理。反过来，如果$\mathrm{frame}^{n,[0,p]}$ 是一个非空序列，我们将长度为p 的直接子序列记作$\mathrm{frame}^{n,[0,p-1]}$，并将序列的最后一个分量记作$\mathrm{frame}^{n,p}$。我们用$\mathrm{frame}^{n}$ 表示 $\mathrm{frame}^{n,[0,-1]}$。

示意地说，用虚线箭头表示类型中的依赖关系，用实线箭头表示定义中的依赖关系，可得到 $p \leq n+1$ 的依赖关系图如下。



特别是，框架$^{n+1,[0,p-1]}$在restr^{n}的类型中，$^{[0,p-1]}$ 框架与restr^{n}的依赖关系相结合，$^{[0,p-1]}$ 框架在框架$^{n+1}$的定义中，$^{[0,p]}$ 要求对restr^{n}的类型、$^{[0,p-1]}$ 框架以及框架$^{n+1}$进行互递归定义，^{p} 使其成为从 restr^{n}到$^{[0,p-1]}$ 框架的函数。

利用这个关于框架$^{n+1}$、$^{[0,p]}$ 和restr^{n}的$^{[0,p-1]}$ 框架递归定义，我们因此能够为该结构的规约添加第三个分量，适用于所有$p \leq n$。

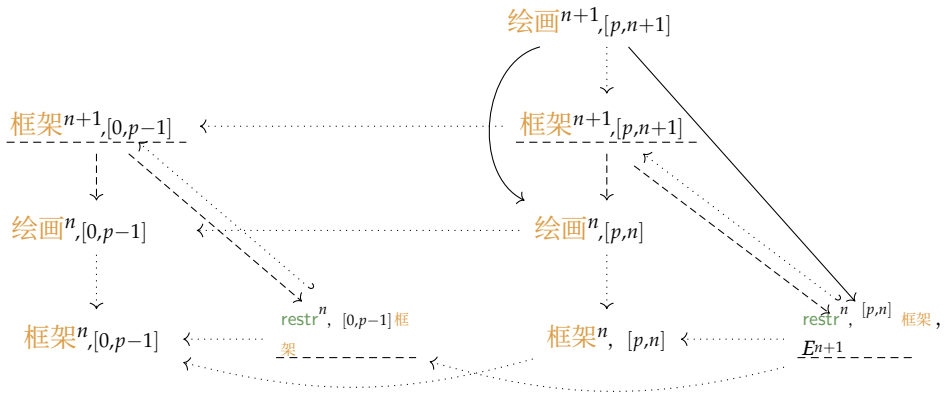
$$
\begin{array}{lcl}
\mathrm{frame}^{n,p} & : & \mathsf{HSet} \\
\mathrm{painting}^{n,p} & : & \mathrm{frame}^{n,p} \to \mathsf{HSet} \\
\mathrm{restr}^{n,p}_{\mathrm{frame}} & : & \Pi q \leq n-p.\,\Pi \epsilon.\,\mathrm{frame}^{n+1,p}\begin{pmatrix} \mathrm{frame}^{n,[0,p-1]} \\ \mathrm{painting}^{n,[0,p-1]} \\ \mathrm{restr}^{n,[0,p-1]}_{\mathrm{frame}} \end{pmatrix} \to \mathrm{frame}^{n,p}
\end{array}
$$

对于适当定义的框架$^{n,[0,p]}$，绘画$^{n,[0,p]}$和$\mathrm{restr}^{n,[0,p]}$ 框架，可以用序列来重新表述：

$$
\begin{array}{lcl}
\mathrm{frame}^{n,[0,p]} & : & \mathrm{FRAME}^{n,[0,p]} \\
\mathrm{painting}^{n,[0,p]} & : & \mathrm{PAINTING}^{n,[0,p]}(\mathrm{frame}^{n,[0,p]}) \\
\mathrm{restr}^{n,[0,p]}_{\mathrm{frame}} & : & \mathrm{restr}^{n,[0,p]}_{\mathrm{FRAME}}\begin{pmatrix} \mathrm{frame}^{n,[0,p-1]} \\ \mathrm{painting}^{n,[0,p-1]} \\ \mathrm{restr}^{n,[0,p-1]}_{\mathrm{frame}} \end{pmatrix}
\end{array}
$$

诸如$\mathrm{restr}^{n,[0,p]}_{\text{框架}}$这类类型将用迭代的 Σ-类型表示，因为每个$\mathrm{restr}^{n,p}_{\text{框架}}$都可能依赖于序列中先前的分量。对于框架和绘画，它们之间不存在依赖关系，因此可以用普通积表示$\text{框架}^{n,[0,p]}$和$\text{绘画}^{n,[0,p]}$。

现在来定义 $\mathrm{restr}^{n,p}$ painting。由于它依赖于 $\text{painting}^{n+1,p}$，我们首先需要定义 $\text{painting}^{n+1,p}$；但除了 $\text{frame}^{n+1,p}$ 中的类型依赖外，还需要通过类型依赖定义 $\text{painting}^{n,p-1}$、$\text{frame}^{n,p-1}$ 和 $\mathrm{restr}^{n,p-1}$ frame。由于当 $p < n+1$ 时，$\text{painting}^{n+1,p}$ 也依赖于 $\text{painting}^{n+1,p+1}$，这些依赖关系实际上存在于 $\text{painting}^{n+1,[p,n+1]}$、$\text{painting}^{n,[p,n]}$、$\text{frame}^{n+1,[p,n+1]}$、$\text{frame}^{n,[p,n]}$ 和 $\mathrm{restr}^{n,[p,n]}$ frame 之间。对于 $p = n$，这要求假设存在一个 E^{n+1}，其属于 ν集合$=\mathrm{n}^{+1}\mathrm{m}$。最后，$\text{frame}^{n+1,[p,n+1]}$ 依赖于 $\text{frame}^{n+1,[0,p-1]}$，$\text{frame}^{n+1,[p,n]}$ 依赖于 $\text{frame}^{n,[0,p-1]}$，而 $\mathrm{restr}^{n,[p,n]}$ frame 依赖于 $\mathrm{restr}^{n,[0,p-1]}$ frame，从而得到如下依赖关系图，其中定义中的间接依赖现在用虚线表示。



这最终足以指定类型 $\mathrm{restr}^{n,[0,p]}_{\text{绘画}}$ 的 $\mathrm{restr}^{n,[0,p]}_{\text{绘画}}$，因此我们可以专注于规约 $\mathrm{coh}^{n,p}_{\text{框架}}$。对 $\mathrm{restr}^{n,p}_{\text{框架}}$ 进行类似分析后，可以得出结论：$\mathrm{coh}^{n,[0,p]}_{\text{框架}}$ 应与 $\mathrm{restr}_{n+1,[0,p+1]}$ 框架 的定义相互配合完成。

接下来，$\mathrm{coh}^{n,p}_{\text{绘画}}$ 需要进行规约；为此，可以采用指定 $\mathrm{restr}^{n,p}_{\text{绘画}}$ 时所做的相同分析。具体而言，首先需要定义 $\mathrm{restr}_{n+1,\text{绘画}}^{[0,p]}$。

六个字段的规约完成后，就可以构造归纳步骤了。所有$\text{框架}^{n+1,[0,p]}$、$\text{绘画}^{n+1,[0,p]}$、$\mathrm{restr}_{n+1,\text{框架}}^{[0,p]}$ 以及$\mathrm{restr}_{n+1,\text{绘画}}^{[0,p]}$ 都已构造完成，因此只剩下构造$\mathrm{coh}_{n+1,\text{框架}}^{[0,p]}$ 和 $\mathrm{coh}_{n+1,\text{绘画}}^{[0,p]}$。具体如下。

3.1 指定框架

假设 $\text{框架}^{n,[0,p-1]}$ 对于 $p \le n+1$ 只需定义类型 $\text{FRAME}^{n,[0,p-1]}$ 它表示框架的集合，只是一个 p元 HSetHSet。

$\text{框架}^{n,[0,p-1]}$: HSet框架^{n} ,
单元$\text{框架}^{n,[0,p]}$, $\text{框架}^{n,[0,p-1]}$ ×HSet

3.2 指定绘画

类型 绘画$^{n,[0,p-1]}$ 的绘画$^{n,[0,p-1]}$ 对于$p \le n+1$ 是框架上的类型族的乘积，因此需要假设 框架$^{n,[0,p-1]}$。

$$\text{绘画}^{n,[0,p-1]}\ \text{框架}^{n,[0,p-1]} : \text{HSet}\ \text{绘画}^{n}_{[0,p-1]}(\text{框架}^{n,[0,p-1]}) \times (\text{框架}^{n,p} \to \text{HSet})$$

，单位 绘画$^{n,[0,p]}$ 框架$^{n,[0,p]}$，绘画^{n}，

3.3 定义框架并指定框架限制

假设对于 p $\le$ n $+1$，有框架$^{n,[0,p-1]}$ 和绘画$^{n,[0,p-1]}$，我们可以相互定义类型 restr$^{n,[0,p-1]}$ FRAME 与框架$^{n+1,[0,p]}$，后者是从 restr$^{n,[0,p-1]}$ 框架映射而来。我们引入缩写 DEPS$^{n,[0,p-1]}$ restr，表示框架$^{n,[0,p-1]}$、绘画$^{n,[0,p-1]}$和 restr$^{n,[0,p-1]}$ 框架序列的组合。

一旦假设存在 依赖$_{\text{restr}}$： DEPS$_{\text{restr}}$，我们就可以自由地引用 框架、绘画、restr$_{\text{frame}}$。

$$\text{DEPS}^{n,[0,p-1]}_{\text{restr}} \triangleq \begin{pmatrix} \text{frame}^{n,[0,p-1]} & : & \text{FRAME}^{n,[0,p-1]} \\ \text{painting}^{n,[0,p-1]} & : & \text{PAINTING}^{n,[0,p-1]}(\text{frame}^{n,[0,p-1]}) \\ \text{restr}^{n,[0,p-1]}_{\text{frame}} & : & \text{restr}^{n,[0,p-1]}_{\text{FRAME}} \begin{pmatrix} \text{frame}^{n,[0,p-1]} \\ \text{painting}^{n,[0,p-1]} \end{pmatrix} \end{pmatrix}$$

$$\text{frame}^{n+1,[0,p]}\ \text{deps}^{n,[0,p-1]}_{\text{restr}} : \text{FRAME}^{n+1,[0,p]}$$

$$\text{frame}^{n+1,[0,0]}\ \text{deps}^{n}_{\text{restr}} \triangleq (*, \text{unit})$$

$$\text{frame}^{n+1,[0,p+1]}\ \text{deps}^{n,[0,p]}_{\text{restr}} \triangleq \begin{pmatrix} \text{frame}^{n+1,[0,p]}(\text{deps}^{n,[0,p-1]}_{\text{restr}}), \\ \Sigma d : \text{frame}^{n+1,p}(\text{deps}^{n,[0,p-1]}_{\text{restr}}).\ \text{layer}^{n,p}(\text{deps}^{n,[0,p]}_{\text{restr}})(d) \end{pmatrix}$$

$$\text{restr}^{n,[0,p-1]}_{\text{FRAME}} \begin{pmatrix} \text{frame}^{n,[0,p-1]} \\ \text{painting}^{n,[0,p-1]} \end{pmatrix} : \text{HSet}$$

$$\text{restr}^{n}_{\text{FRAME}} \begin{pmatrix} * \\ * \end{pmatrix} \triangleq \text{unit}$$

$$\text{restr}^{n,[0,p]}_{\text{FRAME}} \begin{pmatrix} \text{frame}^{n,[0,p]} \\ \text{painting}^{n,[0,p]} \end{pmatrix} \triangleq \begin{cases} \Sigma \text{restr}^{n,[0,p-1]}_{\text{frame}} : \text{restr}^{n,[0,p-1]}_{\text{FRAME}} \begin{pmatrix} \text{frame}^{n,[0,p-1]} \\ \text{painting}^{n,[0,p-1]} \end{pmatrix}. \\ \Pi q \le n-p.\, \Pi \epsilon. \\ \text{frame}^{n+1,p} \begin{pmatrix} \text{frame}^{n,[0,p-1]} \\ \text{painting}^{n,[0,p-1]} \\ \text{restr}^{n+1,[0,p-1]}_{\text{frame}} \end{pmatrix} \to \text{frame}^{n,p} \end{cases}$$

$$\text{LAYER}^{n,p}\ \text{deps}^{n,[0,p]}_{\text{restr}} : \text{HSet}$$

$$\text{LAYER}^{n,p}\ \text{deps}^{n,[0,p]}_{\text{restr}} \triangleq \text{frame}^{n+1,p}(\text{deps}^{n,[0,p-1]}_{\text{restr}}) \to \text{HSet}$$

$$\text{layer}^{n,p}\ \text{deps}^{n,[0,p]}_{\text{restr}} : \text{LAYER}^{n,p}(\text{deps}^{n,[0,p]}_{\text{restr}})$$

$$\text{layer}^{n,p}\ \text{deps}^{n,[0,p]}_{\text{restr}} \triangleq \lambda d.\, \Pi \epsilon.\, \text{painting}^{n,p}(\text{restr}^{n,p}_{\text{frame},0,\epsilon}(d))$$

3.4 定义绘画

为了定义 绘画$^{n+1,\ p}$，我们引入另一个缩写，它将一个完整的 依赖$^{n,\ [0,n]}_{\text{restr}}$： DEPS$^{n,\ [0,n]}_{\text{restr}}$ 与族 E： 框架$^{n+1,\ n+1}$(依赖$^{n,\ [0,n]}_{\text{restr}}$) $\to$ HSet。

$$\text{DEPS}^{n}_{\text{fullrestr}} \triangleq \begin{pmatrix} \text{deps}^{n,[0,n]}_{\text{restr}} & : & \text{DEPS}^{n,[0,n]}_{\text{restr}} \\ E & : & \text{frame}^{n+1,n+1}(\text{deps}^{n,[0,n]}_{\text{restr}}) \to \text{HSet} \end{pmatrix}$$

那么，绘画$^{n+1,p}$ 是通过从基例 $n+1$开始递归，并递归至 p，为 $p \le n+1$ 定义的。

$$\text{绘画}^{n+1,p}\quad \text{deps}^{n}_{\text{fullrestr}}\quad : \text{绘画}^{n+1,p}\text{绘画}^{n+1,n+1}\quad \text{deps}^{n}_{\text{fullrestr}}\quad , \mathrm{E}\ \text{绘画}^{n+1,p\le n}\ \text{deps}^{n}_{\text{fullrestr}},\ \lambda \mathrm{d}.\ \Sigma \mathrm{l} : \text{层}^{n,p}(\text{deps}^{n,[0,p]}\ \text{restr}\ ).\text{绘画}^{n+1,p+1}(\text{deps}^{n}_{\text{fullrestr}})(\mathrm{d},\mathrm{l})$$

由此，我们可以推导出 $\text{绘画}^{n+1,[0,p-1]}$ 对于 $p \le n+2$，通过第二次递归，这次递归针对 p。

$$\text{绘画}^{n+1,[0,p-1]}\text{依赖}^{n}_{\text{fullrestr}}\quad : \text{绘画}^{n+1,[0,p-1]}(\text{框架}^{n+1,[0,p-1]}(\text{依赖}^{n,[0,p-1]}\quad \text{限制}\ ))\ \text{绘画}^{n+1}\quad \text{依赖}^{n}_{\text{fullrestr}}\quad , \text{单位}\ \text{绘画}^{n+1,[0,p]}\quad \text{依赖}^{n}_{\text{fullrestr}}\quad , (\text{绘画}^{n+1,[0,p-1]}(\text{依赖}^{n}_{\text{fullrestr}}),\text{绘画}^{n+1,p}(\text{依赖}^{n}_{\text{fullrestr}}))$$

3.5 指定绘制限制

要进一步推进，我们需要指定绘画限制序列的类型，对于 $p \le n+1$。

$$\begin{array}{lll} \text{限制}^{n,[0,p-1]}\ \text{绘画} & \text{deps}^{n}_{\text{fullrestr}} & : \quad \mathrm{HSet} \\ \text{restr}^{n}_{\text{绘画}} & \text{deps}^{n}_{\text{fullrestr}} & \triangleq \quad \mathrm{unit} \\ \text{restr}^{n,[0,p]}_{\text{绘画}} & \text{deps}^{n}_{\text{fullrestr}} & \triangleq \quad \text{restr}^{n,[0,p-1]}_{\text{绘画}} \times \left(\begin{array}{l} \Pi q \le n-p.\Pi \epsilon.\Pi d : \text{框架}^{n+1,p}(\text{deps}^{n,[0,p-1]}\ \text{restr}\ ). \\ \text{绘画}^{n+1,p}(\text{依赖项}^{n}\quad)(d) \to \\ \text{绘画}^{n,p}(\text{restr}^{n,p}_{\text{框架}_{q,\epsilon}}\ d)) \end{array} \right) \end{array}$$

3.6 定义框架限制并指定框架相干性定律

现在，我们可以相互定义 $\text{restr}^{n+1,[0,p]}$ frame，并根据 $\text{frame}^{n,[0,n]}$、$\text{painting}^{n,[0,n]}$、$\text{restr}^{n,[0,n]}$ frame、$\text{restr}^{n,[0,p-1]}$ painting 以及某个 $\mathrm{E} : \text{frame}^{n+1,n+1}(\text{deps}^{n,[0,n]}\ \text{restr}\ ) \to \mathrm{HSet}$ 来指定 $\text{coh}^{n,[0,p-1]}$ FRAME。我们将这些依赖关系以及 $\text{coh}^{n,[0,p-1]}$ frame 中的依赖关系缩写为 $\text{DEPS}^{n,[0,p-1]}$ coh，并在行文中将 $\text{restr}^{n,p}$ layer 和 $\text{restr}^{n,p}$ LAYER 简写。根据 $\text{deps}^{n,[0,p-1]}$ coh 构造 $\text{deps}^{n+1,[0,p]}$ restr 更为方便，这样我们现在就能进一步构造 $\text{frame}^{n+2,p}$。

$$
\begin{array}{lll}
\mathrm{DEPS}^{n,[0,p-1]}_{\mathrm{coh}} & \triangleq & \begin{pmatrix} \mathrm{deps}^{n}_{\mathrm{fullrestr}} & : & \mathrm{DEPS}^{n}_{\mathrm{fullrestr}} \\ \mathrm{restr}^{n,[0,p-1]}_{\mathrm{painting}} & : & \mathrm{restr}^{n,[0,p-1]}_{\mathrm{PAINTING}}(\mathrm{deps}^{n}_{\mathrm{fullrestr}}) \\ \mathrm{coh}^{n,[0,p-1]}_{\mathrm{frame}} & : & \mathrm{coh}^{n,[0,p-1]}_{\mathrm{FRAME}}\begin{pmatrix} \mathrm{deps}^{n}_{\mathrm{fullrestr}} \\ \mathrm{restr}^{n,[0,p-1]}_{\mathrm{painting}} \end{pmatrix} \end{pmatrix} \\
\overline{\mathrm{deps}_{\mathrm{restr}}} \quad \mathrm{deps}^{n,[0,p-1]}_{\mathrm{coh}} & : & \mathrm{DEPS}^{n+1,[0,p]}_{\mathrm{restr}} \\
\overline{\mathrm{deps}_{\mathrm{restr}}} \quad \mathrm{deps}^{n,[0,p-1]}_{\mathrm{coh}} & \triangleq & \begin{pmatrix} \mathrm{frame}^{n+1,[0,p]}(\mathrm{deps}^{n,[0,p-1]}_{\mathrm{restr}}) \\ \mathrm{painting}^{n+1,[0,p]}(\mathrm{deps}^{n}_{\mathrm{fullrestr}}) \\ \mathrm{restr}^{n+1,[0,p]}_{\mathrm{frame}}(\mathrm{deps}^{n,[0,p-1]}_{\mathrm{coh}}) \end{pmatrix} \\
\mathrm{restr}^{n+1,[0,p]}_{\mathrm{frame}} \quad \mathrm{deps}^{n,[0,p-1]}_{\mathrm{coh}} & : & \mathrm{restr}^{n+1,[0,p]}_{\mathrm{FRAME}}\begin{pmatrix} \mathrm{frame}^{n+1,[0,p]}(\mathrm{deps}^{n,[0,p]}_{\mathrm{restr}}) \\ \mathrm{painting}^{n+1,[0,p]}(\mathrm{deps}^{n}_{\mathrm{fullrestr}}) \end{pmatrix} \\
\mathrm{restr}^{n+1,[0,0]}_{\mathrm{frame}} \quad \mathrm{deps}^{n}_{\mathrm{coh}} & \triangleq & (*, \lambda q.\lambda\epsilon.\lambda\_.*) \\
\mathrm{restr}^{n+1,[0,p+1]}_{\mathrm{frame}} \quad \mathrm{deps}^{n,[0,p]}_{\mathrm{coh}} & \triangleq & \begin{pmatrix} \mathrm{restr}^{n+1,[0,p]}_{\mathrm{frame},q,\epsilon}(\mathrm{deps}^{n,[0,p-1]}_{\mathrm{coh}}), \lambda q.\lambda\epsilon.\lambda(d,l). \\ \left(\mathrm{restr}^{n+1,p}_{\mathrm{frame},q,\epsilon}(\mathrm{deps}^{n,[0,p-1]}_{\mathrm{coh}})(d), \mathrm{restr}^{n,p}_{\mathrm{layer},q,\epsilon}(\mathrm{deps}^{n,[0,p]}_{\mathrm{coh}})(d)(l)\right) \end{pmatrix} \\
\mathrm{coh}^{n,[0,p-1]}_{\mathrm{FRAME}}\begin{pmatrix} \mathrm{deps}^{n}_{\mathrm{fullrestr}} \\ \mathrm{restr}^{n,[0,p-1]}_{\mathrm{painting}} \end{pmatrix} & : & \mathrm{HSet} \\
\mathrm{coh}^{n}_{\mathrm{FRAME}}\begin{pmatrix} \mathrm{deps}^{n}_{\mathrm{fullrestr}} \\ * \end{pmatrix} & \triangleq & \mathrm{unit} \\
\mathrm{coh}^{n,[0,p]}_{\mathrm{FRAME}}\begin{pmatrix} \mathrm{deps}^{n}_{\mathrm{fullrestr}} \\ \mathrm{restr}^{n,[0,p]}_{\mathrm{painting}} \end{pmatrix} & \triangleq & \begin{cases} \Sigma\mathrm{coh}^{n,[0,p-1]}_{\mathrm{frame},q,r,\epsilon,\omega} : \mathrm{coh}^{n,[0,p-1]}_{\mathrm{FRAME}}\begin{pmatrix} \mathrm{deps}^{n}_{\mathrm{fullrestr}} \\ \mathrm{restr}^{n,[0,p-1]}_{\mathrm{painting}} \end{pmatrix}. \\ \Pi d : \mathrm{frame}^{n+2,p}\left(\overline{\mathrm{deps}_{\mathrm{restr}}}\begin{pmatrix} \mathrm{deps}^{n}_{\mathrm{fullrestr}} \\ \mathrm{restr}^{n,[0,p-1]}_{\mathrm{painting}} \\ \mathrm{coh}^{n,[0,p-1]}_{\mathrm{frame}} \end{pmatrix}\right). \\ \mathrm{restr}^{n,p}_{\mathrm{frame},q,\epsilon}(\mathrm{restr}^{n+1,p}_{\mathrm{frame},r,\omega}\begin{pmatrix} \mathrm{deps}^{n}_{\mathrm{fullrestr}} \\ \mathrm{restr}^{n,[0,p-1]}_{\mathrm{painting}} \\ \mathrm{coh}^{n,[0,p-1]}_{\mathrm{frame}} \end{pmatrix}(d)) = \\ \mathrm{restr}^{n,p}_{\mathrm{frame},r,\omega}(\mathrm{restr}^{n+1,p}_{\mathrm{frame},q+1,\epsilon}\begin{pmatrix} \mathrm{deps}^{n}_{\mathrm{fullrestr}} \\ \mathrm{restr}^{n,[0,p-1]}_{\mathrm{painting}} \\ \mathrm{coh}^{n,[0,p-1]}_{\mathrm{frame}} \end{pmatrix}(d)) \end{cases} \\
\mathrm{restr}^{n,p}_{\mathrm{LAYER}} \quad \mathrm{deps}^{n,[0,p]}_{\mathrm{coh}} & : & \mathrm{HSet} \\
\mathrm{restr}^{n,p}_{\mathrm{LAYER}} \quad \mathrm{deps}^{n,[0,p]}_{\mathrm{coh}} & \triangleq & \begin{cases} \Pi q \le n-p.\Pi\epsilon.\Pi d : \mathrm{frame}^{n+2,p}(\overline{\mathrm{deps}_{\mathrm{restr}}}(\mathrm{deps}^{n,[0,p-1]}_{\mathrm{coh}})). \\ \mathrm{layer}^{n+1,p}(\overline{\mathrm{deps}_{\mathrm{restr}}}(\mathrm{deps}^{n,[0,p-2]}_{\mathrm{coh}})(d) \to \\ \mathrm{layer}^{n,p}(\mathrm{deps}^{n,[0,p]}_{\mathrm{restr}})(\mathrm{restr}^{n,p}_{\mathrm{frame}}(\mathrm{deps}^{n,[0,p-1]}_{\mathrm{coh}})(d)) \end{cases} \\
\mathrm{restr}^{n,p}_{\mathrm{layer}} \quad \mathrm{deps}^{n,[0,p]}_{\mathrm{coh}} & : & \mathrm{restr}^{n,p}_{\mathrm{LAYER}}(\mathrm{deps}^{n,[0,p]}_{\mathrm{coh}}) \\
\mathrm{restr}^{n,p}_{\mathrm{layer},q,\epsilon} \quad \mathrm{deps}^{n,[0,p]}_{\mathrm{coh}} & \triangleq & \lambda d.\lambda l.\lambda\omega.\overrightarrow{\mathrm{coh}^{n,p}_{\mathrm{frame},q,0,\epsilon,\omega}(d)} \\
 & & \mathrm{restr}^{n,p}_{\mathrm{painting},q,\epsilon}(\mathrm{restr}^{n,p}_{\mathrm{frame},q,\epsilon}(\mathrm{deps}^{n,[0,p-1]}_{\mathrm{coh}})(d))(l_\omega)
\end{array}
$$

3.7 定义绘制限制

有了 $\mathrm{restr}^{n,[0,p]}_{n+1}$ frame 和 $\mathrm{coh}^{n,[0,p-1]}$ FRAME，也就有了 $\mathrm{restr}^{n,[0,p-1]}$ layer，因此可以定义 $\mathrm{restr}^{n,p}_{n+1}$ painting。首先，我们在相干条件层级定义完全依赖的概念，它将完全的 $\mathrm{deps}^{n,[0,n]}$ coh 与集合 $E' : \mathrm{frame}^{n+2,n+2}(\mathrm{deps}_{\mathrm{restr}}(\mathrm{deps}^{n,[0,n]}, \cancel{\mathrm{coh}})) \to \mathrm{HSet}$ 结合起来。

$$
\mathrm{DEPS}^{n}_{\mathrm{fullcoh}} \triangleq \begin{pmatrix} \mathrm{deps}^{n,[0,n]}_{\mathrm{coh}} & : & \mathrm{DEPS}^{n,[0,n]}_{\mathrm{coh}} \\ E' & : & \mathrm{frame}^{n+2,n+2}(\overline{\mathrm{deps}_{\mathrm{restr}}}(\mathrm{deps}^{n,[0,n]}_{\mathrm{coh}})) \to \mathrm{HSet} \end{pmatrix}
$$

然后我们可以扩展 $\overline{\mathrm{deps}_{\mathrm{restr}}}$，构造 $\mathrm{deps}_{\mathrm{fullrestr}}$，从 $\mathrm{deps}_{\mathrm{fullcoh}}$。

$$\begin{array}{lll} \overline{\mathsf{deps}_{\mathsf{fullrestr}}}\ \mathsf{deps}^{n}_{\mathsf{fullcoh}} & : & \mathsf{DEPS}^{n+1}_{\mathsf{fullrestr}} \\ \overline{\mathsf{deps}_{\mathsf{fullrestr}}}\ \mathsf{deps}^{n}_{\mathsf{fullcoh}} & \triangleq & \left(\begin{array}{l} \overline{\mathsf{deps}_{\mathsf{restr}}}(\mathsf{deps}^{n,[0,n]}_{\mathsf{coh}}) \\ E' \end{array}\right) \end{array}$$

接着，我们可以定义 $\mathsf{restr}^{n+1,\ p}_{\text{以递归方式绘制}}$ for $p \leq n+1$，对 $n+1-p$ 进行递归。

$$\begin{array}{lll} \mathsf{restr}^{n+1,p}_{\mathsf{painting},q,\epsilon}\ \mathsf{deps}^{n}_{\mathsf{fullcoh}} & : & \left(\begin{array}{l} \Pi d : \mathsf{frame}^{n+2,p}(\overline{\mathsf{deps}_{\mathsf{restr}}}(\mathsf{deps}^{n,[0,p-2]}_{\mathsf{coh}})). \\ \mathsf{painting}^{n+2,p}(\overline{\mathsf{deps}_{\mathsf{fullrestr}}}(\mathsf{deps}^{n}_{\mathsf{fullcoh}}))(d) \to \\ \mathsf{painting}^{n+1,p}(\mathsf{deps}^{n}_{\mathsf{fullrestr}})(\mathsf{restr}^{n+1,p}_{\mathsf{frame}}(\mathsf{deps}^{n,[0,p-1]}_{\mathsf{coh}})(d)) \end{array}\right) \\ \mathsf{restr}^{n+1,n+1}_{\mathsf{painting},q,\epsilon}\ \mathsf{deps}^{n}_{\mathsf{fullcoh}} & \triangleq & \lambda d.\lambda(l,c).l_{\epsilon} \\ \mathsf{restr}^{n+1,p\leq n}_{\mathsf{painting},q,\epsilon}\ \mathsf{deps}^{n}_{\mathsf{fullcoh}} & \triangleq & \lambda d.\lambda(l,c).\left(\begin{array}{l} \mathsf{restr}^{n,p}_{\mathsf{layer},q,\epsilon}(\mathsf{deps}^{n,[0,p]}_{\mathsf{coh}})(d)(l), \\ \mathsf{restr}^{n+1,p+1}_{\mathsf{painting},q,\epsilon}(\mathsf{deps}^{n}_{\mathsf{fullcoh}})(d,l)(c) \end{array}\right) \end{array}$$

由此，我们可以通过第二次递归再次推导出 $\mathsf{restr}^{n+1,\ [0,p-1]}_{\text{绘画}}$，其中 $p \leq n+2$，这次递归是针对 p 进行的。

$$\begin{array}{lll} \mathsf{restr}^{n+1,[0,p-1]}_{\mathsf{painting}}\ \mathsf{deps}^{n}_{\mathsf{fullcoh}} & : & \mathsf{restr}^{n+1,[0,p-1]}_{\mathsf{PAINTING}}(\mathsf{deps}^{n}_{\mathsf{fullrestr}}) \\ \mathsf{restr}^{n+1}_{\mathsf{painting}}\ \mathsf{deps}^{n}_{\mathsf{fullcoh}} & \triangleq & \mathsf{unit} \\ \mathsf{restr}^{n+1,[0,p]}_{\mathsf{painting}}\ \mathsf{deps}^{n}_{\mathsf{fullcoh}} & \triangleq & (\mathsf{restr}^{n+1,[0,p-1]}_{\mathsf{painting}}(\mathsf{deps}^{n}_{\mathsf{fullcoh}}), \mathsf{restr}^{n+1,p}_{\mathsf{painting}}(\mathsf{deps}^{n}_{\mathsf{fullcoh}})) \end{array}$$

3.8 指定绘画相干律

进一步来说，我们需要指定绘画相干律的类型 $\mathsf{coh}^{n,\ [0,p-1]}_{\text{绘画}}$，其中 $p \leq n+1$。

$$\begin{array}{lll} \mathsf{coh}^{n,\ [0,p-1]}_{\text{绘画}}\ \mathsf{deps}^{n}_{\mathsf{fullcoh}} & : & \mathsf{HSet} \\ \mathsf{coh}^{n}_{\text{绘画}}\ * & \triangleq & \mathsf{unit} \\ \mathsf{coh}^{n}_{\text{绘画}}\ \mathsf{deps}^{n}_{\mathsf{fullcoh}} & \triangleq & \mathsf{coh}^{n,\ [0,p-1]}_{\text{绘画}}(\text{依赖}^{n}_{\text{完全相干}})\times \\ \multicolumn{3}{l}{\left\{\begin{array}{l} \Pi q \leq n-p_{\circ}\ \Pi r \leq q_{\circ}\ \Pi\epsilon_{\circ}\ \Pi\omega_{\circ} \\ \Pi d : \text{框架}^{n+2,\ p}(\overline{\text{依赖}^{n}_{\mathsf{restr}}}(\text{依赖}^{n,\ [0,p-2]}_{\mathsf{coh}}))_{\circ}\ \Pi c : \text{绘画}^{n+2,\ p}(\overline{\text{依赖}_{\text{完整restr}}}(\text{依赖}^{n}_{\text{完整coh}})). \\ \mathsf{restr}^{n,\ p}(\mathsf{restr}^{n+1,\ p}_{\text{框架}}(\mathsf{deps}^{n,\ [0,p-1]}_{\mathsf{coh}})(d))(\mathsf{restr}^{n+1,\ p}_{\text{绘画}}(\text{依赖}^{n}_{\text{完整相干性}})(d)(c)) = \mathsf{coh}^{n,p}_{\text{框架},\ q,\ r,\ \epsilon,\ \omega} \\ \mathsf{restr}^{n,\ p}_{\text{绘画},\ q,\ \epsilon}(\mathsf{restr}^{n+1,\ p}_{\text{框架}}(\text{依赖}^{n,\ [0,p-1]}_{\mathsf{coh}})(d))(\mathsf{restr}^{n+1,\ p}_{\text{绘画},\ r,\ \omega}(\text{依赖}^{n}_{\text{完全相干性}})(d)(c)) \end{array}\right.} \end{array}$$

其中，我们使用“关于等式”的记号 $t =_e u$ 表示 $\overrightarrow{e}(t) = u$；当 t 属于某个类型 $P(a)$，u 属于类型 $P(b)$，且 e 是 $a = b$ 的证明时，均如此。

3.9 定义框架相干律并指定框架二维相干律

借助 $\mathsf{coh}^{n,[0,p-1]}$ PAINTING，我们可以相互定义 $\mathrm{p} \leq \mathrm{n}+1$ 中的 $\mathsf{coh}_{n+1}^{,[0,p]}$ frame 和 $\mathsf{coh2}^{n,[0,p-1]}$ FRAME，它们来自 $\mathsf{deps}^{n}_{\mathsf{fullcoh}}$。由于推理是在 HSet 中进行的，我们不需要让 $\mathsf{coh}_{n+1}^{,[0,p]}$ frame 依赖于 $\mathsf{coh2}^{n,[0,p-1]}$ FRAME 的证明，因为后者可由恒等证明的唯一性证明。只需考虑 $\mathsf{coh2}^{n,p}$ FRAME，而不是 $\mathsf{coh2}^{n,[0,p-1]}$ FRAME。

同样方便的是构造 $\text{依赖}^{n+1,\ [0,p]}_{\text{coh}}$ 从 $\text{依赖}^{n,\ [0,p-1]}_{\text{coh2}}$，这样我们现在就可以构造 $\text{框架}^{n+3,p}_{\circ}$。我们在 $\mathsf{coh}_{n+1}^{,[0,p]}$ 框架 中隐式地使用了这一点： Σ-类型中的相等等价于底层相等的Σ-类型。

XX:12 索引形式的迭代参数关系性的高度依赖递归结构

$$
\begin{array}{lll}
\mathrm{DEPS}^{n,[0,p-1]}_{\mathrm{coh2}} & \triangleq & \begin{pmatrix} \mathrm{deps}^{n}_{\mathrm{fullcoh}} & : & \mathrm{DEPS}^{n}_{\mathrm{fullcoh}} \\ \mathrm{coh}^{n,[0,p-1]}_{\mathrm{painting}} & : & \mathrm{coh}^{n,[0,p-1]}_{\mathrm{PAINTING}}(\mathrm{deps}^{n}_{\mathrm{fullcoh}}) \end{pmatrix} \\
\overline{\mathrm{deps}_{\mathrm{coh}}}\ \mathrm{deps}^{n,[0,p-1]}_{\mathrm{coh2}} & : & \mathrm{DEPS}^{n+1,[0,p]}_{\mathrm{coh}} \\
\overline{\mathrm{deps}_{\mathrm{coh}}}\ \mathrm{deps}^{n,[0,p-1]}_{\mathrm{coh2}} & \triangleq & \begin{pmatrix} \overline{\mathrm{deps}_{\mathrm{fullrestr}}}(\mathrm{deps}^{n}_{\mathrm{fullcoh}}) \\ \mathrm{restr}^{n+1,p}_{\mathrm{painting}}(\mathrm{deps}^{n}_{\mathrm{fullcoh}}) \\ \mathrm{coh}^{n+1,p}_{\mathrm{frame}}(\mathrm{deps}^{n,[0,p-1]}_{\mathrm{coh2}}) \end{pmatrix} \\
\mathrm{coh}^{n+1,[0,p]}_{\mathrm{frame}}\ \mathrm{deps}^{n,[0,p-1]}_{\mathrm{coh2}} & : & \mathrm{coh}^{n+1,[0,p]}_{\mathrm{FRAME}} \begin{pmatrix} \overline{\mathrm{deps}_{\mathrm{fullrestr}}}(\mathrm{deps}^{n}_{\mathrm{fullcoh}}) \\ \mathrm{restr}^{n+1,[0,p]}_{\mathrm{painting}}(\mathrm{deps}^{n}_{\mathrm{fullcoh}}) \end{pmatrix} \\
\mathrm{coh}^{n+1,[0,0]}_{\mathrm{frame}}\ \mathrm{deps}^{n}_{\mathrm{coh2}} & \triangleq & (*, \lambda q.\lambda r.\lambda \epsilon.\lambda \omega.\lambda \_.\mathsf{refl}) \\
\mathrm{coh}^{n+1,[0,p+1]}_{\mathrm{frame}}\ \mathrm{deps}^{n,[0,p]}_{\mathrm{coh2}} & \triangleq & \begin{pmatrix} \mathrm{coh}^{n+1,[0,p]}_{\mathrm{frame}}(\mathrm{deps}^{n,[0,p-1]}_{\mathrm{coh2}}), \\ \lambda q.\lambda r.\lambda \epsilon.\lambda \omega.\lambda (d,l). \\ (\mathrm{coh}^{n+1,p}_{\mathrm{frame},q,r,\epsilon,\omega}(\mathrm{deps}^{n,[0,p]}_{\mathrm{coh2}})(d), \mathrm{coh}^{n,p}_{\mathrm{layer},q,r,\epsilon,\omega}(\mathrm{deps}^{n,[0,p]}_{\mathrm{coh2}})(d)(l)) \end{pmatrix} \\
\mathrm{coh2}^{n,p}_{\mathrm{frame}}\ \mathrm{deps}^{n}_{\mathrm{fullcoh}} & : & \mathrm{coh2}^{n,p}_{\mathrm{FRAME}}(\mathrm{deps}^{n}_{\mathrm{fullcoh}}) \\
\mathrm{coh2}^{n,p}_{\mathrm{frame},q,r,\epsilon,\omega,\theta}\ \mathrm{deps}^{n}_{\mathrm{fullcoh}} & \triangleq & \lambda d.\mathsf{UIP} \\
\mathrm{coh}^{n,p}_{\mathrm{layer}}\ \mathrm{deps}^{n,[0,p]}_{\mathrm{coh2}} & : & \begin{cases} \Pi q \le n-p.\Pi r \le q.\Pi \epsilon.\Pi \omega.\Pi d.\Pi l. \\ \mathrm{restr}^{n,p}_{\mathrm{layer},q,\epsilon}(\mathrm{deps}^{n,[0,p]}_{\mathrm{coh}})(\mathrm{restr}^{n+1,p}_{\mathrm{layer},r,\omega}(\overline{\mathrm{deps}_{\mathrm{coh}}}(\mathrm{deps}^{n,[0,p-2]}_{\mathrm{coh2}}))(l)) \\ =_{\mathrm{coh}^{n+1,p}_{\mathrm{frame},r+1,q+1,\epsilon,\omega}(\mathrm{deps}^{n,[0,p-1]}_{\mathrm{coh2}})(d)} \\ \mathrm{restr}^{n,p}_{\mathrm{layer},r,\omega}(\overrightarrow{\mathrm{deps}_{\mathrm{coh}}}(\mathrm{deps}^{n,[0,p]}_{\mathrm{coh}}))(\mathrm{restr}^{n+1,p}_{\mathrm{layer},q+1,\epsilon}(\overline{\mathrm{deps}_{\mathrm{coh}}}(\mathrm{deps}^{n,[0,p-2]}_{\mathrm{coh2}}))(l)) \end{cases} \\
\mathrm{coh}^{n,p}_{\mathrm{layer},q,r,\epsilon,\omega}\ \mathrm{deps}^{n,[0,p]}_{\mathrm{coh2}} & \triangleq & \begin{cases} \lambda d.\lambda l.\lambda \theta.\overrightarrow{\mathrm{coh2}^{n,p}_{\mathrm{frame},q,r,\epsilon,\omega,\theta}(\mathrm{deps}^{n}_{\mathrm{fullcoh}})(d)} \\ (\mathsf{ap}\ \overrightarrow{\mathrm{coh}^{n,p}_{\mathrm{frame},r,0,\omega,\theta}(\mathrm{restr}^{n+2,p}_{\mathrm{frame},q+2,\epsilon}(\overline{\mathrm{deps}_{\mathrm{coh}}}(\mathrm{deps}^{n,[0,p]}_{\mathrm{coh2}}))(d))}) \\ (\mathsf{ap}\ \mathsf{ap}\ \mathrm{restr}^{n,p}_{\mathrm{frame},r,\omega}(\overrightarrow{\mathrm{coh}^{n+1,p}_{\mathrm{frame},q+1,0,\epsilon,\theta}(\mathrm{deps}^{n,[0,p-1]}_{\mathrm{coh2}})(d)}) \\ (\mathrm{coh}^{n,p}_{\mathrm{painting},q,r,\epsilon,\omega}(l_\theta))) \end{cases}
\end{array}
$$

其中 ap 将一个函数应用于等式的两侧，而$\mathrm{coh2}^{n,p}_{\text{框架}}$为：

$$
\mathrm{coh2}^{n,p}_{\mathrm{FRAME}}\ \mathrm{deps}^{n,[0,p]}_{\mathrm{coh2}} \triangleq \begin{cases} \Pi q \le n-p.\Pi r \le q.\Pi \epsilon.\Pi \omega.\Pi \theta.\Pi d. \\ \mathrm{coh}^{n,p}_{\mathrm{frame},r,0,\omega,\theta}(\mathrm{restr}^{n+2,p}_{\mathrm{frame},q+2,\epsilon}(\overline{\mathrm{deps}_{\mathrm{coh}}}(\mathrm{deps}^{n,[0,p]}_{\mathrm{coh2}}))(d)) \circ \\ \mathsf{ap}\ \mathrm{restr}^{n,p}_{\mathrm{frame},r,\omega}(\mathrm{coh}^{n+1,p}_{\mathrm{frame},q+1,0,\epsilon,\theta}(\mathrm{deps}^{n,p}_{\mathrm{coh2}})(d)) \circ \\ \mathrm{coh}^{n,p}_{\mathrm{frame},q,r,\epsilon,\omega}(\mathrm{restr}^{n+2,p}_{\mathrm{frame},0,\theta}(\overline{\mathrm{deps}_{\mathrm{coh}}}(\mathrm{deps}^{n,[0,p]}_{\mathrm{coh2}}))(d)) = \\ \mathsf{ap}\ \mathrm{restr}^{n,p}_{\mathrm{frame},0,\theta}(\mathrm{coh}^{n+1,p}_{\mathrm{frame},q+1,r+1,\epsilon,\omega}(\mathrm{deps}^{n,p}_{\mathrm{coh2}})(d)) \circ \\ \mathrm{coh}^{n,p}_{\mathrm{frame},q,0,\epsilon,\theta}(\mathrm{restr}^{n+2,p}_{\mathrm{frame},r+1,\omega}(\overline{\mathrm{deps}_{\mathrm{coh}}}(\mathrm{deps}^{n,[0,p]}_{\mathrm{coh2}}))(d)) \circ \\ \mathsf{ap}\ \mathrm{restr}^{n,p}_{\mathrm{frame},q,\epsilon}(\mathrm{coh}^{n+1,p}_{\mathrm{frame},r,0,\omega,\theta}(\mathrm{deps}^{n,p}_{\mathrm{coh2}})(d)) \end{cases}
$$

其中 $\mathrm{coh2}_{\mathrm{frame}}$、抽象与应用在传递下的交换性、等式的传递性与同余，以及相对于 θ 的函数外延性，均用于从
下面的类型[4] of ($\overrightarrow{\mathrm{ap}\ \mathrm{coh}^{n,p}}$框架,r,0,$\omega$,$\theta$...($\overrightarrow{\mathrm{ap}\ \mathrm{ap}\ \mathrm{restr}^{n,p}}$框架,r,$\omega$($\mathrm{coh}^{n+1,p}$框架,q+1,0,$\epsilon$,$\theta$...)($\mathrm{coh}^{n,p}$绘画,q,r,$\epsilon$,$\omega$($l_\theta$))))

$$
\begin{array}{l} \overrightarrow{\mathrm{coh}^{n,p}_{\mathrm{frame},r,0,\omega,\theta}\mathsf{ap}\ \mathrm{restr}^{n,p}_{\mathrm{frame},r,\omega}(\mathrm{coh}^{n+1,p}_{\mathrm{frame},q+1,0,\epsilon,\theta})} \\ \overrightarrow{\mathrm{coh}^{n,p}_{\mathrm{frame},q,r,\epsilon,\omega}(\mathrm{restr}^{n,p}_{\mathrm{painting},q,\epsilon}(\mathrm{restr}^{n+1,p}_{\mathrm{painting},r,\omega}(l_\theta)))} \end{array} = \begin{array}{l} \overrightarrow{\mathrm{coh}^{n,p}_{\mathrm{frame},r,0,\omega,\theta}\mathsf{ap}\ \mathrm{restr}^{n,p}_{\mathrm{frame},r,\omega}(\mathrm{coh}^{n+1,p}_{\mathrm{frame},q+1,0,\epsilon,\theta})} \\ (\mathrm{restr}^{n,p}_{\mathrm{painting},r,\omega}(\mathrm{restr}^{n+1,p}_{\mathrm{painting},q+1,\epsilon}(l_\theta))) \end{array}
$$

向下展开类型 coh^{n}，p 层、q、r、ϵ、ω

$$
\overrightarrow{\mathrm{coh}^{n+1,p}_{\mathrm{frame},q+1,r+1,\epsilon,\omega}}\lambda\theta.\begin{pmatrix} \overrightarrow{\mathrm{coh}^{n,p}_{\mathrm{frame},q,0,\epsilon,\theta}}(\mathrm{restr}^{n,p}_{\mathrm{painting},q,\epsilon} \\ \overrightarrow{(\mathrm{coh}^{n+1,p}_{\mathrm{frame},r,0,\omega,\theta}}(\mathrm{restr}^{n+1,p}_{\mathrm{painting},r,\omega}(l_\theta)))) \end{pmatrix} = \lambda\theta.\begin{pmatrix} \overrightarrow{\mathrm{coh}^{n,p}_{\mathrm{frame},r,0,\omega,\theta}}(\mathrm{restr}^{n,p}_{\mathrm{painting},r,\omega} \\ \overrightarrow{(\mathrm{coh}^{n+1,p}_{\mathrm{frame},q+1,0,\epsilon,\theta}}(\mathrm{restr}^{n+1,p}_{\mathrm{painting},q+1,\epsilon}(l_\theta)))) \end{pmatrix}
$$

[4] 省略 框架 参数

3.10 定义绘画相干律

在维度 2 的相干条件层级引入完全依赖的概念会很方便，它将完整的 依赖^{n}, $[0,n]$ coh2 与层级 $n+3$上的额外 E'' 结合起来。

$$\mathrm{DEPS}^{n}_{\mathrm{fullcoh2}} \triangleq \begin{pmatrix} \text{依赖}^{n},\ {}^{[0,n]}_{\mathrm{coh2}} & : & \text{依赖}^{n},\ {}^{[0,n]\,\mathrm{coh2}} \\ E'' & : & \text{框架}^{n+3},\ {}^{n+3}(\text{依赖项}^{}_{\mathrm{restr}}(\overline{\text{依赖项}^{}_{\mathrm{coh}}}(\text{依赖项}^{n},\ {}^{[0,n]}_{\mathrm{coh2}}))) \to \ \mathsf{HSet} \end{pmatrix}$$

接下来，我们可以扩展$\overline{\text{依赖项}_{\mathrm{coh}}}$以构造依赖项$^{n+1}_{\mathrm{fullcoh}}$，它来自依赖项$^{n}_{\mathrm{fullcoh2}}$。

$$\begin{array}{lll} \overline{\mathrm{deps}}_{\mathrm{fullcoh}}\ \mathrm{deps}^{n}_{\mathrm{fullcoh2}} & : & \mathrm{DEPS}^{n+1}_{\mathrm{fullcoh}} \\ \overline{\mathrm{deps}}_{\mathrm{fullcoh}}\ \mathrm{deps}^{n}_{\mathrm{fullcoh2}} & \triangleq & \begin{pmatrix} \overline{\mathrm{deps}}_{\mathrm{coh}}(\mathrm{deps}^{n,[0,n]}_{\mathrm{coh2}}) \\ E'' \end{pmatrix} \end{array}$$

接下来，我们可以定义构造的最后一个分量，即$\mathrm{coh}_{n+1,\ p}$ 绘画，其中 $p \leq n+1$，通过对 $n+1-p$进行递归定义。

$$\begin{array}{lll} \mathrm{coh}^{n+1,p}_{\mathrm{painting}}\ \mathrm{deps}^{n}_{\mathrm{fullcoh2}} & : & \mathrm{coh}^{n+1,p}_{\mathrm{PAINTING}}(\overline{\mathrm{deps}}_{\mathrm{fullcoh}}(\mathrm{deps}^{n}_{\mathrm{fullcoh2}})) \\ \mathrm{coh}^{n+1,n+1}_{\mathrm{painting},q,r,\epsilon,\omega}\ \mathrm{deps}^{n}_{\mathrm{fullcoh2}} & \triangleq & \lambda d.\lambda(l,c).l_{\epsilon} \\ \mathrm{coh}^{n+1,p\leq n}_{\mathrm{painting},q,r,\epsilon,\omega}\ \mathrm{deps}^{n}_{\mathrm{fullcoh2}} & \triangleq & \lambda d.\lambda(l,c).\begin{pmatrix} \mathrm{coh}^{n,p}_{\mathrm{layer},q,r,\epsilon,\omega}(\mathrm{deps}^{n,[0,p-1]}_{\mathrm{coh2}})(d)(l), \\ \mathrm{coh}^{n+1,p+1}_{\mathrm{painting},q,r,\epsilon,\omega}(\mathrm{deps}^{n}_{\mathrm{fullcoh2}})(d,l))(c) \end{pmatrix} \end{array}$$

由后者，我们可以通过对$\mathrm{coh}_{n+1,\ }^{[0,p-1]}$ 绘画进行推导，其中 $p \leq n+2$，并对 p作归纳。

$$\begin{array}{lll} \mathrm{coh}^{n+1,[0,p-1]}_{\mathrm{painting}}\ \mathrm{deps}^{n}_{\mathrm{fullcoh2}} & : & \mathrm{coh}^{n+1,[0,p-1]}_{\mathrm{PAINTING}}(\mathrm{deps}^{n}_{\mathrm{fullcoh}}) \\ \mathrm{coh}^{n+1}_{\mathrm{painting}}\ \mathrm{deps}^{n}_{\mathrm{fullcoh}} & \triangleq & \mathsf{unit} \\ \mathrm{coh}^{n+1,[0,p]}_{\mathrm{painting}}\ \mathrm{deps}^{n}_{\mathrm{fullcoh2}} & \triangleq & (\mathrm{coh}^{n+1,[0,p-1]}_{\mathrm{painting}}(\mathrm{deps}^{n}_{\mathrm{fullcoh2}}), \mathrm{coh}^{n+1,p}_{\mathrm{painting}}(\mathrm{deps}^{n}_{\mathrm{fullcoh2}})) \end{array}$$

3.11 通过归纳完成构造

有了所有这些构造，我们现在就能够指定整个结构，并据此进行归纳推理，其中 $D : \iota$集合$^{<}$ ^{n+1}m 给定：

$$\begin{array}{lll} \mathrm{frame}^{n,[0,n]} & : & \mathrm{FRAME}^{n,[0,n]} \\ \mathrm{painting}^{n,[0,n]} & : & \mathrm{PAINTING}^{n,[0,n]}(\mathrm{frame}^{n,[0,n]}) \\ \mathrm{restr}^{n,[0,n]}_{\mathrm{frame}} & : & \mathrm{restr}^{n,[0,n]}_{\mathrm{FRAME}}\begin{pmatrix} \mathrm{frame}^{n,[0,n]} \\ \mathrm{painting}^{n,[0,n]} \end{pmatrix} \\ \mathrm{deps}^{n,[0,n]}_{\mathrm{restr}} & \triangleq & \begin{pmatrix} \mathrm{frame}^{n,[0,n]} \\ \mathrm{painting}^{n,[0,n]} \\ \mathrm{restr}^{n,[0,n]}_{\mathrm{frame}} \end{pmatrix} \\ \mathrm{deps}^{n}_{\mathrm{fullrestr}}(E) & \triangleq & \begin{pmatrix} \mathrm{deps}^{n,[0,n]}_{\mathrm{restr}} \\ E \end{pmatrix} \\ \mathrm{restr}^{n,[0,n]}_{\mathrm{painting}}(E) & : & \mathrm{restr}^{n,[0,n]}_{\mathrm{PAINTING}}(\mathrm{deps}^{n}_{\mathrm{fullrestr}}(E)) \\ \mathrm{coh}^{n,[0,n]}_{\mathrm{frame}}(E) & : & \mathrm{coh}^{n,[0,n]}_{\mathrm{FRAME}}\begin{pmatrix} \mathrm{deps}^{n}_{\mathrm{fullrestr}}(E) \\ \mathrm{restr}^{n,[0,n]}_{\mathrm{painting}}(E) \end{pmatrix} \\ \mathrm{deps}^{n,[0,n]}_{\mathrm{coh}}(E) & \triangleq & \begin{pmatrix} \mathrm{deps}^{n}_{\mathrm{fullrestr}}(E) \\ \mathrm{restr}^{n,[0,n]}_{\mathrm{painting}}(E) \\ \mathrm{coh}^{n,[0,n]}_{\mathrm{frame}}(E) \end{pmatrix} \\ \mathrm{deps}^{n}_{\mathrm{fullcoh}}(E)(E') & \triangleq & \begin{pmatrix} \mathrm{deps}^{n,[0,n]}_{\mathrm{coh}}(E) \\ E' \end{pmatrix} \\ \mathrm{coh}^{n,[0,n]}_{\mathrm{painting}}(E)(E') & : & \mathrm{coh}^{n,[0,n]}_{\mathrm{PAINTING}}(\mathrm{deps}^{n}_{\mathrm{fullcoh}}(E)(E')) \end{array}$$

其中

$$E : \mathsf{frame}^{n+1,n+1}(\mathsf{deps}^{n,[0,n]}_{\mathsf{restr}}) \to \mathsf{HSet}$$

$$E' : \mathsf{frame}^{n+2,n+2}(\overline{\mathsf{deps}_{\mathsf{restr}}}(\mathsf{deps}^{n,[0,n]}_{\mathsf{coh}}(E))) \to \mathsf{HSet}$$

要从层级 n 从层级 $n+1$提升，只需假设 E 处于层级 $n+1$。我们得到$\mathsf{deps}^{n,[0,n]}_{\mathsf{restr}}$，它们来自层级 n 处的假设，我们可以构造如下内容：

- 框架$^{n+1,[0,n+1]}$ 来自 deps$^{n,[0,n]}$ restr（第3.3节）。绘画$^{n+1,[0,n+1]}$ 来自 deps$^{n}_{\mathsf{fullrestr}}$(E)（第3.4节）。restr$_{n+1,}^{[0,n+1]}$ 框架来自 deps$^{n,[0,n]}$ coh (E)（第3.6节）。restr$_{n+1,}^{[0,n+1]}$ 绘画 (E′) 来自 deps$^{n}_{\mathsf{fullcoh}}$(E)(E′)（第3.7节）。我们根据 deps$^{n}_{\mathsf{fullcoh}}$(E)(E′) 和 coh$^{n,[0,n]}$ 绘画(E)(E′) 定义 deps$^{n,}_{[0,n]}$ coh2 (E)(E′)。coh$_{n+1,}^{[0,n+1]}$ 框架 (E′) 来自 deps$^{n,}_{[0,n]}$ coh2 (E)(E′)（第3.9节）。我们定义 deps$^{n}_{\mathsf{fullcoh2}}$(E)(E′)(E″)，其中 E″ : 框架$^{n+3,n+3}$(deps$_{\mathsf{restr}}$ (deps$_{\mathsf{coh}}$(deps$^{n,}_{[0,n]}$ coh2))) →HSet，它来自 deps$^{n,}_{[0,n]}$ coh2 (E)(E′) 和 E″。coh$_{n+1,}^{[0,n+1]}$ 绘画 (E′)(E″) 来自 deps$^{n}_{\mathsf{fullcoh2}}$(E)(E′)(E″)（第3.10节）。

因此，假设我们可以从 D : ν集合$^{<}$n^{+1}m 构造 frame^{n,n}，就能够在 D 中扩展一些 E : frame$^{n+1,n+1}$(deps$^{n,[0,n]}$ restr) → HSet，从而构造 frame$^{n+1,n+1}$。这足以构造 ν集合$^{=}$ n^{+1}m (D)，其中 D : ν集合$^{<}$n^{+1}m，进而构造 ν集合$_m$：从空的 D 开始即可，因为当 n $= 0$ 时，该结构的所有字段都能由 $\star$ 平凡地填充。

4 呈现与 Rocq 形式化之间的差异

- 形式化并非以 n 为参数，而是以 $k \triangleq n - p$为参数。这样一来，呈现不够直观，因为必须在脑中将 p 加到 k上，重新计算维度，但这让形式化变得更容易。构造某个分量直到 p 时，值得关注的信息并非维度，而是它与预期维度之间的距离（从计算的角度来说）。

- 非正式呈现中的结构 $\mathsf{deps}_{\mathsf{fullrestr}}$、$\mathsf{deps}_{\mathsf{fullcoh}}$ 和$\mathsf{deps}_{\mathsf{fullcoh2}}$ 通过使用一种“拉链”结构来实现 [12]：每当我们在 p 处构造某种结构时，就将 $\mathsf{frame}^{n,[0,n]}$ 拆分为两个方向相反地构造的列表：$\mathsf{frame}^{n,}_{[0,p-1]}$和$\mathsf{frame}^{n,}_{[p,n]}$，这有助于对$p$进行归纳证明。

- 在第 3 节中，我们没有明确说明如何处理限制和相干条件中出现的不等式证明。实际上，我们将不等式表述为严格命题 [8], 或证明均按定义相等的命题，从而简化了形式化。Rocq 不支持一个分量的类型依赖于另一分量的互递归不动点[5], ，也不支持包含缩写的互递归不动点，而我们预计第 3.3、3.6 和 3.9 节中会需要这种不动点。为解决缺少缩写的问题，我们预先定义缩写；为解决缺少递归-递归定义的问题，我们改为递归地构造递归-递归定义的两个分量的 Σ 类型

n,

并通过投影恢复各个组成部分。

[5] 也称为递归-递归定义 [6]

5 与先前工作的比较

为了解决 $frame^{n+1,p}$ 中 $restr^{n,p}$ $frame$ 的依赖关系，以及 $frame^{n+2,p}$ 中 $coh^{n,p}$ $frame$ 和 $frame^{n+2,p}$ and $restr_{n+1,}{}^{p}$ $frame$ 的依赖关系，同时避免将归纳假设的规约与下一步的构造混在一起，先前工作 [10] 假设了所有的 $frame^{n,p}$、$frame^{n+1,p}$、$frame^{n+2,p}$、$restr^{n,p}$ $frame$、$restr_{n+1,}{}^{p}$ $frame$ 和 $coh^{n,p}$ $frame$，以及定义 $frame^{n+1,p}$、$frame^{n+2,p}$ 和 $restr_{n+1,}{}^{p}$ $frame$ 的命题等式。随后，归纳步骤构造了 $frame^{n+3,p}$、$restr_{n+2,}{}^{p}$ $frame$ 和 $coh_{n+1,}{}^{p}$ $frame$，并从归纳假设中继承了 $frame^{n+1,p}$、$frame^{n+2,p}$ 和 $restr_{n+1,}{}^{p}$ $frame$。当前工作可以看作是分析依赖关系如何被精细组织的结果，从而用 $frame^{n+1,p}$、$frame^{n+2,p}$ 和 $restr_{n+1,}{}^{p}$ $frame$ 的定义取代这些命题等式。换言之，在先前工作中，归纳假设结合了形如 $frame^{n+1,p} : HSet$ 的假设，以及形如 $frame^{n+1,0} = unit$ 和 $frame^{n+1,p+1} = \Sigma d : frame^{n+1,p}.\Pi\epsilon.painting^{n,p}(restr^{n,p} frame, 0, \epsilon(d$ 的等式。

$))$.

这些均由定义取代：框架 $^{n+1,p}$、绘画 $^{n+2,p}$、$restr^{n}$ $restr_{n+1,}{}^{p}$ 绘画。

6 展望

我们的构造呈现出一种模式：从 FRAME 出发，定义 PAINTING，相互定义框架与 $restr_{FRAME}$，定义绘画，定义 $restr_{PAINTING}$，相互定义 $restr_{frame}$ 与 coh_{FRAME}，定义 $restr_{painting}$，定义 $coh_{PAINTING}$，相互定义 coh_{frame} 与 $coh2_{FRAME}$，定义 $coh_{PAINTING}$。在 HSet 上，$coh2_{FRAME}$ 可平凡地证明，构造至此停止。如果我们在未截断宇宙 U 中工作，该构造将递归地继续：在维度 i 上证明框架的相干性，并与维度 i 上框架的相干性类型 $+1$ 相互进行，证明维度 i 上绘画的相干性，定义维度 i 上绘画的相干性类型 $+1$。这样一个无限过程同样可以用流来表示，从而由层级 n 的流构造出层级 n $+1$ 的流，如第 3 节所述。这将解决半单纯类型问题 [14]。

参考文献

1 Thorsten Altenkirch、Paolo Capriotti 和 Nicolai Kraus。《以严格相等扩展同伦类型论》。载于 Jean-Marc Talbot 和 Laurent Regnier 编，第*25*届 *EACSL* 计算机科学逻辑年会（*CSL 2016*），*2016*年*8*月*29*日至*9*月*1*日，法国马赛，第 *LIPIcs* 卷 62，页码 21:1–21:17。Dagstuhl 城堡 - 莱布尼茨信息学中心¨，2016。附有形式化证明，见 。 doi:10.4230/LIPIcs.CSL.2016.21。2 Robert Atkey、Neil Ghani 和 Patricia Johann。依赖类型论的关系参数化模型。载于 第*41*届 *ACM SIGPLAN-SIGACT* 编程语言原理研讨会（*POPL '14*），美国加利福尼亚州圣迭戈，*2014*年*1*月*20*日至*21*日。，2014。3 Jean-Philippe Bernardy、Thierry Coquand 和 Guilhem Moulin。参数化类型论的预层模型。《电子理论计算机科学札记》，319:67–82，2015。doi:10.1016/j.entcs.2015.12.006。4 Jean-Philippe Bernardy、Patrik Jansson 和 Ross Paterson。参数关系性与依赖类型。载于 Paul Hudak 和 Stephanie Weirich 编，第*15*届 *ACM SIGPLAN* 函数式编程国际会议论文集（*ICFP 2010*），美国马里兰州巴尔的摩，*2010*年*9*月*27*日至*29*日，页码 345–356。ACM，2010。网址：http://doi.acm.org/10.1145/1863543.1863592, doi:10.1145/1863543.1863592。5 Jean-Philippe Bernardy 和 Guilhem Moulin。参数关系性的计算诠释。载于 *2012*年第*27*届*IEEE*计算机科学逻辑年会，页码 135–144。IEEE，2012。

6 Fredrik Nordvall Forsberg。归纳-归纳定义。博士论文，斯旺西大学，英国，2013。网址：https://ethos.bl.uk/OrderDetails.do?uin=uk.bl.ethos.752308。7 Neil Ghani、Fredrik Nordvall Forsberg 和 Federico Orsanigo。证明相关参数性。载于 Sam Lindley、Conor McBride、Philip W. Trinder 和 Donald Sannella 编，《改变世界的成功之道——献给 Philip Wadler 六十岁生日的论文集》，《计算机科学讲义》第9600卷，页码 109–131。施普林格，2016。8 Ga¨etan Gilbert、Jesper Cockx、Matthieu Sozeau 和 Nicolas Tabareau。无需 K 的定义性证明无关性。《ACM 程序设计语言学报》，3(POPL):3:1–3:28，2019。doi:10.1145/3290316。9 Hugo Herbelin。半单纯类型的依赖类型构造。《计算机科学中的数学结构》，25(5):1116–1131，2015。10 Hugo Herbelin 和 Ramkumar Ramachandra。基于参数性的半单纯集合与半立方集合形式化。《计算机科学中的数学结构》，35:e13，2025。doi:10.1017/S096012952500009X。11 Jason J. Hickey。使用高度依赖类型构造类型论中的形式对象。载于 FOOL 3：第三届面向对象语言基础国际研讨会，1996年7月。网址：https://api.semanticscholar.org/CorpusID:14983194。12 G´erard P. Huet。拉链。《函数式编程杂志》，7(5):549–554，1997。网址：https://doi.org/10.1017/s0956796897002864， doi:10.1017/S0956796897002864。13 Patricia Johann 和 Kristina Sojakova。用于高维参数化的立方范畴。CoRR，abs/1701.06244，2017。网址：http://arxiv.org/abs/1701.06244， arXiv:1701.06244。14 Peter LeFanu Lumsdaine。半单纯类型，2012年9月。网址：https://web.archive.org/web/20180728221737/http://uf-ias-2012.wikispaces.com/Semi-simplicial+type

s.

15 Fedor Part 和 Zhaohui Luo。逻辑丰富的同伦类型论中的半单纯类型。*CoRR*，abs/1506.04998，2015。网址：http://arxiv.org/abs/1506.04998， arXiv:1506.04998。**16** Gordon D. Plotkin。Lambda-可定义性与逻辑关系。技术报告备忘录 SAI-RM-4，爱丁堡大学人工智能学院，1973 年。**17** John C. Reynolds。类型、抽象与参数多态。载于 R. E. A. Mason 编，信息处理 *83*：国际信息处理联合会第九届世界计算机大会论文集，法国巴黎，*1983* 年 *9* 月 *19–23* 日，第 513–523 页。北荷兰出版社/国际信息处理联合会，1983 年。**18** William W. Tait。有限类型泛函的内涵解释 I。《符号逻辑杂志》，32(2):198–212，1967 年。**19** Vladimir Voevodsky。半单纯类型，2012 年 11 月。在 [9]第 8 节中有简要介绍。